Framework for Privacy-aware Content Distribution in Peer-to-Peer Networks with Copyright Protection

Amna Qureshi
aqureshi@uoc.edu

Directors:
Dr. David Megías Jiménez, Dr. Helena Rifà Pous

KISON Research Group
Doctoral Programme in the Network and Information Technologies
Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC)

October 2014

Abstract

The use of Peer-to-Peer (P2P) networks for multimedia distribution has spread out globally in the recent years. The mass popularity is primarily driven by the efficient distribution of content, also giving rise to piracy and copyright infringement, and to privacy concerns. An end user (buyer) of a P2P content distribution system does not want to reveal his/her identity during a transaction with a content owner (merchant), whereas the merchant does not want the buyer to further re-distribute the content illegally. Therefore, there is a strong need for content distribution mechanisms over P2P networks that do not pose security and privacy threats to the copyright holders and end users, respectively. However, the current systems that are developed with a purpose of providing copyright and privacy protection to the merchant and end users employ cryptographic mechanisms at a cost of high computational and communicational burdens which make these systems impractical to distribute large sized files, such as music albums or movies.

In order to develop a framework that could provide an appropriate balance between distributing copyrighted contents on a large-scale and preserving the privacy rights of end users, a review analysis of the existing P2P content distribution systems is conducted with a focus on the design challenges and possible solutions to achieve both copyright protection and user's privacy. The review of current P2P systems satisfying either one or both security and privacy properties shows that most of the systems incur high computational and communicational burdens at the content owner's end and/or at the end user's end. Consequently, to preserve multimedia owners' ownership properties and end users privacy in an efficient manner, a secure and privacy-aware multimedia content distribution framework is proposed that enables content owners' to distribute their large-sized digital contents without a fear of copyright violation at reduced delivery costs and simultaneously allows end users to receive legal content without fear of privacy breach. Based on this framework, two different asymmetric fingerprinting protocols are proposed for the distribution of fingerprinted content from a merchant to an end user of a P2P system. In the first scheme, homomorphic encryption of selected wavelet coefficients is used for achieving asymmetric fingerprinting. The second solution does not require homomorphic encryption and uses a collection of non-trusted proxy peers for distributing the most relevant part of the content from the merchant to the buyer, applying fragmentation, permutation and symmetric encryption.

Finally, a detailed security and performance analysis is provided to show that the proposed content distribution framework provides a fine balance between security, privacy and efficiency.

A comparative analysis of the proposed systems shows that the second alternative is more efficient than the first one both as computation time and communicational burden are concerned, at the price of involving more parties (the proxies) in the protocol. The proposed systems are also compared to other proposals of the literature showing their advantages.

Resumen

El uso de soluciones Peer-to-Peer (P2P) para la distribución multimedia se ha extendido a nivel mundial en los últimos años. La amplia popularidad de este paradigma se debe, principalmente, a la distribución eficiente de los contenidos, pero también da lugar a la piratería, a la violación del copyright y a problemas de privacidad. Un usuario final (comprador) de un sistema de distribución de contenidos P2P no quiere revelar su identidad durante una transacción con un propietario de contenidos (comerciante), mientras que el comerciante no quiere que el comprador pueda redistribuir ilegalmente el contenido más adelante. Por lo tanto, existe una fuerte necesidad de mecanismos de distribución de contenidos a través de redes P2P que no supongan un riesgo de seguridad y privacidad a los titulares de derechos y los usuarios finales, respectivamente. Sin embargo, los sistemas actuales que se desarrollan con el propósito de proteger el copyright y la privacidad de los comerciantes y los usuarios finales emplean mecanismos de cifrado que implican unas cargas computacionales y de comunicaciones muy elevadas que convierten a estos sistemas en poco prácticos para distribuir archivos de gran tamaño, tales como álbumes de música o películas.

Con el fin de desarrollar un marco que pueda proporcionar un equilibrio adecuado en la distribución de un contenido con derechos de autor a gran escala y la preservación de los derechos de privacidad de los usuarios finales, se ha llevado a cabo un análisis de revisión de los sistemas de distribución de contenidos P2P existentes poniendo énfasis en los retos de diseño y las posibles soluciones para lograr tanto la protección de los derechos de autor como la privacidad de los usuarios. La revisión de los sistemas P2P actuales que cumplen una o ambas propiedades de seguridad y privacidad muestra que la mayoría de estos sistemas requieren de altas cargas computacionales y de comunicaciones en el extremo del propietario del contenido y/o en el extremo del usuario final. En consecuencia, para preservar la propiedad intelectual de los propietarios de los contenidos y la privacidad de los usuarios finales de una manera eficiente, se propone un marco de distribución de contenidos multimedia seguro y respetuoso con la privacidad que permite a los propietarios distribuir contenidos digitales de gran tamaño sin temor a la violación del copyright con costes de entrega reducidos y, al mismo tiempo, proporciona la posibilidad a los usuarios finales de recibir contenido legal sin temor a la violación de su privacidad. Sobre la base de este marco, se proponen dos protocolos de fingerprinting asimétrico diferentes para la distribución de contenidos de un comerciante a un usuario final a través de un sistema P2P. En el primer esquema, se utiliza criptografía homomórfica para cifrar un conjunto seleccionado de coeficientes wavelet para conseguir el fingerprinting asimétrico. La segunda

solución no requiere cifrado homomórfico y utiliza un conjunto de proxies no necesariamente honestos para la distribución de la parte más relevante del contenido del comerciante al comprador, aplicando fragmentación, permutación y criptografía simétrica.

Finalmente, se realiza un análisis detallado de la seguridad y el rendimiento que muestra que el marco de distribución de contenidos propuesto ofrece un buen equilibrio entre la seguridad, la privacidad y la eficiencia. Un análisis comparativo de los sistemas propuestos demuestra que la segunda alternativa es más eficiente que la primera, tanto en lo referente al tiempo de cómputo como a la carga de comunicaciones, al precio que intervengan más participantes (los proxies) en el protocolo. Los sistemas propuestos también se comparan con otras propuestas de la literatura para mostrar sus ventajas.

Resum

L'ús de solucions Peer-to-Peer (P2P) per a la distribució multimèdia s'ha estès a nivell mundial en els últims anys. L'àmplia popularitat d'aquest paradigma rau, principalment, en la distribució eficient dels continguts, però també dóna lloc a la pirateria, a la violació del copyright i a problemes de privadesa. Un usuari final (comprador) d'un sistema de distribució de continguts P2P no vol revelar la seva identitat durant una transacció amb un propietari de continguts (comerciant), mentre que el comerciant no vol que el comprador pugui redistribuir il·legalment el contingut mès endavant. Per tant, existeix una forta necessitat de mecanismes de distribució de continguts a través de xarxes P2P que no suposin un risc de seguretat i privadesa als titulars de drets i als usuaris finals, respectivament. No obstant això, els sistemes actuals que es desenvolupen amb el propòsit de protegir el copyright i la privadesa dels comerciants i dels usuaris finals empren mecanismes de xifrat que impliquen unes càrregues computacionals i de comunicacions molt elevades que converteixen aquests sistemes en poc pràctics per distribuir arxius de mida gran, com ara àlbums de música o pel·lícules.

Amb la finalitat de desenvolupar un marc que pugui proporcionar un equilibri adequat en la distribució d'un contingut amb drets d'autor a gran escala i la preservació dels drets de privadesa dels usuaris finals, s'ha dut a terme una anàlisi de revisió dels sistemes de distribució de continguts P2P existents posant èmfasi en els reptes de disseny i les possibles solucions per aconseguir tant la protecció dels drets d'autor com la privadesa dels usuaris. La revisió dels sistemes P2P actuals que compleixen una o ambdues propietats de seguretat i privadesa mostra que la majoria d'aquests sistemes requereixen d'altes càrregues computacionals i de comunicacions en l'extrem del propietari del contingut i/o en l'extrem de l'usuari final. En conseqüència, per preservar la propietat intel·lectual dels propietaris dels continguts i la privadesa dels usuaris finals d'una manera eficient, es proposa un marc de distribució de continguts multimèdia segur i respectuós amb la privadesa que permet als propietaris distribuir continguts digitals de mida gran sense por de la violació del copyright amb costos de lliurament reduïts i, al mateix temps, proporciona la possibilitat als usuaris finals de rebre contingut legal sense por de la violació de la seva privadesa. Sobre la base d'aquest marc, es proposen dos protocols de fingerprinting asimètric diferents per a la distribució de continguts d'un comerciant a un usuari final a través d'un sistema P2P. En el primer esquema, s'utilitza criptografia homomórfica per xifrar un conjunt seleccionat de coeficients wavelet per aconseguir el fingerprinting asimètric. La segona solució no requereix xifrat homomórfic i utilitza un conjunt de proxies no necessàriament honestos per a la distribució de la part més rellevant del contingut del comerciant al comprador,

aplicant fragmentació, permutació i criptografia simètrica.

Finalment, es realitza una anàlisi detallada de la seguretat i el rendiment que mostra que el marc de distribució de continguts proposat ofereix un bon equilibri entre la seguretat, la privadesa i l'eficiència. Una anàlisi comparativa dels sistemes proposats demostra que la segona alternativa és més eficient que la primera, tant pel que fa al temps de cómput com a la cárrega de comunicacions, al preu que intervinquin més participants (els proxies) en el protocol. Els sistemes proposats també es comparen amb altres propostes de la literatura per mostrar els seus avantatges.

Acknowledgements

I would take this opportunity to sincerely thank all the people who have helped and supported me during a three ear period to complete this dissertation.

First and foremost, I would like to express my gratitude to my thesis supervisors, David Megías and Helena Rifà, for their constant guidance, invaluable advice, encouragement and unwavering support from the initial to the final stage of this dissertation. Thank you both of you for giving me full freedom, believing in my research abilities, inspiring me for fruitful work and providing valuable guidance. I cannot thank you enough for proofreading my research articles and giving me helpful suggestions which always resulted in improved versions of scientific documents. Your critical feedback and constructive advice have greatly improved the quality of this thesis.

I am also very grateful to all the lecturers with whom I have taken online courses in the initial stage of my PhD studies. The courses provided me with a strong foundation which I could build upon in my research.

I would also like to thank the Internet Interdisciplinary Institute (IN3) of Universitat Oberta de Catalunya (UOC) for offering me a research grant for my PhD studies in Network and Information Technologies. I gratefully acknowledge the funding provided by the Spanish government for national projects TSI2007-65406-C03-03 "E-AEGIS", TIN2011-27076-C03-02 "CO-PRIVACY" and CONSOLIDER INGENIO 2010 CSD2007-0004 "ARES" that supported my research at IN3.

Thanks to all my colleagues and friends at IN3 for making my stay a most rewarding and memorable one.

This thesis would not have been possible without the love and support of my family. Big thanks to my husband Shahwaiz, for being so supportive, patient and understanding during this roller-coaster journey. This thesis would never have completed without your support. To my sister Ayesha: thank you so much for your continued moral support. To my brother Ally: thank you for always inspiring me to do my very best in life and to stay strong through hard times. I would especially like to express my gratitude and love to my mother for her continued support and encouragement. Without your love, valuable advice, unconditional support and countless prayers, I could have never achieved this goal. I owe this achievement to you.

Finally, thank you God Almighty for giving me the strength, motivation and ability to accomplish this milestone in my life within stipulated time.

Contents

List of Figures

List of Tables

Abbreviations

ACC	**A**nti-**C**ollusion **C**ode
AES	**A**dvanced **E**ncryption **S**tandard
AKE	**A**uthenticated **K**ey **E**xchange protocol
APFS	**A**nonymous **P**2P **F**ile **S**haring
ART-P2P	**A**utonomous **R**ange **T**ree **P**2P
AVC	**A**dvanced **V**ideo **C**oding
AWGN	**A**dditive **W**hite **G**uassian **N**oise
BER	**B**it **E**rror **R**ate
BF	**B**ase **F**ile
BIBD	**B**alanced **I**ncomplete **B**lock **D**esign
B-S	**B**oneh-**S**haw code
CA	**C**ertification **A**uthority
CAN	**C**ontent **A**ddressable **N**etwork
CCA	**C**opy **C**ontrol **A**ssociation
CDN	**C**ontent **D**elivery or **D**istribution **N**etwork
CPTWG	DVD **C**opy **P**rotection **T**echnical **W**orking **G**roup
CSS	**C**ontent **S**cramble **S**ystem
CP	**C**ontent **P**rovider
CVES	**C**haotic **V**ideo **E**ncryption **S**cheme
C2C	**C**ustomer-to-**C**ontent provider
db4	4-coefficient **D**aubechies filter
DC-net	**D**ining **C**ryptographers **n**etwork
DCT	**D**iscrete **C**osine **T**ransform
DCRA	**D**ecisional **C**omposite **R**esiduosity **A**ssumption
DC-QIM	**D**istortion **C**ompensated-**Q**uantization **I**ndexed **M**odulation

DDHA	**D**ecisional-**D**iffie **H**ellman **A**ssumption
DES	**D**ata **E**ncryption **S**tandard
DFT	**D**iscrete **F**ourier **T**ransform
DHT	**D**istributed **H**ash **T**able
DLL	**D**ynamic **L**ink **L**ibrary
DM	**D**ither **M**odulation
DRM	**D**igital **R**ights **M**anagement
DSA	**D**igital **S**ignature **A**lgorithm
DWT	**D**iscrete **W**avelet **T**ransform
D-H	**D**iffie-**H**ellman key exchange
EKE	**E**ncrypted **K**ey **E**xchange
FFmpeg	**F**ast **F**orward **M**oving **P**icture **E**xperts **G**roup
FPSUM-HE	**F**ramework for preserving **P**rivacy and **S**ecurity of **U**ser and **M**erchant based on **H**omomorphic **E**ncryption
FPSUM-PD	**F**ramework for preserving **P**rivacy and **S**ecurity of **U**ser and **M**erchant with **P**roxy-based **D**istribution
GoP	**G**roup of **P**ictures
IPP	**I**dentifiable **P**arent **P**roperty code
IT	**I**nformation **T**echnology
IDWT	**I**nverse **D**iscrete **W**avelet **T**ransform
JPEG	**J**oint **P**hotographic **E**xperts **G**roup
LSB	**L**east**S**ignificant **B**it
MCLT	**M**odulated **C**omplex **L**aplace **T**ransform
MIMA	**M**an-**I**n-the-**M**iddle **A**ttack
MLE	**M**aximum **L**ikelihood **E**stimations
MoRE	**M**aster of **R**everse **E**ngineering
MP3	**M**ovie **P**icture **E**xpert **G**roup-3
MSE	**M**ean **S**quare **E**rror
MTW	**M**odified **T**ransform **W**atermark
NC	**N**ormalized **C**orrelation
ODG	**O**bjective **D**ifference **G**rade
OMA	**O**pem **M**obile **A**lliance
OS	**O**perating **S**ystem

PAKE	**P**assword-**A**uthenticated **K**ey **E**xchange
PCA	**P**rinciple **C**omponent **A**nalysis
PEAQ	**P**erceptual **E**valuation of **A**udio **Q**uality
PKI	**P**ublic **K**ey **I**nfrastructure
PGP	**P**retty **G**ood **P**rivacy
PSNR	**P**eak **S**ignal-to-**N**oise **R**atio
P2P	**P**eer-to-**P**eer
P^5	**P**eer-to-**P**eer **P**ersonal **P**rivacy **P**rotocol
QIM	**Q**uantization **I**ndexed **M**odulation
QIMM	**Q**uantization **I**ndexed **M**odulus **M**odulation
RAM	**R**andom **A**ccess **M**emory
RC4	**R**ivest **C**ipher 4
RC6	**R**ivest **C**ipher 6
RDM	**R**ational **D**ither **M**odulation
RSA	**R**ivest **S**hamir **A**dleman
SDL	**S**imple **D**irectMedia **L**ayer
SD-QIM	**S**ubtractive **D**ither-**Q**uantization **I**ndexed **M**odulation
SF	**S**upplementary **F**ile
SFP	**S**ecure **F**rame **P**roof
SFW	**S**hared **F**ragment **W**atermarking
SHA	**S**ecure **H**ash **A**lgorithm
SMC	**S**ecure **M**ulti-party **C**computation protocol
SNR	**S**ignal-to-**N**oise **R**atio
SRP	**S**ecure **R**emote **P**assword protocol
SS	**S**pread **S**pectrum
SVD	**S**ingular **V**alue **D**ecomposition
TCPA	**T**rusted **C**omputing **P**latform **A**lliance
TCG	**T**rusted **C**omputing **G**roup
TTP	**T**rusted **T**hird **P**arty
VQMT	**V**ideo **Q**uality **M**easurement **T**ool
WAKE	**W**ord **A**uto **K**ey **E**ncryption
ZKPI	**Z**ero **K**nowledge **P**roof-of-**I**dentity

Operators

$\mathcal{E}_k(\cdot)$	Paillier encryption with a key K
$\mathcal{D}_k(\cdot)$	Paillier decryption with a key K
$E_k(\cdot)$	asymmetric/symmetric encryption with a key K
$\odot_{\mathcal{M}}$	linear operator in the plaintext space $\mathcal{M}$
$\odot_{\mathcal{C}}$	linear operator in the ciphertext space $\mathcal{C}$
$h(\cdot)$	hash
$Q_{2\Delta}(\cdot)$	quantization with 2Δ
$()^{-1}$	modular inverse in the cyclic group $\mathbb{G}$
$\otimes$	embedding of a fingerprint in the encrypted domain
$\mid$	concatenation

Symbols

Symbol	Description
a_f	approximation coefficients of fingerprinted file
a_j or a_k	approximation coefficients of audio/video file
a_p	approximation coefficients of a pirated copy Y'
a_t	approximation coefficients of selected key frames $\mathcal{I}_t$
$a_{t,j}$	approximation coefficients of $\mathcal{I}_t$ with j bits
a'	a constant used in Canny-edge detection technique
a'_k	permuted and encrypted a_k
$attack_{SNR}$	a constant used in audio fingerprinting
AGR	an agreement between a merchant and a buyer
B_i	an i^{th} buyer
$\mathcal{B}_k$	non-overlapping blocks of approximation coefficients a_j of video
BF	a base file
BF^0	a base file embedded with all 0s
BF^1	a base file embedded with all 1s
c	total number of colluders
$\hat{c}$	a ciphertext obtained using a homomorphic encryption
c'	a scaling factor used in the embedding process
c_0	a coalition size
C_i	a collusion set
CA_L	L-level approximation coefficients
CA_{ext}	an external certification authority
CA_R	an internal certification authority
CEK	a content encryption key
$\text{CERT}_{\text{CA}_\text{R}}(\cdot)$	a certificate certified by CA_R

$\mathrm{CERT}_{\mathrm{CA_R}}(B_i)$	public-certificate of a buyer B_i certified by $\mathrm{CA_R}$
$\mathrm{CERT}_{\mathrm{K_{pB_i}}}(\cdot)$	a certificate certified by B_i
$\mathrm{CK_i}$	number of content keys
d_j	a dither vector containing j bits
DA	DRM agent
DP	a downloading peer
$\hat{E}$	an eavesdropper
E_1	an encrypted hash
$E_{K_{pB}}(\cdot)$	an encryption with a buyer's public key
$E_{K^*_{pB_i}}(\cdot)$	an encryption with a buyer's anonymous key
$E_{K_{pJ}}(\cdot)$	an encryption with a judge's public key
$E_{K_{pM}}(\cdot)$	an encryption with a merchant's public key
$E_{K_{pMO}}(\cdot)$	an encryption with a monitor's public key
f_i	a fingerprint of a buyer i
$f_{i,j}$	a fingerprint of a buyer i with j bits
fa^0_j	permuted and encrypted fragments of a_k for 0-bit
fa^1_j	permuted and encrypted fragments of a_k for 1-bit
fl	an index of a file
f_{pr_i}	a permuted segment of a fingerprint assigned to a proxy peer
f'_i	an extracted fingerprint
f''_i	a colluded fingerprint
F	a collection of fingerprint codewords
$\mathcal{F}_k$ or $\mathcal{F}_\ell$	non-overlapping frames of approximation coefficients of an audio file
$\mathcal{F}_1$	a non-overlapping frame $\in \mathcal{F}_k$
$\mathcal{F}^1_\ell$	frames of an audio file embedded with 1 bits
$\mathcal{F}^0_\ell$	frames of an audio file embedded with 0 bits
$\mathcal{F}'_k$ or $\mathcal{F}'_\ell$	non-overlapping frames of approximation coefficients of a fingerprinted audio file
FP	a digital fingerprint
g	a generator of $\mathbb{G}$
$\mathcal{G}$	a generator $\in \mathbb{Z}^*_{\mathcal{N}^2}$
I	an initiating peer
$\mathcal{I}_t$	number of key frames of a video selected for embedding
ID_{P_a}	a real identity of a peer P_a

ID_{P_b}	a real identity of a peer P_b
J	judge
K	a constant used in the fingerprint generation
K_1	one-time session key generated during AKE protocol
(K_{pJ},K_{sJ})	a public and private key pair of a judge
(K_{pM},K_{sM})	a public and private key pair of a merchant
(K_{pMO},K_{sMO})	a public and private key pair of a monitor
(K_{pB},K_{sB})	a public and private key pair of a buyer B_i
$(K^*_{pB_i}, K^*_{sB_i})$	an anonymous public and private key pair of a buyer B_i
K_{ses_j}	a set of one-time session keys
$K_{s_{P_a}}$	a secret key of P_a
$K_{s_{P_b}}$	a secret key of P_b
l	length of a permuted fingerprint segment and permutation key
L	level of the DWT
$\mathcal{L}$	length of non-overlapping audio frames
LP	a license provider
m	length of a fingerprint code
m_1	a plaintext message $\in \mathcal{M}$
m_2	a plaintext message $\in \mathcal{M}$
M_i	number of Chaum mixes
MP	a music player
MO	monitor
n	total number of proxy peers
N	total number of users (buyers) in the system
$N^{'}$	number of pixels in a window
$\mathcal{N}$	a large prime number that is a product of $\mathcal{P}$ and $\mathcal{Q}$
p	a secret vector generated during fingerprint generation
pc	a pirated codeword
ps_j	permuted segments assigned to Pr_j
P	a large prime number used in pseudo-identity generation
$\mathcal{P}$	a large prime number of Paillier cryptosystem
P_a	a receiving peer
P_b	a content providing peer

P^i	a set of buyers (peers)
P_{B_i}	pseudonym of a buyer B_i
PI_{P_a}	a pseudo-identity of P_a
PI_{P_b}	a pseudo-identity of P_b
PI	pseudo-identity
PIC	a pseudo-identity certificate
PW	password
Pr_j	a set of proxy peers
q	a query
Q	a large prime number used in pseudo-identity generation
$\mathbb{Q}$	a large prime number of Paillier cryptosystem
r	a secret number used in pseudo-identity generation
R	a responding peer
R_i	number of routers
RC	a registration centre
RP	a requesting peer
s_j	segments of the fingerprint f_i
sk	a secret watermark embedding key
S_i	a score of a user
SF	a supplementary file
SP	super peer
$\text{Sign}_{B_i}(\odot)$	($\odot$) signed by a buyer B_i
$\text{Sign}_{K^*_{pB_i}}(\odot)$	($\odot$) signed by a buyer B_i using anonymous key
$\text{Sign}_{MO}(\odot)$	($\odot$) signed by a monitor MO
t_a	L-level DWT approximation coefficients
$t_{i,i+1}$	a transaction record
T	a threshold value in Canny-Edge detection technique
T_a	a pseudo-identity of a tail node of a requesting peer a
T_b	a pseudo-identity of a tail node of a provider peer b
$T_{\hat{E}}$	a pseudo-identity of a tail node of an eavesdropper $\hat{E}$
T_I	a pseudo-identity of a tail node of an initiating peer I
T_R	a pseudo-identity of a tail node of a responding peer R
TID	a transaction ID

U	colluders
V	total number of frames in a video
w_i	number of watermark bits
ω_ℓ	vector norm of non-overlapping frames of an audio file
W_i	a set of secret keys
x_i	samples of an original signal
X	an original content
X_1	total samples in an original audio content
Y	a fingerprinted copy
Y'	a pirated copy
$\mathcal{YUV}$	luminance and chrominance components of an image
$z_1/z2$	random integers $\in \mathbb{Z}^*_{\mathcal{N}^2}$
z_k^0/z_k^1	values used in video file partitioning algorithm
$\mathbb{Z}_\mathcal{N}$	a set of integer numbers
$\mathbb{Z}^*_\mathcal{N}$	a set of integer numbers $\in \mathbb{Z}_\mathcal{N}$
$\mathbb{G}$	a finite cyclic group
α_1	a constant used in audio watermarking algorithm
α_2	a constant used in audio watermarking algorithm
$attack_{SNR_i}$	a constant used in audio watermarking algorithm
Δ	quantization step size
τ	fixed time period set for *MO*
$\varepsilon/\varepsilon_1$	probability of accusing an innocent user
ε_2	probability of missing a colluder
σ_j	set of permutation keys
$(\sigma_j)^{-1}$	set of inverse permutation keys
Υ_1	a secret number of a peer
Υ_2	a secret number of a peer
ζ	a constant used in Paillier cryptosystem
κ	a constant used in Paillier cryptosystem
β	a constant used in Paillier cryptosystem
λ	a private key in Paillier cryptosystem
γ_1	a constant used in a session key generation
γ_2	a constant used in a session key generation

Chapter 1

Introduction

In this chapter, the basics of P2P content distribution systems are briefly introduced, followed by a discussion about the main reasons that why these P2P systems raise security and privacy concerns despite their potential benefits. Then, the motivation of the thesis is provided by discussing the research problems of the content protection and privacy-preserving mechanisms, and the problem faced in the integration of copyright and privacy protection techniques in a distributed environment. The research objectives, the methodology and the main contributions of this thesis are presented in the subsequent sections and, finally, at the end of this chapter, the thesis organization is outlined.

1.1 Introduction

In recent years, the prosperity of digital and information technologies has opened limitless channels for distribution of content such as text, audio, video, graphics, animations and software. In the past, the content distribution was limited to tightly controlled broadcasts or the sale of analog media, but, with the digital revolution, the Internet has emerged as a new and efficient content distribution channel. The market for digital content distribution continues to grow due to technological improvements in the bandwidth of network connections and the decline in bandwidth consumption price. Examples of content distribution include, but are not limited to, bulk data transfer, streaming continuous media, shared data applications, web cache updating and interactive gaming. The content providers need to distribute their respective content efficiently to users. This requires delivery of data from one or more senders to multiple receivers. Many different

service architectures, ranging from centralized client-server to fully distributed architectures are available in today's world for content distribution on the Internet.

The conventional model for delivering content to a user or group of users is a client-server model. Traditional client-server systems are dependent on a centralized server to distribute the content to the clients. Under this model, a centralized server sends its contents to the interested clients. Eventually, the server suffers congestion and bottlenecks due to the increasing demands on its content, leading it to a single point of failure. In order to improve the distribution service quality and efficiency to large audiences, a new technology called Content Distribution Network or Content Delivery Network (CDN) emerged. A CDN is a network of dedicated servers that are strategically spread across the Internet and that cooperate to deliver content to end users, e.g. *Akamai* (1998) is the largest commercial content delivery network that delivers the content through a global network powered by more than 100,000 servers. However, content providers using a CDN have to bear an initial infrastructure investment and high maintenance costs of servers, thus making it out of reach for small enterprises and non-profit organizations. Moreover, a CDN suffers a scalability problem such that the system efficiency severely degrades when a large numbers of users access the network simultaneously. Thus, a new distribution revolution is needed.

This revolution is coming in the form of P2P networks. In recent years, P2P networks have emerged as a popular solution to deliver multimedia content efficiently to a large number of Internet users. The popularity of these systems is attested by the fact that, in some countries, P2P traffic accounts for more than 60% of the overall Internet traffic (García-Dorado et al., 2012). A P2P system can be defined as a decentralized computing system in which nodes, referred to as peers, use the Internet to communicate with each other directly. All the peers in this interconnected network provide resources to other peers, including bandwidth, storage space and computing power. P2P networks underlie numerous applications, e.g. instant messaging (Instant Messaging Computer (*ICQ*, 1996)), grid computing (*Seti@home*, 1999), content delivery (*BitTorrent*, 2000), file sharing (*eDonkey2000*, 2000) and content streaming (*PeerCast*, 2006). However, the most popular P2P applications remain file sharing and content distribution. The success of *Napster* (2011) (originally founded in 1999, ceased its operations due to copyright infringement and was eventually acquired by Rhapsody in 2011), the first commercial P2P content distribution system, paved the way for many new distribution systems such as *Internap* (1996), *gtk-Gnutella* (2000), and *BitTorrent* (2000). Today, P2P traffic levels are still growing, with 300 million users sharing files via BitTorrent every month.

Unlike traditional client-server models and CDNs, the P2P technology provides cost efficiency (low infrastructure cost), scalability, fault tolerance, less administrative and control requirements and exposure to a large number of users. These benefits are the attractive features for content providers towards the adoption of P2P systems and many parties, ranging from individual artists and producers to large multimedia content providers, are interested in using this technology. The cost of content distribution in P2P is much lower for the content provider, which results in lower prices for buyers and increased profits for the multimedia content owners.

Despite the valuable characteristics offered by P2P systems, there is a major obstacle to their widespread acceptance and usage. The main problem of P2P content distribution systems is the lack of security: the P2P technology is not sufficiently mature to support a secure method for distributing copyrighted content through these systems. Unfortunately, the content providers are reticent about using P2P networks for content distribution. The reason for this reluctance is the fact that P2P systems are considered to be associated with the illegal sharing of copyrighted materials, especially music and videos. The content providers apparently fear losing control of content ownership and worry about the illegal activity promotion. Additionally, the larger the number of users in a P2P system, the more illegal copies are reproduced and re-distributed. Consequently, tracing a copyright violator in such a large-scale network is an immense task.

The copyright infringement problem motivates the development of content protection techniques to prevent piracy. The content protection technologies allow the creators of an original digital content to enforce his/her copyright in the content and trace a person responsible of illegally re-distributing that content. However, these content protection techniques have been criticized for implicating users' privacy by collecting information about the users, such as transaction history, usage habits, purchasing behaviors or other profiling information. A priori it places the user into an adversarial relation with the content provider. Hence, the incorporation of a content protection mechanism in a P2P system can have serious effects on the privacy interests of the users: the fact that a tracing mechanism makes use of a systematic record which details what multimedia files are downloaded through a specific IP address, the history of files shared or downloaded, or a list of the peers with whom a user has interacted in the past, ultimately disrespects the private space of the user. A great deal of information regarding the user preferences can be collected in multimedia distribution by tracking the user activities at the provider side, thus compromising the user's privacy. Also, while downloading the file, the user reveals his/her details, such as plain-text queries and IP addresses, to another user that provides the services.

Privacy includes anonymity and unlinkability. Anonymity refers to the requirement that a

user should be able to participate in the network without revealing his/her identity. However, anonymity must not imply impunity for malicious users who try to disrupt the network. Unlinkability means that different interactions between a specific user and an entity within the network communicating system cannot be related to each other neither by the system nor by an external observer. If a system is anonymous but the different actions by the same user are linkable, the user's purchasing activities can be obtained from such linkages; this might suffice to infer the user's identity.

Various mechanisms exist to provide privacy to the end users, but at the cost of less accountability. This creates the conflict between the basic starting point of preserving the interests of the content provider or copyright owner and protecting the privacy rights of the user, i.e. increased accountability (more security to provider) is proportional to decreased anonymity (less user privacy). Thus, the issue of maintaining a trade-off between security concerns and privacy interests should be carefully addressed in the development of P2P content distribution systems. In this way, the content owners would be able to distribute their contents to a large number of people without the fear of copyright violation and end users would receive legal content without fear of privacy breach.

1.2 Motivation

In this section, the current research problems related to content and privacy protection mechanisms are discussed.

1.2.1 Content Protection

Recent developments in digital technologies have had a great influence on the content providers such as music and movies' distributors and on their users (buyers). It has become extremely easy for a user to make a high-quality copy and to re-distribute it. In the past, users had a limited access to professional recording equipment. The copies made by users were of a poor quality or too expensive to produce. For these reasons, illegal copying and re-distributing of music and video was kept at a reasonable level. Nowadays, digital technologies allow users to make copies of digital content identical to the original. These copies are cheap to produce. Therefore, the amount of the digital data which is illegally re-distributed is growing, making businesses lose their income. Content creators and owners are concerned about the consequences of illegal

copying and re-distribution on a massive scale. Consequently, the need of a protection system that can provide copyright protection by prosecuting unauthorized copying has arisen.

Traditionally, copyright protection of multimedia data has been accomplished by utilizing encryption techniques to prevent unauthorized users access to digital media content. For example, in 1997, a method of implementing device control, known as the Content Scramble System (CSS) was created by the DVD Copy Protection Technical Working Group (CPTWG). CSS is an encryption and decryption system for compliant DVD players. Compliant DVD players possess certain keys, licensed by the DVD Copy Control Association (CCA), which allows them to decrypt the encrypted content on the DVD (Kesden, 2000). In 1999, a European group "Masters of Reverse Engineering" (MoRE) created a program called DeCSS, which copies the content of a DVD directly into a user's hard drive. This copying was possible due to an error on the part of one of the manufacturers, Xing Technology Corporation, in failing to properly encrypt its decryption key. Not only was Xing Technology Corporation's key exposed, but because of the relationship between each of the CSS keys, some 170 keys belonging to other manufacturers were uncovered through reverse engineering and trial and error. This effectively rendered CSS obsolete (*CSS Demystified*, 1999). Then, the industry was forced to recognize that once encryption is removed from a digital content, that content is no longer protected, and that a compliant device is not enough to provide protection. This led to the development of digital watermarking schemes that track and enable the prosecution of people who are involved in illegal re-distribution.

Digital watermarking is based on the science of steganography or data hiding. Steganography comes from the Greek meaning "covered writing". The goal of steganography is to hide a message in a media content in such way that the presence of a message cannot be detected. Watermarking is the process of embedding hidden information, called a watermark, into the digital media, such that the watermark is imperceptible, robust and difficult to remove or alter. With the help of these watermarks, the content provider can find users involved in illegal re-distribution of digital content. Such kind of watermarking is known as forensic watermarking, transaction tracking or digital fingerprinting.

Digital fingerprinting is a method by which a copyright owner can uniquely embed a buyer-specific serial number (representing the fingerprint) into every copy of a digital content that is legally sold. The buyer of a legal copy is then deterred from distributing further copies, because the unique fingerprint can be used to trace back the origin of the piracy. The major challenge in fingerprinting, however, is that all legally distributed copies of the same digital data are similar, with the exception of the unique buyer-specific fingerprint. A coalition of pirates who possess

distinctly fingerprinted copies of the same data can therefore exploit this diversity by comparing their digital data, and possibly detecting and then rendering the fingerprints unreadable. Such an attack is known as collusion. One goal of fingerprinting is thus to ensure that some part of the fingerprint is capable of surviving a collusion attack so as to identify at least one of the pirates. In addition to the collusion attack, a coalition of pirates might individually modify their multimedia content through user-generated distortions. Examples of common user-generated distortions are additive white Gaussian noise, linear filtering, compression, and geometric distortions such as cropping and resizing. Since fingerprinting has the goal of traceability, fingerprinting for digital media should be robust to both collusion as well as user-generated distortions.

Much work on collusion-secure fingerprinting (Boneh & Shaw, 1999; Trappe, Song, Poovendran, & Liu, 2003; Tardos, 2003) has been proposed in the literature. However, some proposed fingerprint codes are too long to be embeddable in the multimedia content and the others provide low collusion resistance. In addition, the longer codewords affect the imperceptibility of the content. On the other hand, low collusion-resistant codes are impractical in real-world scenarios, since the attackers can easily work together to pirate multimedia content due to the rise of multimedia processing techniques. Thus, there is a need to embed a fingerprinting codeword into the content which provides strong collusion resistance, traceability, and is smaller in length.

Traditional digital fingerprinting schemes provide protection to a content provider (merchant) but do not protect the rights of the buyers. These systems implicitly assume the honesty of the content provider and allows a content provider a complete control of the fingerprinting process, thus causing the fingerprinting scheme to be biased and unfair to buyers. If a content provider knows the exact fingerprint inserted to a buyer's copy, he/she can easily reproduce copies of the content containing a user's fingerprint and illegally re-distribute them. As a result, it enables the content provider to falsely accuse and frame an innocent buyer. This unpleasant situation defines the customer's right problem (Cox et al., 1997). It is clear that the customer's right problem actually nullifies the objective and the purpose of fingerprinting itself. It can cause an irresolvable dispute by opening a chance for a malicious user to deny his/her unlawful act and claim that the unauthorized copy was originated from the content provider. To solve this customer's right problem, the concept of asymmetric fingerprinting protocols accommodating the rights of both the buyer and the merchant was introduced by B. Pfitzmann and Schunter (1996). The asymmetric fingerprinting protocol provides (i) non-repudiation: a traitor cannot deny his/her responsibility in the generation of a pirate codeword if he/she is indeed involved in such a piracy and (ii) non-framing: a malicious merchant cannot frame an innocent buyer by distributing a pirated copy which incriminates that particular buyer.

There are various proposals for asymmetric fingerprinting protocols (Martínez-Ballesté, Sebé, Domingo-Ferrer, & Soriano, 2003; Choi, Sakurai, & Park, 2003; Kuribayashi & Mori, 2008; Kuribayashi, 2010). Many works (including but not limited to (Memon & Wong, 2001; Ju, Kim, Lee, & Lim, 2003)) rely on a trusted party that is called watermark certification authority or registration authority, which embeds the fingerprint and sends the fingerprinted content to the buyer. Attempts to remove such trusted party leads to the proposals with double-watermarking techniques or multi-party computation protocols. Double watermarking is discussed to be vulnerable to many deficiencies like quality degradation or ambiguity attacks. Secure multi-party computation (SMC) protocols have been considered only for theoretical evaluation. Consequently the SMC-based asymmetric fingerprinting protocols are found to be inefficient for any practical application. A more recent approach of asymmetric fingerprinting is based on homomorphic cryptosystems that operate on very large algebraic structures, thus increasing the computational and communicational costs. In all the fingerprinting schemes referred above, the complexity of the algorithms deters their practical implementation, since they rely on at least one of the following highly demanding technologies: secure multi-party computation protocols, general zero-knowledge proofs or public key cryptography of the contents. Thus, there is a need to design such a fingerprinting scheme that reduces the computational overhead, large communication bandwidth and also fulfils the desired security requirements.

1.2.2 Privacy Protection

As a result of the dramatically growing popularity of the Internet, the P2P architecture has gradually become the main trend in file distribution systems. In recent years, P2P systems, such as *Napster* (2011), *gtk-Gnutella* (2000) and *BitTorrent* (2000), have become essential media for information dissemination and sharing over the Internet. Concerns about privacy, however, have grown with this rapid development of P2P systems. The major privacy concerns for P2P users is that the users' identities and actions can be revealed by other users. In current P2P systems, attackers may make use of some flaws, such as plain-text query, exposed IP address, and direct file-downloading, to compromise user privacy. Moreover, anyone can take part in the system without having his/her identity verified and any malicious attacker in the system can easily monitor any part of the system and learn who has just provided or requested a certain file, as well as what the file is about, thus compromising data privacy. Thus, with the open and distributed features of P2P systems, achieving data and user privacy is a challenging task.

A consolidated terminology of privacy has been proposed by A. Pfitzmann and Hansen

(2010). It motivates and develops a number of definitions, including anonymity, unlinkability, undetectability, pseudonymity and unobservability. This terminology is developed based on a setting where senders send messages to recipients using a communication network. The definitions in the terminology are made from the perspective of an attacker, who may be interested in monitoring which communication is occurring, which patterns of communication exist, or even in manipulating the communication. The attacker may be an outsider tapping communication lines or an insider able to participate in normal communications and controlling at least some entities.

Anonymity has been defined by A. Pfitzmann and Hansen (2010) as "the state of being not identifiable within a set of subjects, namely the anonymity set". From a communication perspective, the anonymity set is the set of all (uncompromised) network members in the network. In P2P networks, each peer can play three different roles: provider (responder), receiver (requester) and middle (relay) nodes. The provider of a file is the one who offers the file to the file requester, the receiver of a file is the one who requests for the file, and the nodes that help relay the file in the network are middle nodes. Most of the existing literature on P2P agrees on at least three types of anonymity: receiver, provider and mutual anonymity.

1. **Receiver Anonymity:** It deals with hiding the identity of the user who initiated a communication by requesting a file from a provider.

2. **Provider Anonymity:** This deal with hiding the identity of the user who responds to sender's queries and sends files accordingly.

3. **Mutual Anonymity:** A P2P sharing system with mutual anonymity hides the identities of both sender and receiver from each other and from other users in the system. It also hides the communication between a sender and a receiver.

An additional requirement which relates to anonymity is unlinkability, defined as the notion of an attacker being unable to determine the relationship between the sender and the receiver in a communication. Anonymity in terms of unlinkability is defined as "an inability to link a particular message to any provider-receiver pair and any message to a particular provider-receiver pair". The problem of unlinkability is related to anonymity. While a sender might be anonymous with respect to a message's content, by relating messages of the same sender, an attacker gains knowledge from multiple messages which can lead to an anonymity compromise. At least, a sender is identified, with any of his/her messages serving as pseudonym. The attacker can then derive behavioural patterns from the linked messages and thus, uncover the identity of

the sender. Therefore, to have a perfect anonymity, messages have to be unlinkable.

Various privacy-preserving mechanisms have been proposed that serve as tools for privacy protection in content distribution applications such as anonymity techniques that also provide unlinkability, trust-based management and cryptographic techniques. Anonymity techniques are used to make a user indistinguishable from other users, thus providing anonymity among a group of users. These techniques are also used to make communication ambiguous in order to make it difficult for malicious users to collect information about the system entities and the shared data, thus providing unlinkability. Among various anonymity techniques, anonymous communication systems and anonymous authentication techniques are mostly used. Anonymous communication aims to preserve communication privacy within the shared network setting. Works in this domain include mix networks (Chaum, 1988) and onion routing (Reed, Syverson, & Goldschlag, 1998). However, these anonymous communication approaches incur extra overhead to both the system and the users. The overhead is caused by encryptions and decryptions, anonymous transmissions, insertion of fake traffic and an increased routing to provide anonymity between two communicating users. On the other hand, anonymous authentication aims to provide a balance between privacy and accountability. Accountability has traditionally been achieved through authentication mechanisms (group signatures (Chaum & Van-Heyst, 1991), authenticated key exchange (Bellovin & Merrit, 1992)) which verify the identity of a client who requests a service. In P2P systems, finding a reasonable trade-off between anonymity and accountability is rather hard, since existing accountability systems assume a client-server architecture in which only the clients, but not the servers, care about their privacy.

Trust management techniques (reputation-based (Kamvar, Schlosser, & Garcia-Molina, 2003), credentials-based (Xiong & Liu, 2004)) have been proposed as mechanisms that allow potentially unknown parties to decide whom is trusted enough to provide or access requested data. They allow unknown parties to access resources by showing appropriate credentials that prove their qualifications to acquire data. Most prior approaches of trust and trust management are identity-based, which means that real user identities are needed to make authentication and verification. However, this mechanism does not work when considering user's anonymity. Even though many anonymous schemes correlate a real ID with a pseudonym, the trust problem becomes more difficult in the proof of the correlation between these two entities. Therefore, trust management schemes need to be further explored in anonymous P2P environments.

Cryptographic techniques include cryptographic protocols such as zero-knowledge proofs of identity (Feige, Fiat, & Shamir, 1998) and secret sharing (Schaathun, 2003). A zero-knowledge proof is a cryptographic protocol between two parties whereby the first party wants to prove

his/her identity to the second party without revealing anything about his/her identity. However, the zero-knowledge proof of identity protocols are based on complex mathematical algorithms and thus require heavy computations for both parties involved, i.e. the prover and the verifier. In another cryptographic protocol, i.e. a secret-sharing scheme, shares are distributed to parties such that only authorized subsets of parties can reconstruct the secret. Secret-sharing schemes are important tools in cryptography and they are used as a building block in many anonymous protocols. However, one of the main problems in secret sharing is to find constructions for which the shares given to the users are as small as possible.

In distributed environments, neither of these mechanisms alone can ensure privacy, thus, combining these techniques can provide more privacy guaranties to a user. Every approach described in this section has its own strengths and weaknesses and thus, the choice entirely depends on the requirements of the system and its operating context. Moreover, in achieving anonymity, there is always a trade-off between the following factors: efficiency and anonymity, and accountability and anonymity. Thus, a P2P system with low-latency anonymous mechanism providing better anonymity, efficiency and accountability trade-offs is of high practical importance.

It is, however, pertinent to mention here that in the literature of P2P content distribution systems, much of the proposed systems either offer copyright protection or privacy preservation. This is due to a fact that combining security and privacy mechanisms in a P2P environment is a challenging task. Efforts in addressing security and privacy properties simultaneously are still unsuccessful because of the intricacy of each other. Often, exertion for addressing one of these factors may increase the severity of the other, i.e. strategies with the intention of enhancing privacy in P2P systems are often characterized with security concerns and vice versa. Consequently, strategic solutions for addressing security issues in P2P networks should be sensitive to ideas of privacy.

1.3 Research Objectives

This thesis focuses on development of secure and privacy-preserving algorithms in a P2P network. Some of the specific sub-tasks associated with this work include:

1. To develop a framework that provides a firm foundation for designing content protection and privacy-preserving protocols simultaneously. The framework must provide the threats, security requirements, trust assumptions and the various building blocks on which these protocols are based.

2. Within a proposed P2P content distribution framework, develop an asymmetric fingerprinting protocol with low computational overheads and low communication bandwidth requirements for secure distribution of the content from the merchant to an end user of the system. The designed scheme must also be able to provide non-repudiation, non-framing, unlinkability and collusion resistance properties.

3. To provide privacy preserving and privacy control mechanisms within the framework such that there is a harmonization between anonymity, accountability and efficiency. The proposed privacy mechanism must ensure anonymity for the honest users, traceability for misbehaving users and a safe environment for data sharing.

4. To develop formal proofs and perform computer-based simulations and experiments to analyze that the proposed framework is secure and robust against various attacks (watermarking, collusion, anonymity and communication) and is efficient.

1.4 Research Methodology

In order to achieve the mentioned objectives, the methodologies used in this thesis are design and creation and simulations strategy. The design and creation research strategies focuses on developing a new IT product, also called artefact. Types of artefacts include: constructs where the concepts are used in IT-related domain (for example: notion of entities or data flows), models where combinations of constructs that represent a situation are used to aid problem understanding and solution development (for example: a data flow diagram), methods where guidance on the models to be produced and process stages to be followed to solve problems using IT (for example: formal mathematical algorithms) and instantiations where a working system that demonstrates that constructs, models, methods, ideas or theories can be implemented in a computer-based system (Oates, 2005). The privacy-preserving secure framework is a combination of artefacts such as, models, methods and instantiations. The flowcharts, mathematical algorithms, formal proofs and computer-based programming are used to explain the model as well as to aid in the solution development.

The design and creation research strategy is used in an iterative process involving five steps: awareness, suggestion, development, evaluation and conclusion (Oates, 2005).

- *Awareness* is the recognition and articulation of a problem, which can come from studying the literature where authors identify areas for further research, or from new developments in technology.
- *Suggestion* involves a creative leap from curiosity about the problem to offering a tentative idea of how the problem might be addressed.
- *Development* is an implementation of a tentative design.
- *Evaluation* examines the developed artefact.
- *Conclusion* involves documentation of the results of the design.

According to the design and creation research strategy, for the *Awareness* step, it is needed to study the existing P2P content distribution systems that fulfil copyright protection and/or privacy. The literature is studied to understand the problems related to the research that has been carried out in this field. Also, a comparison of recent state-of-the-art P2P content distribution systems is made to analyse the challenges faced by the researchers working in this field. Under the *Suggestion* step, the need to design and develop privacy-preserving and secure frameworks for both copyright and privacy protection is studied. The required concepts and tools are also identified. Based on that, the design and development of the frameworks is carried out under the *Development* step with flowcharts, mathematical algorithms and software codes. Then, the developed frameworks are evaluated under the *Evaluation* step by constructing validation studies through formal proofs, experiments and simulations. To validate the security and privacy features of the frameworks, formal proofs are used, whereas results of simulations and experiments are used to validate the performance of the frameworks in terms of robustness, efficiency and cost reduction. Also, simulations are performed to compare the performance of the frameworks with some recent work in literature. In the *Conclusion* step, the conclusions are deduced for the frameworks from the results obtained though the security and performance analysis. Finally, the future work regarding the design of frameworks for copyright and privacy protection are discussed.

These steps are carried out in an iterative manner, with the cycle being repeated until the desired research objectives have been attained.

1.5 Contributions

In this section, a brief summary of the contributions of this thesis is presented.

- First, a state-of-the art is presented that contains techniques proposed to protect copyrighted content and user privacy. An overview of the existing techniques, such as encryption, digital rights management, digital watermarking and fingerprinting, trust management, pseudo anonymity, anonymous communication and cryptographic techniques is provided.

- Second, a survey is conducted on recent state-of-the-art research works proposing copyright protection and end user privacy in P2P content distribution systems. First, the challenges are identified that P2P systems inherit in terms of security and privacy because of its loose peer management and distributed working principles. Then, a comprehensive overview of existing P2P distribution systems is provided in terms of their features, implementation details and the open problems of these systems. Finally, these systems are compared on the basis of security and privacy properties.

- Third, a Framework for preserving Privacy and Security of User and Merchant based on Homomorphic Encryption (FPSUM-HE) is proposed that facilitates the prevention of content owners' copyright infringement and end users' privacy violation without introducing high computational costs for the merchant. In FPSUM-HE, an asymmetric fingerprinting protocol is proposed for the distribution of copyright content between a merchant (as the content provider) and a number of buyers (as the end users). The protocol employs collusion-resistant codes, a robust and imperceptible embedding scheme, and a homomorphic public-key cryptosystem with restricted usage. The proposed protocol is able to provide all the required security properties, namely traceability, anonymity, unlinkability, dispute resolution, non-framing and non-repudiation, simultaneously. Also, to ensure anonymous communications between buyers of P2P system, onion routing is used for an anonymous data transfer. Moreover, to provide accountability within this framework, a key agreement protocol has been adopted in the scheme. The security properties of asymmetric fingerprinting and anonymous communication protocols are discussed through several attack scenarios. The experimental results confirm that FPSUM-HE provides an efficient, secure and fair solution to copyright infringement and privacy issues over P2P networks.

- Fourth, a Framework for preserving Privacy and Security of User and Merchant with Proxy-based Distribution (FPSUM-PD) is proposed to improve the robustness, computational and communicational costs of FPSUM-HE. The main goals of FPSUM-PD are (1) buyer security and privacy preservation, (2) collusion-resistance, (3) piracy tracing, and (4) efficient distribution of large-sized multimedia content by avoiding the use of cryptographic protocols, such as homomorphic encryption and multi-party secure protocols. In FPSUM-PD, an asymmetric fingerprinting protocol is proposed for distribution of copyright content between a merchant and buyers through untrustworthy proxy peers. The fingerprinted content is distributed to the buyer through the proxy peers in such a way that the merchant does not know about the fingerprinted copy received by the buyer, the proxies are unable to frame an honest buyer, and buyer's privacy is preserved until he/she is found guilty of illegal re-distribution. The protocol employs collusion-resistant codes, a robust and imperceptible embedding technique and symmetric cryptography. The anonymity mechanism deployed in FPSUM-PD is the same as the one used in FPSUM-HE. The security of the distribution and communication protocol of FPSUM-PD under various security compromising attacks is proved through formal proofs. The experimental results reveal that FPSUM-PD provides excellent results with respect to computational and communicational costs compared to FPSUM-HE and existing P2P content distribution systems, which imply that the proposed framework can be implemented in real-world content distribution scenarios.

- Fifth, a comparative analysis of the results obtained from the performance evaluation of FPSUM-HE and FPSUM-PD is presented. The comparison is done in two phases. In the first step, FPSUM-HE is compared with FPSUM-PD in terms of efficiency. Then, both frameworks are compared with existing P2P content distribution systems in terms of guaranteed security and privacy properties. The comparative analysis of both frameworks with existing systems proves that FPSUM-HE and FPSUM-PD provide P2P content distribution mechanisms that handle both merchant's content ownership and user's privacy violation problems in an efficient manner.

1.6 Thesis Organization

The thesis is organized as follows:

- Chapter 2 presents an overview of existing copyright and privacy protection techniques. The purpose of this study is to analyse the strength and limitations of these techniques and to select which techniques are more suitable for our research work.

- Chapter 3 gives a detailed review and comparison of existing P2P content distribution systems that employ copyright and/or privacy protection techniques. The purpose of this review is to analyse the challenges being faced by the P2P researchers in the development of secure and privacy-preserving P2P content distribution systems. The comparison of these systems is carried out on the basis of the protection techniques and the guaranteed security and privacy properties offered by these systems.

- In Chapter 4, FPSUM-HE is proposed, which is aimed at assuring copyright and privacy protection to the merchant and the user in a P2P content distribution system. An exhaustive description of all the framework's components and phases is provided, together with a security and performance analysis.

- Chapter 5 presents a new approach for distribution of copyright content from the merchant to the buyer through untrustworthy parties. It presents FPSUM-PD, a framework for preserving privacy and security of user and merchant based on proxy peers. The entire proposal is analysed to evaluate the security and privacy properties. Also, the performance of the framework is evaluated to compute the computational and communicational overheads introduced by the security and anonymity schemes.

- Chapter 6 provides the comparative analysis of both frameworks with each other and also with some recent P2P content distribution systems in terms of security and privacy.

- The dissertation is concluded in Chapter 7, which summarizes the concluding remarks and also provides discussions on future perspectives that could be developed as extensions of this thesis.

Chapter 2

Content and Privacy Protection Techniques

This chapter presents the state-of-the-art of the main two research lines of this thesis, namely, content protection and privacy preservation. Such background information provides the knowledge needed to design and develop a secure and privacy-preserving content distribution system.

The first part of the chapter describes a basic concept of content protection followed by a detailed explanation of content protection techniques, namely, encryption, digital rights management, digital watermarking and fingerprinting. A brief review of the systems that implement these techniques for multimedia (audio and video) data is also presented. The limitations and research challenges of a viable protection technique are also discussed.

In the second part of the chapter, the concept of privacy is discussed and the privacy issues in content protection are described. An overview of privacy techniques which have been largely used in the recent decades and have been proved to be efficient in resolving privacy issues in content distribution systems are also presented. These techniques are listed below:

1. Cryptographic techniques: Techniques used to provide privacy and remain currently the most used ones.

2. Anonymity techniques: Techniques that protect privacy by making users indistinguishable of other users.

3. Trust techniques: Techniques that provide privacy protection by handling the trustworthiness of the users.

2.1 Content Protection

The introduction and proliferation of P2P systems have facilitated a large scale piracy among the end users of these systems. The exponential rise in piracy has certainly infringed the rights of copyright holders. From first generation P2P system, *Napster* (2011), to third generation, *BitTorrent* (2000), P2P systems are always blamed for illegally sharing copyright content. The legal attempts to alleviate this problem have shown limited success. Thus, this has led the scientific community to focus its interest towards developing content protection techniques to fight against piracy of multimedia content.

Content protection is a generalized term that means restricting access to multimedia content to a user or group of users that are authorized to access the content. From a security perspective, it is to ensure confidentiality, integrity and availability of content, in its distribution, reproduction and use, through content protection mechanisms. The major tasks that the content protection mechanisms are expected to resolve are copy protection, distribution tracing, usage monitoring, authentication of content source and receivers, association of digital rights with content and secure distribution of content and access keys. These are the basic security requirements for an end-to-end content protection system, suggested by Arnold, Schmucker, and Wolthusen (2003). Thus, it is expected from an end-to-end content protection system to ensure not only a legitimate access of a content, but also control the usage of the content once it is in the user's possession.

2.2 Classification of Content Protection Techniques

In this section, an overview of the fundamental techniques and processes for securing multimedia content are presented. Specifically, encryption, digital rights management, digital watermarking and fingerprinting techniques are explored as these are currently seen as the most effective approaches for content protection.

2.2.1 Encryption

Encryption is based on science of cryptography which has been used as long as humans have wanted to keep their information secret. It is probably the most common method of protecting digital content. In simple terms, encryption can be defined as *the process of hiding a message*

in such a way as to conceal its substance. Here, the message is a plain-text, also known as a clear-text, and a hidden message is called a cipher-text. The reverse of an encryption process is called decryption and is defined as *the process of converting cipher-text into plain-text*. The cipher-text is obtained using a cryptographic algorithm also known as cipher. A cipher is a mathematical function that is used for both an encryption and decryption. Data encryption and decryption of data require a key, which is a pre-determined value that determines the functional output of a cipher. Without a key, the algorithm would produce no useful output. An encryption scheme can thus be defined as a procedure of three algorithms: key generation (produces a pair of encryption/decryption keys), encryption (converts a plain-text message into an encoded message using an encryption key) and decryption (converts an encrypted message to an original message using a decryption key).

2.2.1.1 Functional Requirements of an Encryption Algorithm

An encryption scheme is expected to provide one or more of the following properties:

1. **Confidentiality:** It deals with the secrecy of data. Confidentiality refers to limiting data access or disclosure to authorized users and preventing access or disclosure to unauthorized ones.

2. **Integrity:** Integrity addresses the unauthorized alteration of data. This property refers to data that has not been changed inappropriately, whether by an accident or deliberately malicious activity.

3. **Authentication:** It ensures the genuineness of data. Authentication is a property that enables a receiver of data to ascertain its origin.

2.2.1.2 Classification of Encryption Algorithms

The encryption algorithms can be categorized into three types: symmetric, asymmetric (public-key) and hybrid.

1. **Symmetric-key Algorithm:** Symmetric-key algorithms (Schneier, 1996) are algorithms that use the same key, known as a secret key, for encryption of a plain-text and decryption of a cipher-text. The key is kept secret and must be known at sender and receiver end to perform encryption or decryption as shown in Fig. 2.1. Symmetric-key encryption is

effective only if the key is kept secret by the two parties involved. If anyone else discovers the key, it affects confidentiality, authentication and integrity. Symmetric-key algorithms are divided into two categories: stream ciphers and block ciphers.

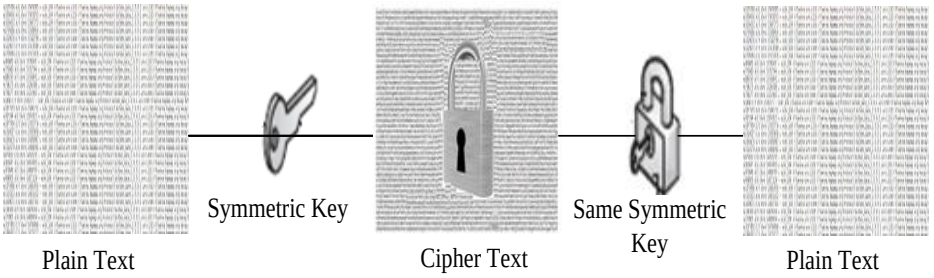


FIGURE 2.1: Symmetric-key encryption

- **Stream Ciphers:** In a stream-cipher encryption algorithm, the input data is encrypted one bit (or sometimes one byte) at a time. This is achieved by combining the plain-text data, bit-by-bit with key bits using an exclusive-OR operation. Some examples of stream ciphers are SEAL, Rivest Cipher 4 (RC4), Word Auto Key Encryption (WAKE), etc. (Schneier, 1996).

- **Block Ciphers:** A block-cipher takes as input a fixed-length group of bits of plain-text, called a block, and transforms into another block (cipher-text) of the same length under the action of user-provided secret key. Decryption is performed by applying the reverse transformation to a cipher-text block using the same secret key. In practice, the vast majority of block-ciphers either have a block-length of 64 bits or 128 bits. Some famous block-ciphers are Data Encryption Standard (DES), Advanced Encryption Standard (AES), Rivest Cipher 6 (RC6), Twofish, etc. (Schneier, 1996). Among these ciphers, AES is used world-wide and has been adopted by the United States of America (USA) government to protect classified information.

Symmetric-key algorithms are fast, easier to implement and require less processing power compared to asymmetric-key counterparts. Although the aforementioned symmetric-key algorithms offer high security and computational efficiency, they also exhibit several drawbacks (Gupta, Agarwala, & Agarwala, 2005):

a) **Key Exchange:** The secret key must be exchanged between the sender and the recipient before transmitting the message. The exchange of a key requires a secure

channel to transport the key generated at one side (sender) of communication channel to the other side (receiver).

b) **Key Management:** A new secret key has to be generated for communication with every new user. This creates a problem of managing and ensuring the security of all the keys.

c) **Non-repudiation:** Symmetric-key encryption does not support non-repudiation. It is possible for one party to falsely claim that it received a message from the other party as it has the key.

d) **Brute-force Attack:** Symmetric ciphers can be cracked though a brute-force attack, in which all possible keys are attempted until the right key is found.

2. **Asymmetric-key Algorithms:** Asymmetric-key algorithms (also called public-key algorithms) not only solve the key-exchange and key-management problems, but they also provide a tool for implementing non-repudiation (Schneier, 1996). Asymmetric algorithms use a different key for encryption and decryption, and the decryption key cannot be easily calculated (within a reasonable amount of time) from the encryption key. The algorithms are called public-key because the encryption key can be made public, but only a specific person with the corresponding decryption key can decrypt the message. The encryption key is called the public key, and the decryption key is called the private key. Fig. 2.2 shows a simplified view of the way asymmetric-key algorithms work. An important characteristics of any asymmetric-key algorithm is that the public and private keys are related in such a way that only the public key can be used to encrypt (decrypt) messages and only the corresponding private key can be used to decrypt (encrypt) the messages. Some public key algorithms provide key exchange (Diffie-Hellman key exchange), some provide non-repudiation and authentication (Digital Signature Algorithm) and some provide all (Rivest-Shamir-Adleman algorithm).

   - **Diffie-Hellman key exchange:** Diffie-Hellman (D-H) key exchange (Schneier, 1996) is a widely used key exchange algorithm. It is not an encryption algorithm; instead, it is a method to securely exchange the keys that are used for encrypting data. D-H accomplishes the task of secure key exchange by generating a shared secret over an insecure communication channel between two communicating parties. It provides security through the difficulty of calculating the discrete logarithm in a finite field.

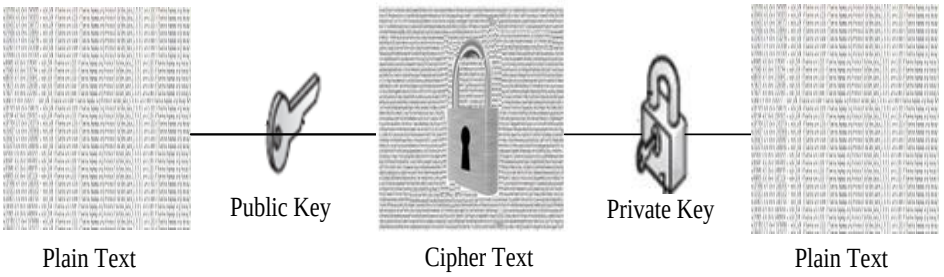


FIGURE 2.2: Asymmetric-key encryption

- **Digital Signature Algorithm:** The Digital Signature Algorithm (DSA) was developed by the USA government for digital signatures (Schneier, 1996). A digital signature is obtained by a private signature algorithm and is verified by a public verification algorithm. It can only be used for signing data and cannot be used for encryption or key exchange. A digital signature uniquely identifies the originator of digitally signed data and also ensures the integrity of the signed data against tampering. DSA signatures can be created quickly, but their verification can take much longer. Its security is based on the intractability of the discrete logarithm problem. Although, the maximum key-size was assumed to be 1024 bits, longer key sizes are now supported.

- **RSA Algorithm:** The Rivest-Shamir-Adleman (RSA) is the most commonly used asymmetric-key algorithm (Rivest, Shamir, & Adleman, 1978). It can be used for encrypting and signing data. To achieve authentication and confidentiality, the sender of the message includes the receiver's identity in the message, signs it using his/her private key, and then encrypts both the message and the signature using the receiver's public key. As compared to DSA, when RSA is used, the process of verifying the digital signature is faster than the generation of a signature. The security of the RSA algorithm is based on the difficulty of factoring of the product of two large prime numbers.

Although asymmetric-key encryption resolves the problems of key distribution and non-repudiation associated to symmetric-key encryption, and offers high security against adversary attacks, it also exhibits a few disadvantages:

a) **Speed:** Asymmetric-key algorithms are slow compared to symmetric-key algorithms. The complexity of the encryption algorithm makes asymmetric algorithms very slow.

b) **Certification Problem:** No party can be absolutely sure that a public key belongs to a person it specifies. Any malicious person can publish his/her public key and masquerade as the intended sender of the data. Many asymmetric-key algorithms use a third party, also known as Certification Authority (*CA*) to certify the reliability of the public key. The certification authority issues a public certificate that certifies the ownership of a public key. However, if the certification authority is compromised, then the security of the entire algorithm is lost.

c) **Key Size:** The key size is typically much larger than the size of the keys required in a symmetric-key encryption algorithm. Asymmetric keys are typically 1024 or 2048 bit long. However, keys shorter than 2048 bits are no longer considered secure. The larger keys can be created only at a cost of increased computational burden and a longer decryption time, e.g. doubling an RSA key length, slows the decryption by 6-7 times and increases the computation time by a factor of 4. Thus, a larger key size has important implications for the practical usage of asymmetric-key algorithms.

3. **Hybrid Encryption:** Hybrid encryption combines the convenience of public-key cryptography with the efficiency of symmetric-key cryptography. Hybrid cryptography can be constructed using any two separate cryptographic systems:

   - A symmetric-key encapsulation scheme, which is an asymmetric-key cryptography.
   - A data encapsulation scheme, which is symmetric-key cryptography.

   Hybrid encryption is achieved by generating a random secret key for a symmetric cipher, and then encrypt this key via an asymmentric cipher using the recipient's public key. The message itself is then encrypted using the symmetric cipher and the secret key. Both the encrypted secret key and the encrypted message are then sent to the recipient. Since the key sharing method is secure, the symmetric key used for the encryption changes for each message sent. For this reason it is sometimes called the session key. This means that if the session key was intercepted, the interceptor would only be able to read the message encrypted with that key. In order to decrypt other messages the interceptor would have to intercept other session keys. The session key, encrypted using the public key algorithm, and the message being sent, encrypted with the symmetric algorithm, are automatically combined into a single package. The recipient decrypts the session key first, using his/her

own private key, and then uses the session key to decrypt the message. Pretty Good Privacy (Schneier, 1996) and Password-based encryption (Kaliski, 2000) are two examples of hybrid cryptosystems.

2.2.1.3 Prior work on Audio and Video Encryption

Depending on the type of plain-text, data encryption schemes are classified as text, audio, image and video encryption. Encrypting the entire multimedia content using standard encryption methods is referred to as the naïve approach. The audio and video data usually are very large in size, which makes the naïve approach computationally demanding. Nowadays, many new algorithms for audio and video encryption have been proposed to avoid the naïve approach and gain better efficiency. In this section, an overview of a few audio and video encryption schemes is presented.

- **Audio Encryption:** Audio encryption algorithms are classified into the following three categories:

  a) **Fully Layered Encryption:** In this class, the whole content is first compressed and then encrypted using standard encryption algorithms, such as AES, DES, etc. However, this technique is not suitable for real-time applications due to heavy computation. Gnanajeyaraman, Prasadh, and Ramar (2009) proposed an audio encryption scheme based on a look-up table which is generated by using higher dimensional chaotic map. The experimental results show that the scheme has the characteristic of high key space, audio signal uniformity and resistance to brute-force and chosen/-known plain-text attacks. However, the computational complexity of the scheme is very high and cannot be implemented in real-time applications.

  b) **Partial Encryption:** The algorithms in this category encrypt only a selected part of the audio data, while leaving other parts unchanged. Since the whole file is not encrypted, the encryption process is faster. Servetti, Testa, and Carlos de Martin (2003) proposed a low-complexity scheme based on partial encryption for protection of MPEG-1 layer 3 (MP3) audio. Their proposed encryption algorithm employs low-pass filters in the compressed domain to limit the frequency content of an audio data. The resulting bit-stream can be decoded without an error by any MP3 standard decoder. Moreover, the cut-off frequency is modified by increasing or decreasing the

number of coefficients. The result shows that low-pass filtering at 5.5 KHz preserves the audio contents.

c) **Perceptual Encryption:** Perceptual encryption is a type of controlled encryption process which degrades the quality of the content depending on the requirements. To carry out this encryption, the sensitive parameters of the content are extracted out and some of them are then encrypted. A perceptual quality based MP3 encryption was suggested by Torrubia and Mora (2002), where the Huffman code bits were modified by substituting the Huffman codeword by another codeword of the same size and then encrypting with a pseudo-random bit-stream. However, the proposed method is complex and vulnerable to brute-force attacks.

- **Video Encryption:** Video encryption algorithms can also be divided into three categories:

a) **Fully Layered Encryption:** It is a traditional approach for content access control. Initially, the data is encoded with a standard compressor and then full encryption is applied to the compressed bit-stream with a standard cipher (DES, AES, RC6, etc.). S. Li, Zheng, Mou, and Cai (2002) proposed a chaotic video encryption scheme (CVES) for a digital video based on multiple digital chaotic systems. In CVES, each plain-text block is first XORed by a chaotic signal and then substituted by a pseudo-random S-box based on multiple chaotic maps. The CVES is secure against brute-force and known/chosen-plain-text attacks. The security depends on the proposed chaotic cipher and, as long as the cipher is well-designed, it provides higher security, but at a cost of high complexity.

b) **Partial Encryption:** The algorithms in this category selectively encrypt the bytes within the video frames. Since, all the bytes of the video are not encrypted, it reduces computational complexity. Lian, Liu, Ren, and Wang (2006) proposed a video encryption scheme based on Advanced Video Coding (AVC), which utilizes intra-prediction mode and includes a sign-bit encryption of non-zero transform coefficients and motion vectors. However, this scheme causes a high computational cost since each non-zero coefficient needs one random bit and the number of non-zero coefficients in a frame is very large.

c) **Perceptual Encryption:** Perceptual encryption requires that the quality of video data is partially degraded by encryption, i.e. the encrypted multimedia content is

still partially perceptible after encryption. Lian, Sun, and Wang (2004) proposed a perceptual encryption algorithm for 3D Set Partitioning in Hierarchical Trees encoded videos. In their scheme, a video is degraded to different degrees by confusing a different number of wavelet coefficients and encrypting different number of coefficients' signs. Its encryption strength can be adjusted according to a certain quality factor. However, the proposed scheme is not secure against known chosen plain-text attacks.

2.2.1.4 Limitation of Encryption

Encryption can be used to package the content securely and enforce all access rules to the protected content. The encryption techniques can protect the multimedia content during its transmission from the sender to the recipient by scrambling the content and making it unintelligible unless a decryption key is known. However, once an authorized user has decrypted the content, it does not provide any protection to the decrypted content. After decryption, the content can be perfectly duplicated, manipulated and re-distributed at a large scale. Thus, encryption alone is not enough to prevent an authorized user from copying and re-distributing illegal copies of the content.

2.2.2 Digital Rights Management

Digital Rights Management (DRM) systems have been developed to manage the content distribution and protect the rights of the content provider against the malicious actions of legitimate users. DRM allows content providers to specify their own business model in managing the use of the content, such as time-limited use of content, subscription, multiple views of a video, and restrictions on transferring a song to a portable device. A DRM system operates on three levels: establishing a copyright for a piece of content, managing the distribution of that copyrighted content and controlling what a consumer can do with that content once it has been distributed. To accomplish this level of control, a DRM system has to effectively define and describe three entities: the user, the content and the usage rights and also the relationship between them.

DRM's core technologies for combating piracy can be categorized as cryptographic-based and watermarking-based mechanisms. They include encryption, passwords, watermarking, digital signature and payment systems. Encryption and password technologies are used to control who has access to the content and how it is used. Watermarks and digital signatures are used

to protect the authenticity and integrity of the content, the copyright holders and the user. The generic DRM architecture (Arsenova, 2002) consists of three entities: content provider (*CP*), license provider (*LP*) and a user as shown in Fig. 2.3. *CP* is mainly in charge of generating the multimedia content and protecting it by encrypting it using well-known encryption algorithms. *CP* also generates meta-data, which contains some useful information such as the place where to get the encrypted content, which algorithm to decrypt the content and where to obtain the decryption key. *CP* attaches the meta-data with the encrypted content. The meta-data guides the consuming device, or an application of the user, to the location of *LP*. *CP* provides *LP* with the corresponding content encryption keys (CEKs). *LP* is mainly responsible for creating permissions (licenses), which include terms and conditions, as well as managing the CEK for enabling the consuming device or application to expose the corresponding hidden content. The user downloads the hidden content via local software, called a DRM agent (*DA*) or renderer, which is designed to enforce usage policies. The renderer extracts the information from the meta-data, negotiates with *LP* for providing licenses according to the user's payment amount, downloads the license, checks the integrity and the validity of the license, extracts the CEK and enforces the terms and conditions. The license file required for completion of rendering process of the content must be paid for. Therefore, controlling and managing the license helps the content owners in terms of profit.

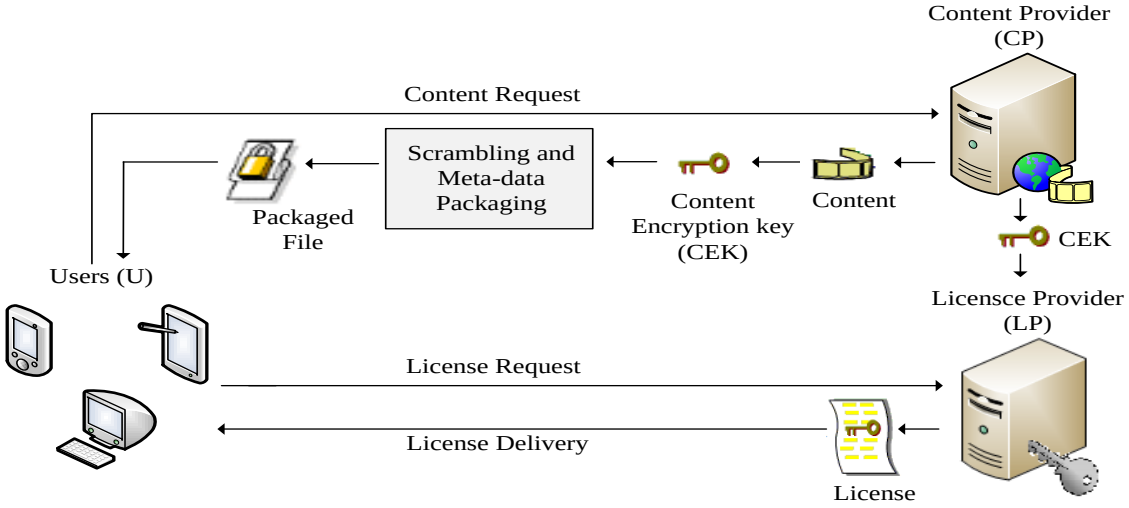


FIGURE 2.3: Digital Rights Management system

2.2.2.1 Security Requirements of Digital Rights Management System

The following are security properties (Jonker & Mauw, 2004) that form a solid foundation for the core functionality of DRM systems. These properties are necessary for DRM systems.

- The desired security property for CP is the following:

  1. Content is only accessible by untampered components created by the official system developers, under the conditions of a valid license issued by a *bona fide LP* for those components.

- The security properties required for LP are the following:

  1. The content is only accessible by a renderer with a valid license issued to that renderer, originating from the license creator, under the terms stated in that license.

  2. Precisely deliver what has been requested, in a consumable form, at the desired time for the licensee.

- The security requirements for the users are the following:

  1. Precisely acquire a consumable form of the content that the user desires, at the moment the user desires it.

  2. To order licenses or content on the user's behalf requires the intentional participation of the user.

  3. Neither content nor licenses can be linked to the user.

  4. The user is aware of all negotiations resulting in an agreement between him/her and LP, and consents to the terms of any such agreement.

2.2.2.2 Types of DRM Systems

There are two types of DRM protection systems: hardware-based and software-based.

1. **Hardware-based DRM Systems:** Hardware-based protection is intended to protect software programs from piracy and tampering. Two main examples of hardware-based DRM protection systems are: smart cards and Trusted Computing Platform Alliance (TCPA).

- **Smart cards:** The smart card system is an integrated circuit used as a portable token that embeds a secure crypto-processor, a random access memory (RAM) and a secure file system to protect cryptographic data such as a secret key. The design of the smart card is considered proprietary, and the secure file system contains private information about the user for identification and authentication purposes (Selimis, Sklavos, & Koufopavlou, 2004). Smart phones with *Open Mobile Alliance* (2002) (OMA) DRM2.0 represents an example of the successful use of hardware-based DRM protection systems. The system has been implemented on many recent phones. DRM agents, as described in the OMA DRM architecture specification embed unique private/public key pairs and certificates, which are used to identify and authenticate mobile devices and to individualize the acquired right objects for that device.

- **Trusted Computing Platform Alliance:** *Trusted Computing Group* (2003) provides a specification for trusted computing environments and protocols that are composed of trusted hardware, BIOS, trusted OS kernel, self encrypting storage, and trusted anti-virus software. TCG specification provides three access privileges: privileged access (TCPA members only), underprivileged access (platform owner) and unprivileged access (non-TCPA applications). In the TCG, the following components are essential for enforcing DRM usage rights and security policies: cryptographic operations (such as public and secret key encryption), key store, key management and secure booting process.

2. **Software-based DRM Systems:** Software-based protection needs to be individualized in order to prevent it from working on more than one device. For example, each instance of the *Apple's FairPlay DRM Copy Protection* (2001) player embeds the hardware information of the device that is supposed to launch it; this is called individualization via binding hardware information. Microsoft media rights manager is another example of individualization via binding hardware information in which the Windows player uses dynamic link library (DLL) files, which are individualized for the distinct player that is supposed to run on a specific computer (A DLL file contain functions and resources that allow Windows-based programs to operate in the Windows environment). The individualization process is achieved by generating a unique DLL file that is embedded with the computer hardware's unique identifier and a private key. When the license provider issues a license to a particular computer, it is encrypted with the related public key. Thus, the only machine that can

use the license is the one with the right private key (Q. Liu, Safavi-Naini, & Sheppard, 2003). The license provider, in turn, individualizes any acquired license by encrypting the media key with a specific DRM player's public key and then embeds the encrypted media key within the license. This process is called individualization via binding certificate (C. L. Chen, 2008). The advantage of binding a license to a unique player is that it prevents the license from being transferable. The individualization process gives the content provider the power to make the digital content work under specific individualized DRM components. Star-Force is an example of software-based DRM protection systems:

- **Star-Force:** *Star-Force* (2000) is a professional copy protection software designed to discourage software piracy. Star-Force is well-known by gamers for its invasive techniques which can cause problems such as optical drive failure. There are several different variations of Star-Force and each is designed to protect content at different levels.

2.2.2.3 Prior Work on Audio and Video DRM

In this section, an overview of DRM systems for audio and video data is presented.

- **Audio DRM:** Serrao et al. (2006) proposed a method for protecting an MP3 file and its integration within a DRM platform to provide a new service called MediaBox. MediaBox allows the access and management of DRM-protected content. The presented protection method uses AES (in Output Feedback cipher mode) to protect the audio data while maintaining the bit-stream structure intact. A protection tag named digital object rights management has been defined and added to the file format to allow the inclusion of encryption parameters inside the MP3 file format. The method does not propose to create a new file format or a new file container that would require specific modifications to the current established players.

- **Video DRM:** Cheng et al. (2011) proposed a DRM system based on Fast Forward Moving Picture Experts Group *FFmpeg* (2000). The proposed DRM system includes a converter, a player, a content server and a license server. The scheme protects the video content by encrypting only the packets of intra-frames (key-frames) of the video. The video frames are arranged into groups of pictures (GoPs). A GoP includes the intra frames (I-frames) and inter-frames (P and B-frames). The key-frames carry a complete video picture. These are

coded without reference to other frames, whereas P and B-frames use pseudo-differences from the previous and next frame. In the proposed system, first the data is encoded by the converter. If the frame is a key-frame, the converter encrypts its packets using XOR encryption and then written to the file. The converter gets the encryption key from the license server when encrypting the key-frames. When using the media player to play the video, the encrypted packets of key frames are decrypted at first, and then decoded. If the frame is a B or a P-frame, the player decodes and plays the files directly. The media player is designed using *Simple DirectMedia Layer* (1998) and FFmpeg libraries. Other players can not open the encrypted video files.

2.2.2.4 Research Challenges

The emerging problem is that most DRM systems are neither standardized nor interoperable. In general, each content provider has its own technique and model to protect digital content, with little or no regard for its interoperability with other DRM systems. As a result, consumers often find they cannot render the digital content they have purchased on the device of their choice. This non-interoperability can cause dissatisfied customers to avoid DRM systems in the future and, consequently, look for other options to acquire the content, e.g. P2P file-sharing systems. Device manufacturers or application developers can choose to either integrate a single DRM technology, thereby limiting the flexibility of their devices and applications, or implement multiple DRM technologies adding to the cost of their devices and applications.

A typical DRM system typically provides means for protecting content, creating and enforcing rights, identification of users and monitoring of the content usage. However, this level of protection and control often leads to severe tension between copyright owners and users because the users' freedom is greatly affected (Y. Sun, 2014). These systems are privacy-invasive as they violate users' privacy in a number of ways: they do not support anonymous and un-linkable buying or transfer of content and keep track of the usage of content in order to keep control over the content. In an increasingly privacy-aware world, such possibilities of creating user profiles or tracking users create numerous privacy concerns. Thus, with privacy-preserving and interoperable DRM architectures, content providers can potentially reach a wider audience because their content can be accessible by any compliant device or application and will not violate users' privacy.

2.2.3 Digital Watermarking

For many content owners, DRM is expensive, and it is not an effective access-control technique. Also, legitimate consumers are frustrated by the overly restrictive technology, which prevents them from easily sharing content between devices or applications. For content owners, digital watermarking proves to be a more effective anti-piracy solution than DRM technologies. Digital watermarking has become a significant area of research and development and the usage of these techniques is now being considered a requisite to address the issues faced by the proliferation of digital content. Watermarking schemes work by embedding a content provider (merchant) specific mark (watermark) imperceptibly, which upon extraction enables provable ownership. Multimedia encryption is often combined with digital watermarking to protect certain properties of the multimedia content, such as ownership authentication, copyright protection, etc.

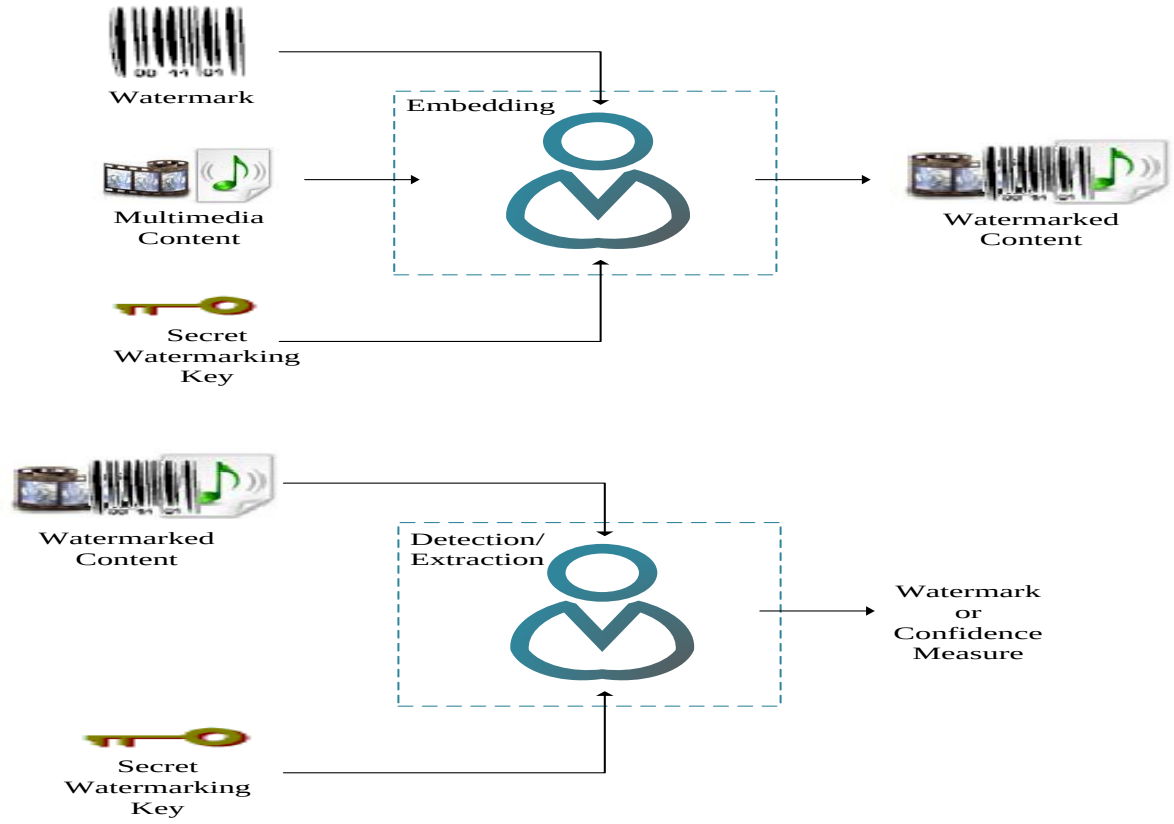


FIGURE 2.4: Digital watermarking

A watermarking system is usually divided into two distinct steps: (1) watermark embedding to indicate copyright, and (2) watermark detection/extraction to identify the owner. Fig. 2.4 shows the basic block diagram of the watermarking process.

- **Watermark Embedding:** Embedding is an algorithm to which a watermark (customer Id, or unique text or an image), a multimedia file (image, audio or a video file), and optionally the secret key (possessed by the owner) are given as inputs. The output of the algorithm is a watermarked copy containing that particular watermark.

- **Watermark Extraction/Detection:** A watermark must be extractable or detectable to prove/verify the content ownership. In order to extract/detect the watermark embedded into the content, a watermarked content and a decoding algorithm are required. The decoder takes the watermarked content and possibly a secret key to detect/extract the hidden watermark. The extraction of the watermark can be divided into two phases: (1) locating the watermark, and (2) recovering the watermark information. A watermarked detection algorithm, on the other hand, provides a measure to indicate whether or not a specific given watermark is present in the content. The watermark detection can only verify ownership whereas watermark extraction can prove ownership.

2.2.3.1 Functional Requirements of Digital Watermarking

Digital watermarking systems can be measured on the basis of certain properties that depend on the application. These properties include the difficulties of notice (imperceptibility), the survival of common distortions and resistance to common signal processing attacks (robustness), the capacity of bit information (payload) and the security of the watermarking method. Each of these properties must be taken into consideration when applying a certain watermarking technique (Cox et al., 2007).

1. **Imperceptibility:** A watermark must be embedded into the content such that no obvious difference in the content fidelity can be noticed. Cox et al. (2007) define imperceptibility as *perceptual similarity between the original and the watermarked versions of the digital content*. The embedded watermark must be transparent and must not introduce distortion, which can cause quality degradation. A watermarking scheme which is not imperceptible is not suitable for high fidelity applications.

2. **Robustness:** Robustness against attacks is a key requirement of watermarking. According to Cox et al. (2007), robustness is defined as *the ability to detect the watermark after common signal processing operations*. A watermark must be robust enough to withstand all kind of signal processing operations (at least below some distortion threshold). Some

of the common signals processing attacks are cropping, compression, scaling, additive noise and filtering. For robust watermarking, any attempt to remove or destroy a watermark should result in severe quality degradation of the data before the watermark is lost or becomes undetectable.

3. **Capacity:** Capacity or data payload is defined as *the number of bits a watermark encodes within a unit of time (or space in the case of still images)* (Cox et al., 2007). Capacity is usually given in bits per pixel for images and bits per second for audio. The required watermark payload varies greatly from application to application.

4. **Security:** Different from the robustness property, the watermark security refers to the ability to resist intentional or hostile attacks. A watermarking algorithm must be secure in the sense that an attacker must not be able to detect/extract the existence of embedded data, let alone remove the embedded data. Watermark information should only be accessible to the authorized parties. In cryptography, Kerckhoff's assumption states that one should assume that the method used to encrypt the data is known to an unauthorized party and that the security must lie in the choice of a key (Barni & Bartolini, 2004). Thus, in the context of watermarking, it implies that it should be difficult for an attacker to remove or forge a watermark without the knowledge of the proper secret key even if the watermarking algorithm is publicly known.

2.2.3.2 Classification of Digital Watermarking

Digital watermarks can be sub-divided into various categories. For example, they can be classified according to the applications, human perception and techniques.

1. **Watermarking Applications:** The following are few applications of digital watermarking:

   - **Copyright Protection:** This is the most prominent application of digital watermarking. The aim is to evade other parties from claiming the copyright by embedding the information such as a logo that identifies the copyright owner of the multimedia data. This application is of great interest to big multimedia organizations.

   - **Digital Fingerprinting:** Unlike copyright protection application in which the same watermark is embedded in all the copies of content, in fingerprinting applications, a

unique fingerprint is embedded in each individual copy of the content. A fingerprint is a type of a watermark that identifies the recipient of a multimedia content (i.e. a serial number assigned by the content provider to a given buyer). This application acts as a deterrent to illegal re-distribution by enabling the owner of the content to trace the source of the re-distributed copy.

- **Content Authentication:** Another application of watermarking is the authentication of multimedia content. The goal of this application is to provide assurance that the origin of the content is authentic and its integrity can be proved. Effective authentication enables the owner to reliably authenticate data and identify possible tampering of the content.

2. **Types of Watermark:** Digital watermarks can be divided, on the basis of human perception, as follows:

   - **Visible Watermark:** The embedded watermark, which can be a text or a logo, is detectable to human eye. It is used to identify the content owner and prevents unauthorized use by reducing the commercial value of the content.

   - **Invisible Watermark:** The watermark is embedded into the content in such a way that it cannot be perceived by an human eye. It is used to provide content authentication and prevent it from being copied.

3. **Watermark Extractability:** A watermark can be differentiated in accordance to the security requirements. These can be blind or non-blind.

   - **Non-blind Extraction/Detection:** Non-blind watermarking techniques extract/detect the embedded watermark by comparing the watermarked content with the original unmarked content. These schemes are robust but are not practically usable in many applications.

   - **Blind Extraction/Detection:** If the extraction/detection of the digital watermark can be done without the original data and original watermark, such a technique is called blind extraction/detection. Blind methods are more useful than non-blind counterparts because the original content may be not available in real-world scenarios. However, blind extraction/detection methods tend to be less robust and harder to implement than non-blind ones. Currently, most researchers are focusing on blind watermarking techniques rather than non-blind counterparts.

2.2.3.3 Embedding Domains and Techniques

Recent years have witnessed a flood of novel watermarking schemes and techniques to prevent copyright infringement. Among these, spread spectrum (SS) and quantization schemes are the two most appealing watermark embedding techniques. Using any of these two schemes, a watermark can be inserted into a spatial or frequency domain of the content to produce a watermarked copy. In this section, a classification of watermarking algorithms with respect to embedding domains and techniques is presented.

- **Classification of Embedding Domain:** To embed a watermark in a digital content, watermark embedding techniques apply minor modifications to the data content in a perceptually invisible manner, where the modifications are related to the watermark information. The watermark can be retrieved afterwards from the watermarked data by detecting the presence of these modifications. Prior to embedding or extracting a watermark, the content can be converted, e.g. to the transform domain or remains in time domain. Watermarking schemes based on their processing domain, can be divided into two categories: Spatial/time domain watermarking and frequency domain watermarking.

  1. **Spatial/Time Domain:** The spatial/time domain watermarking techniques make use of content attributes (pixels in case of images, amplitude in case of audio, and luminance/chrominance components in case of video) in the spatial/time domain to embed a watermark. There are two major types of spatial techniques: Least Significant Bit (LSB) substitution and the patch-work method.

     - **LSB:** The LSB technique is a frequently used method. It can be applied to any form of watermarking (image, audio and video). For example, in image watermarking (Van-Schyndel, Tirkel, & Osborne, 1994), the LSB of the original image is substituted with the watermark bit. The bits are embedded in a sequence which acts as the key. In order to retrieve the watermark, this sequence should be known.

     - **Patch-work:** This method is based on a pseudo-random statistical model. It works by invisibly embedding a specific statistic, with a Gaussian distribution, into the content. For example, in case of an audio, the watermark embedding process uses a pseudo-random process to insert a certain statistic into an audio

signal, which is extracted with the help of numerical indexes, describing the specific distribution (Bender, Gruhl, Morimoto, & Lu, 1996).

Spatial/time domain watermarking techniques are robust to cropping and translation. However, they are less robust to lossy compressions, such as JPEG and MP3. Generally, these methods are often not robust to signal processing attacks, although they are efficient as complexity is concerned.

2. **Frequency Domain:** Compared to spatial/time-domain watermarking, watermarking in the frequency domain is usually more robust. Thus, frequency domain watermarking obtains much more attention and many techniques are proposed based on frequency domain transforms. In transform domain watermarking, the original data is first transformed using any transformation technique and then the modifications are applied to the transformed coefficients according to the watermark information. The most commonly used transforms for the purpose of copyright protection are: the Discrete Cosine Transform (DCT), the Discrete Fourier Transform (DFT) and the Discrete Wavelet Transform (DWT).

   – **DCT:** It is a process which converts a sequence of data points of the content in the spatial domain to a sum of sine and cosine waveforms with different amplitudes in the frequency domain. Watermarking in DCT domain is popular, since DCT serves as a transform coding module in image and video coding standards, including JPEG, MPEG-1 and H.264. For example, image-based watermarking in the DCT domain is usually performed on the lower or the mid-band frequencies, as higher frequencies are lost when the image is compressed. DCT watermarking can be done for an entire image taken together (Cox & Linnartz, 1998) or block-wise (Petitcolas, Anderson, & Kuhn, 1998).

   – **DFT:** The difference between DCT and DFT is that DFT applies to complex numbers, while DCT uses just real numbers. For example, the DFT (Lian, Kanellopoulos, & Ruffo, 2009) coefficients of an image are complex numbers that consist of a magnitude and a phase.

   – **DWT:** In the last few years, the DWT has become researchers focus for digital watermarking. DWT-based watermarking methods (Tsai & Chen, 2000) exploit the frequency information and spatial information of the transformed data

in multiple resolutions to gain robustness. Multi-resolution analysis is the characteristic of DWT, which is a better simulation for the human auditory system. For example, DWT-based audio watermarking decomposes the audio signal into time domain and frequency domain by different scales corresponding to different frequency ranges. Then the characteristics of time-frequency localization and hierarchal decomposition can be used to embed the watermark.

- **Classification of Embedding Techniques:** Digital watermarking schemes can be classified on the basis of embedding methods. The two most commonly used schemes for embedding a watermark are the Spread-Spectrum (SS) method and the Quantization Index Modulation (QIM).

  1. **Spread Spectrum:** In this method, a watermark is inserted into the spectral components of the data using techniques analogous to spread spectrum communications, i.e. a narrowband signal is spread across a signal of much larger bandwidth. The total energy of the narrowband signal at any particular frequency is very low, and thus it is imperceptible to an observer. The retrieval of the watermark unambiguously identifies the owner and the watermark can be constructed to make counterfeiting almost impossible (Cox et al., 1997). SS is robust to common signal and geometric distortions such as digital-to-analog and analog-to-digital conversion, resampling, quantization, dithering, compression, rotation, translation, cropping and scaling. However, the SS scheme has the disadvantage that it requires the original digital content in the extraction process. As a result, only the owner can extract the watermark.

  2. **Quantization Index Modulation:** QIM is a relatively recent watermarking technique. It has become popular because of the high watermarking capacity and the ease of implementation. The basic QIM scheme embeds a watermark bit w by quantizing a single-signal sample x by choosing between a quantizer with even or odd values, depending on the binary value of w (B. Chen & Wornell, 2001). Some drawbacks of basic QIM watermarking are its sensitivity to amplitude scaling attacks and embedding positions can be retrieved from a single copy. A common solution to the later problem is to add pseudo-random noise, usually called dither, to x before embedding an information bit w and subtracting the dither after embedding. This scheme is called subtractive-dither QIM (SD-QIM) (Shterev & Lagendijk, 2006).

An extension to SD-QIM scheme is Distortion-Compensated QIM (DC-QIM). In this, a fraction ($\alpha \times x$) is used in the SD-QIM procedure. DC-QIM provides a significant improvement in robustness compared to the basic QIM. However, DC-QIM is known to be very sensitive to gain or volumetric attacks. The solution to this problem was given by Perez-Gonzalez et al. (2005), where the usage of QIM was proposed on ratios between signal samples so as to make the watermarking system robust against fixed-gain attacks. This process is called Rational-Dither Modulation (RDM). RDM is robust against both additive-noise and fixed-gain attacks.

2.2.3.4 Prior work on Audio and Video Watermarking

The watermarking algorithms can be classified into three types according to the type of media content: image, audio, and video watermarking. During the past few years, a large number of watermarking schemes have been proposed. Most of the proposed watermarking schemes focus on image and video watermarking and very few focus on audio. In this section, an overview of some audio and video watermarking schemes is presented.

- **Audio Watermarking:** In the following, audio watermarking schemes based on different embedding domains and watermarking techniques are discussed.

  a) **Time Domain:** A robust and blind audio watermarking scheme is presented by Bassia, Pitas, and Nikolaidis (2001) in the time domain. The scheme is based on direct modification of the amplitude values in such a way that it does not produce any perceptual difference. A watermark key is used for generating the watermark to be embedded into the audio signal. The proposed scheme is statistically imperceptible but this imperceptibility is affected if high amplitude values of the watermark bits are used to increase the robustness.

  b) **DCT Domain:** Foo and Dong (2010) proposed an audio watermarking scheme based on the DCT. In this approach, the host audio signals are segmented into frames. Two consecutive frames are assessed, if they are suitable to represent a watermark bit. The proposed scheme adopts a compression-expansion technique to generate a special waveform over two consecutive frames. The authors also applied a psychoacoustic model to calculate a local auditory mask to ensure that the distortion caused by watermarking is not audible. Moreover, it is shown that the design

of the watermarking schemes for mono and stereo audio signals is different. To detect the distortion and extract the embedded watermark bits, the correlation-based detection method is used.

c) **DFT Domain:** Fallahpour and Megías (2009) proposed a high capacity audio watermarking algorithm based on spline interpolation. The proposed algorithm is based on the difference between the original and the interpolated amplitudes of the DFT samples obtained by spline interpolation. A sample is selected for embedding the watermark if the difference between the original and the interpolated amplitude of the DFT sample is lower than a given fraction of the interpolated value (α). To obtain the marked DFT samples, the interpolated value is changed according to the secret bit.

d) **DWT Domain:** Bhat, Sengupta, and Das (2011) presented a blind audio watermarking algorithm based on the DWT and single value quantization. First, the audio signal is divided into non-overlapping frames of 2048 samples each. Then, the DWT is applied to each frame and the maximum value in each frame is selected. The watermark bit is embedded by quantizing the selected maximum values. The watermarked signal is obtained by applying inverse DWT for each frame and reconstructing them.

- **Video Watermarking:** Following is a brief overview of the prevailing literature in watermarking of video for copyright protection.

a) **Spatial Domain:** Lancini, Mapelli, and Tubaro (2002) proposed a spatial domain video watermarking scheme on the uncompressed domain and checked the robustness against compression, cropping and resizing attacks. They used convolution and turbo codes for the improvement in the robustness of the algorithm. The spread spectrum approach is used to generate a mask and this mask is added to the original video to obtain the watermarked video.

b) **DCT Domain:** A watermarking scheme is proposed by C. Wu et al. (2010) to protect the copyright of H.264/AVC videos. The Practical Swarm Optimization method has been employed to find the optimal frequency bands for watermarking in the DCT-based system. This method applied to improve imperceptibility and robustness through finding the balanced bands between low-frequency and high-frequency

bands. Dither-modulation is used as the embedding technique. The video frames are decomposed to macro-blocks and the DCT is adopted for each macro-block. The quantized coefficients are then zigzag re-ordered and the embedding process continues by modifying the quantized integer DCT coefficients of the Intra-frame intensity components.

c) **DFT Domain:** D. He, Sun, and Tian (2003) present a watermarking algorithm based on the DFT transform that selects a group of DFT coefficients in the low-middle frequency band and the coefficients in every group are divided into two sub-groups based on a pre-defined pattern. Then, the energy relationship between these two sub-groups is used to hide the watermark.

d) **DWT Domain:** Hussein and Mohammed (2009) proposed a robust method using the DWT and a motion estimation algorithm. The authors chose the horizontal detail and vertical detail components of the video frame so as to embed the watermark because the motion in these bands does not affect the quality of the extracted watermark. The watermark is embedded in an additive way using random Gaussian distribution in video sequences.

2.2.3.5 Research Challenges

Researchers have proposed various watermarking schemes to protect the ownership of multimedia content. However, it is hard to satisfy all the demands simultaneously. The major research problem that researchers face is achieving a trade-off between robustness, capacity and imperceptibility properties of watermarking schemes. These essential watermarking properties contradict one another, i.e. if one is increased, the other decreases. For example, if a watermark capacity is increased, it affects the fidelity (perceptual similarity) of the content and if the watermark payload (capacity) is decreased, the robustness of the system is usually increased. Thus, it is very important for researchers to achieve a convenient trade-off between the above mentioned properties according to the application requirements.

2.2.4 Digital Fingerprinting

In digital watermarking, an embedded watermark can identify who owns a specific content, normally by the use of an embedded logo or copyright text. The watermarking algorithm can be

used to prove content ownership but it is unable to deal with content leakages, i.e. cases where a user may re-distribute the received content to other unauthorized users. This implies that digital watermarking is capable of determining the copyright of multimedia content, but incapable of tracing back the source of leak. This deficiency of watermarking scheme inspires a lot of research efforts in digital fingerprinting.

Associating unique information about each distributed copy of digital content is called fingerprinting (also known as transaction tracking). Thus, if an unauthorized copy of the content is recovered, extracting the fingerprint will show who the initial receiver was. Digital fingerprinting is an emerging technology that gives content owners and publishers more options to control the distribution of their content. The fingerprints are typically embedded into the content using watermarking techniques that are designed to be robust to a variety of attacks. The fingerprinting techniques of multimedia contents involve the generation of a fingerprint, the embedding operation and the realization of traceability from re-distributed copies. It refers to the complete protocol between a content provider (merchant) and a user (buyer). Fig. 2.5 presents a block diagram of digital fingerprinting.

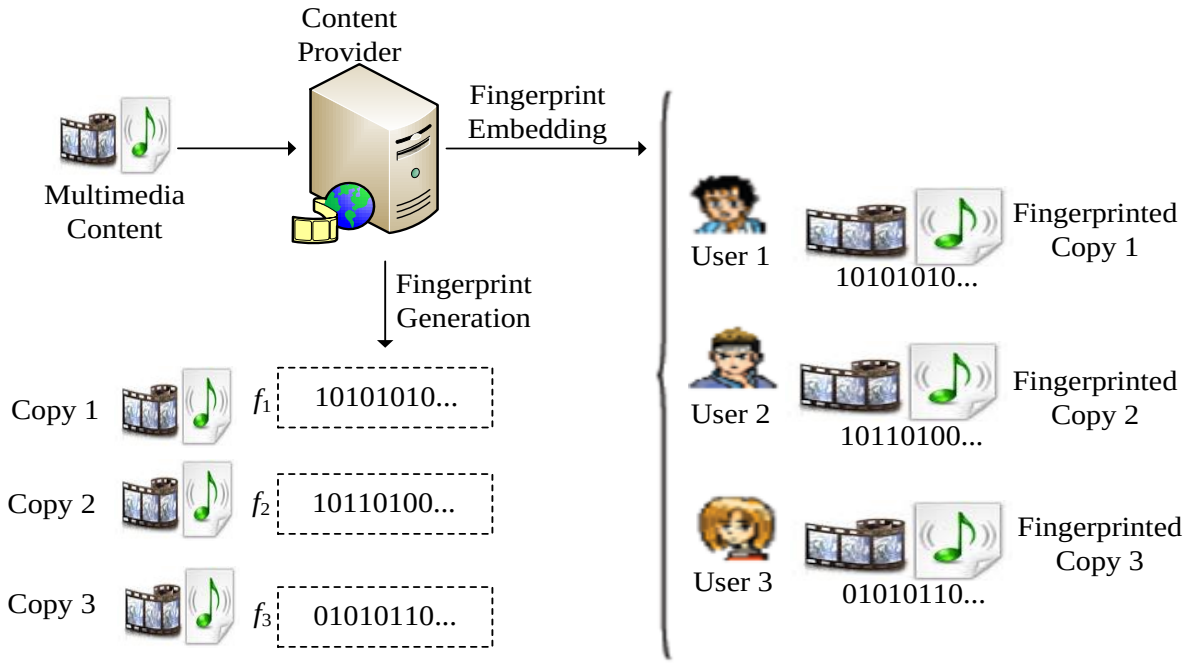


FIGURE 2.5: Digital fingerprinting

2.2.4.1 Functional Requirements of Digital Fingerprinting

There are certain constraints that are to be resolved for designing an effective fingerprinting scheme (Katzenbeisser & Petitcolas, 2000). These are as follows:

1. **Robustness:** A fingerprint's robustness against signal processing operations such as compression, additive noise, filtering, etc. is determined by the adopted watermark embedding method. Thus, a robust watermarking algorithm must be adopted so that the fingerprinting scheme can trace an illegal re-distributor after the digital contents have been manipulated by common signal processing attacks.

2. **Collusion resistance:** While digital fingerprinting may be effective at identifying single adversaries, multiple malicious buyers may collaborate to launch powerful collusion attacks against the fingerprinting system. By comparing their different versions, the colluders can attempt to identify the locations containing the fingerprint signal, remove the information from these locations and thereby create a copy that cannot be traced back to any of them. Thus, a fingerprinting scheme must be designed to withstand such collusion attacks.

3. **Quality Tolerance:** Fingerprinted content should have good visual quality and perceptual similarity to the original content.

4. **Embedding Capacity:** The capacity determines the length of fingerprint allocated to each user. The fingerprint is a binary string that can be of a large length. Therefore, digital fingerprint system should have a large enough embedding capacity to accommodate a full fingerprint.

2.2.4.2 Types of Digital Fingerprinting

Traditionally, in fingerprinting schemes, it was assumed that content providers are trustworthy and always perform the watermark embedding honestly. However, in practice, such assumption is not fully established. Fingerprinting schemes can be classified in three different categories:

1. **Symmetric Fingerprinting:** A symmetric scheme (Z. J. Wang et al., 2005) is a traditional fingerprinting approach in which the merchant is solely responsible for generating and inserting the fingerprint. As a consequence, when an illegal copy is found, the merchant cannot prove to a third party that this copy is indeed distributed by a malicious buyer. This

is because the buyer can claim that he/she is framed by the merchant. This implies that it is possible for a malicious merchant to frame an innocent buyer, or for an accused buyer to repudiate the guilt. K. J. R. Liu et al. (2005) identified this problem as the "customer's rights problem" and this problem can be solved by designing asymmetric fingerprinting schemes.

2. **Asymmetric Fingerprinting:** In asymmetric schemes (B. Pfitzmann & Schunter, 1996) only the buyer obtains the exact fingerprinted content, and hence the buyer cannot claim that a pirated copy was originated from the merchant. In these schemes, the embedding is performed using a protocol designed in such a way that only the buyer obtains the fingerprinted copy of the content. This makes it possible to prove the illegal re-distributor's treachery to a third party but the problem is that the buyer needs to be authenticated by the seller at each purchase. Consequently, the merchant knows the buyer's identity even if the buyer is honest. Thus, it is desirable for buyers to be capable of purchasing fingerprinted digital items anonymously and remain anonymous as long as they do not distribute the digital contents illegally. This problem was solved by Qian and Nahrstedt (1998) by introducing anonymous fingerprinting.

3. **Anonymous Fingerprinting:** Anonymous fingerprinting schemes retain the asymmetric property and also protect the privacy of a buyer, whose identity is only revealed and disclosed in case of illegal re-distribution. Thus, an anonymous fingerprinting protocol ensures copyright protection, privacy and security for both the buyer and merchant simultaneously. A complete and sound anonymous fingerprinting protocol is expected to solve the following requirements (Ju et al., 2003):

   a) **Buyer frameproofness:** The merchant should not be able to frame an honest buyer of illegal re-distribution.

   b) **Piracy tracing:** The merchant should be able to trace and identify an illegal re-distributor in case of finding a pirated copy.

   c) **Anonymity:** The identity of a buyer should remain anonymous during transactions until he/she is proven guilty of copyright violation.

   d) **Collusion resistance:** In a collusion attack, several attackers fabricate a new copy through combining their unique copies in order to avoid the tracing. Thus, the scheme must be collusion-secure to defy collusion attacks.

e) **Unbinding:** Upon discovering a pirated copy, the merchant can fabricate piracy by transplanting the buyer's fingerprint into other digital content. Therefore, it is necessary to bind a chosen fingerprint with a specific transaction.

f) **Non-repudiation:** A malicious buyer who has re-distributed an unauthorized copy should not be able to claim that the copy was created by the merchant.

g) **Unlinkability**: Nobody can determine whether different fingerprinted contents are purchased by the same buyer.

h) **Dispute resolution:** The trusted third party (judge/arbitrator) should be able to resolve disputes, without the buyer revealing his/her identity.

Various asymmetric fingerprinting schemes have been proposed (Martínez-Ballesté et al., 2003; Kuribayashi, 2010) in which the requirement of fair multimedia content distribution has become prevalent. Some asymmetric fingerprinting protocols also provide buyers with anonymity (B. Pfitzmann & Waidner, 1997; B. Pfitzmann & Sadeghi, 1999; Memon & Wong, 2001), in which trusted third parties are usually introduced to provide fairness and anonymity to the merchant and the buyer, respectively. Various fingerprinting schemes do not involve trusted parties for the execution of the protocols (Choi et al., 2003; Deng & Preneel, 2008). The initial asymmetric fingerprinting protocols were based on bit-commitment schemes (B. Pfitzmann & Schunter, 1996; Biehl & Meyer, 2002), which require high enciphering rates to achieve security. Thus, the implementation of these protocols involves a large overhead and high communicational cost. Other proposals, like (Kuribayashi & Tanaka, 2005; Prins, Erkin, & Lagendijk, 2007), apply a homomorphic property of public-key cryptosystem to achieve the asymmetric fingerprinting. The homomorphic property allows the merchant to embed the fingerprint in the encrypted domain in such a way that only the buyer obtains the decrypted fingerprinted content. However, the use of homomorphic encryption expands data and substantially increases the communication bandwidth required for data transfers. Hu and Li (2010) proposed an asymmetric fingerprinting protocol based on 1-out-of-2 oblivious transfer protocol from the communication point of view. Thus, in any case, all the proposed asymmetric fingerprinting schemes involve complex cryptographic protocols which require high bandwidth and heavy computational costs. This makes the schemes impractical in a real-world scenario. Pagnia and Gartner (1999) prove that efficient fair exchange protocols cannot be completely fair without the help of a third party that is mutually trusted by both of the parties performing the exchange.

2.2.4.3 Collusion-Resistant Fingerprinting

In digital fingerprinting, unique identification information is embedded in each distributed copy and serves as a digital fingerprint. This unique information is used to trace and identify the source of illicit copies. In this way, the original buyer is deterred from illegally re-distributing his/her fingerprinted content. A general weakness of digital fingerprinting occurs when a coalition of buyers compare their uniquely fingerprinted multimedia to exploit the differences amongst their unique fingerprints in order to remove or alter the fingerprint so as to evade being traced, and at the same time possibly frame an innocent buyer. This attack is known as collusion attack and such group of collaborating buyers is called a set of colluders or a coalition. A segment of the content is called a detectable position if the colluders have at least two differently marked versions of that segment available. The fingerprint must, therefore, survive both standard distortions (such as compression, filtering, data conversion and channel noise) and collusion attacks by malicious users intending to destroy it. Thus, if not properly designed, a fingerprinting system might fail to detect the traces of any fingerprints under collusion attacks with only a few colluders. The method of identifying colluder(s) is also called the tracing algorithm as depicted in Fig. 2.6. Collusion has been the main research challenge in the realm of fingerprinting and therefore, it is desirable to design fingerprints that can resist collusion and identify the colluders.

A growing number of techniques have been proposed in the literature to provide collu-

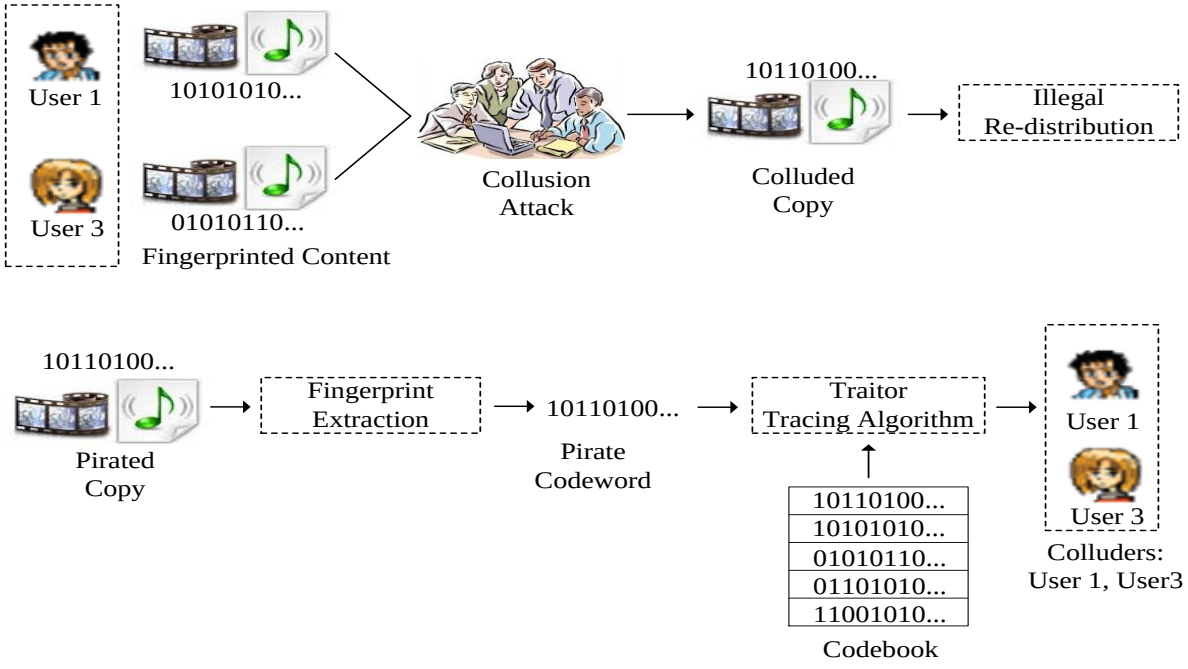


FIGURE 2.6: Traitor-tracing process

sion resistance in multimedia fingerprinting systems. There are many fingerprinting algorithms designed to be resistant against large as well as small collusion sizes. The major existing fingerprinting codes can be categorized into the following types:

1. **c-frameproof code:** : In c-frameproof code, no collusion of at most c users can frame a user who is not a member of the collusion. Boneh and Shaw (1999) (B-S) first introduced the concept of frameproof code based on a marking assumption and presented a binary scheme with the lower bound on the code length $m = O(c_o^4 \log |N/|\varepsilon_1| \times |\log \varepsilon_1{}^{-1}|)$, where ε_1 is a probability of accusing an innocent user, N is the number of users and c_o is the coalition size to be resisted. The marking assumption introduced by B-S states that the colluders can only manipulate the fingerprint at positions where they detect a change when comparing their marked copies.

   Unfortunately, the large length of the B-S code restricts the range of its practical applications. Much research has been carried out to reduce the code length and improve its performance. For example, Schaathun (2003) shows that the B-S fingerprinting scheme performs better by taking a different approach in error-analysis of the B-S scheme and proposed codes with length shorter than B-S codes. Sebé and Domingo-Ferrer (2002) used the same construction principles as the ones of Boneh and Shaw. They considered concatenation of random looking codes with an error-correcting code. However, a weak attacker model was considered and the scheme derived in this work was deterministic 3-collusion resistant.

2. **c-secure frameproof code:** A c-secure frameproof code is a stronger form of c-frameproof code. A code is c-secure frameproof code if it is impossible for collusion C_1 of maximum size c to frame a disjoint collusion C_2 of maximum size c by generating a colluded code that could have been generated by C_2. The codes proposed by Stinson, Tran, and Wei (1998) are c-secure frameproof codes. If a code has the $(1, c)$-secure frameproof property, then, no coalition of size at most c will be able to generate the fingerprint of any user. However, they may generate a pirate codeword and claim that it was generated by another c-coalition. With an (c, c)-SFP code, they would not be able to accuse a completely disjoint coalition. However, this does not guarantee that some traitor may be caught. The c-secure-frameproof codes are also referred to as (c, c)-separating codes (c-frameproof code and c-secure frameproof code do not have traceability, namely, the identification of guilty users cannot be guaranteed).

3. **c-traceability code:** Codes with the traceability (TA) property are of remarkable significance. A c-traceability code has an advantage that it allows an efficient algorithm to determine one member of the collusion by simply finding the codeword closest to the modified codeword. Different designs of traceability codes have been proposed by several researchers in recent years, e.g. Chor et al. (1994), Naor and Pinkas (1998) and Staddon et al. (2001). However, there is no guarantee the tracing result is correct if the size of the coalition is greater than c. The first attempt to gain more information about the coalition when its size is bigger than c was proposed by Anthapadmanabhan and Barg (2009), who defined the concept of two-level fingerprinting codes. Although TA codes have a very efficient algorithm for traitor tracing, there is no explicit construction for two-level TA codes available, except in the trivial case.

4. **c-identifiable parent property code:** In c-identifiable parent property (IPP) codes, no collusion of maximum size c can generate a codeword that cannot be traced back to at least one member of the collusion. Codes with IPP are also capable of identifying traitors, requiring less restrictive conditions than the TA codes at the expense of not having an efficient decoding algorithm. A c-TA code is a c-IPP code, but a c-IPP code is not necessarily a c-TA code. Therefore, the set of c-TA codes is a subset of c-IPP codes. Codes with the IPP were introduced by Hollmann et al. (1998). The IPP has received considerable attention in the recent years, having been studied by several authors (Fernandez & Soriano, 2004; Trung & Martirosyan, 2005). The codes proposed by Trung and Martirosyan (2005) have the best known asymptotic behavior. Their construction is based on IPP code concatenation. The length m of the codewords is $O((c^2)^{\log^*(M)}(\log(M))$, where M is the number of codewords, and the function $\log^* : \mathbb{Z}^+ \to \mathbb{Z}^+$ is defined recursively by $\log^*(1) = 1$ and $\log^*(a) = \log^*(\lceil \log a \rceil) + 1$ (if $a > 1$). Also, these codes allow a traitor tracing algorithm with a runtime of $O(M)$ in general. Fernandez and Soriano (2004) proposed a new decoding algorithm for IPP codes. A strong point of the proposed decoding procedure is its symmetry, because encoding and decoding are based on the same algorithm. Also, the decoding algorithm performs better in terms of complexity. Currently, there are already several papers discussing bounds on the size of IPP codes. The case of a fixed length and large alphabet size has been considered by Hollmann et al. (1998); Alon, Fischer, and Szegedy (2001); Blackburn (2002), the case of a fixed-size alphabet and growing length has been treated by Chor et al. (1994); Naor and Pinkas (1998); Barg and Kabatiansky (2013) and case of efficient decoding of IPP codes including list decoding

techniques has been studied by Naor and Pinkas (1998); Fernandez and Soriano (2004).

5. **c-secure code with ε-error** In a c-secure code with ε-error, a member of collusion of maximum size c can be traced back with probability at least $1-\varepsilon$. Tardos codes, Skoríc codes and Nuida et al. codes are some examples of c-secure codes. Tardos (2003) proposed an efficient code construction that, for the first time, reduced the code length to the theoretical lower bound $O(100c_0^2 \log(N/\varepsilon_1)$ (c_0 is the coalition size to be resisted, N is the total number of users and ε_1 is the probability of accusing an innocent user), thereby making such codes practical. Tardos codes are currently the state-of-the-art for c-secure collusion resistant fingerprinting. Tardos codes is constructed in two phases: First, the provider chooses the codeword distribution and accusation algorithm and then, the merchant builds the fingerprinting matrix. The entries of the matrix are based on a probability vector p generated from Tardos probability density function, which is biased towards the values close to 0 and 1. In case that a merchant finds an unauthorized copy, he/she extracts the fingerprint message. The extracted fingerprint, f', is compared to all the distributed fingerprints. If none of them matches f', the merchant assumes a collusion attack and starts the accusation algorithm. Therefore, the accusation calculation is made according to a pair-wise score between each distributed fingerprint and the manipulated one. Its performance is usually evaluated in terms of the probability of accusing an innocent user (ε_1) and the probability of missing all colluders (ε_2). Most of the articles dealing with the Tardos codes aim at finding a tighter lower bound on the length of the code. Significant reduction in the code length is proposed by Skoríc, Katzenbeisser, and Celik (2008), in which the 0 bits in the fingerprint are equally informative as 1 bits. Tardos only considered bits with value 1 as informative. This improves the code already by a factor of 2 for the code length. Later in their paper, the authors decoupled the false positive rate ε_1 from the false negative rate (ε_2), which further reduced the code length by a factor of 5. Nuida et al. (2007) proposed fingerprinting codes with outstandingly short code lengths and introduced a discrete bias distribution that depends on c_0. The accusation step is almost identical to the Tardos one. The modified bias distribution improves the tracing if $c \leq c_0$, but it has worse properties at $c > c_0$.

6. **k-anti-collusion code:** k-anti-collusion code (ACC) has the property that the composition of any subset of k or fewer codewords is unique and therefore can identify groups of k or fewer colluders. $k-1$-resilient logical AND anti-collusion code (AND-ACC) is a code whose composition is an element-wise logical AND operation. The k-resilient AND-ACC

from balanced incomplete block design (BIBD) is called a BIBD code. A BIBD code is designed originally in the area of combinatorial theory. They are used to arrange elements of a set in which the arrangement possesses certain desired properties in terms of the number of elements per block. Trappe et al. (2003) employ a BIBD code and give an efficient detection method based on a tree structure. A construction for AND-ACCs was proposed by using the bit complement of the incidence matrix of a BIBD design to accommodate more users while providing collusion resistance. Dittmann et al. (1999) proposed an ACC construction scheme based on finite projective, where the relationship between the points and lines of the projective space is used to represent fingerprints. S. He and Wu (2006) proposed a group-based joint coding and embedding technique. The advantage of the proposed scheme is that the fingerprinting strategy of joint coding and embedding substantially improves the collusion resistance of group-based fingerprinting, while preserving its advantages of compact representation and efficient detection.

The existing fingerprinting schemes discussed above have different assumptions about marks, attacks and attackers. In particular, they differ in the following aspects:

- **Assumptions about the possible actions of attackers:** Narrow case attack model (where on every detected positions attackers can output only those symbols that they see on these positions in their codewords), and general case attack model (where attackers are assumed to be able to output any symbol, even an unreadable symbol on detected positions).

- **Approaches to resilience of fingerprinting schemes:** Deterministic (schemes that enable finding at least one of the coalition members with certainty) and probabilistic (schemes that allow some error when detecting guilty users).

- **Assumptions about the robustness of the schemes against particular collusion attacks:** Linear collusion attacks (a generalized form of an average attack in which codewords of colluders are typically averaged with an equal weight) and non-linear collusion attacks (minimum, maximum and median attacks on the fingerprinted codewords).

From researchers' point of view, given different multimedia content, the main question is which fingerprinting scheme is the most appropriate to trace a colluder or number of colluders suffering from different collusion attacks with different marking assumptions. The other major influencing factors are: robustness, imperceptibility, embedding and decoding efficiency and code length.

2.2.4.4 Prior work on Audio and Video Fingerprinting

This section presents an overview of fingerprinting schemes proposed for audio and video content. The goal is to provide a brief overview of the recent advances in fingerprinting for fair content exchange between a content provider and a buyer, collusion resistance and traitor tracing.

- **Audio Fingerprinting:** In the following, some anonymous and collusion-resistant fingerprinting schemes proposed for protection of an audio data are presented:
  Domingo-Ferrer and Megías (2013) proposed a fingerprinting protocol for P2P networks in which each user obtains a different fingerprinted copy of the content which allows the content provider to trace illegal re-distributors without affecting the privacy of honest users. The authors used rewards and punishment concepts of game theory to ensure that peers rationally cooperate in P2P fashion for fingerprint embedding and content distribution. To provide a proof of concept, the protocol has been realized using a robust audio watermarking scheme (Fallahpour & Megías, 2010). The scheme tolerates embedding several successive fingerprints without significant damage to the content utility or the previous fingerprints. In order to preserve the privacy of the input and output information in each execution of the fingerprinting scheme, a secure two-party computation protocol is used as a building block of the anonymous fingerprinting protocol, which results in increased computational and communication costs at the user end.
  Most of the reported collusion-resistant fingerprinting schemes are devoted to digital images and only very few are validated with audio signals. Only a few schemes provide collusion-resistant fingerprinting and all other audio fingerprinting schemes are proposed for content identification and do not consider the collusion attacks. Garcia-Hernandez and Feregrino-Uribe (2013) extended the state-of-the-art collusion-resistant fingerprinting ideas and proposed a collusion-resistant audio fingerprinting scheme. Each fingerprint is formed by a PN-sequence representing a group ID and other representing one user ID. Instead of using the full signal, a block-based fingerprint embedding strategy is adopted. The proposed fingerprinting system uses the DCT basis as fingerprint modulators and the insertion domain is the set of Modulated Complex Laplace Transform (MCLT) magnitudes. The fingerprint is replicated several times along the audio signal in a block-processing fashion and thus, it is possible to detect colluders with only a fraction of the whole audio clip. A detection strategy using several MCLT magnitude blocks is proposed, where for

each available MCLT coefficients block, group and user detection is carried out according to a particular threshold value. The extensive simulations show the security of the proposed algorithm against average collusion attacks. However, the resistance of the scheme is not proven against non-linear collusion attacks.

- **Video Fingerprinting:** This part presents a brief overview of the current fingerprinting schemes proposed for copyright protection of video data.

  Deng and Preneel (2008) proposed a new anonymous fingerprinting protocol, based on dynamic group signatures (Camenisch, 2000) and an additive homomorphism (Islam, Puech, & Brouzet, 2009), to provide all the required security properties, namely traceability, dispute resolution, buyer-frameproofness, anonymity, unlinkability and non-repudiation. The authors have assumed a still image for testing purposes and propose that the protocol can be applied to other data format such as video. The proposed protocol provides revocable anonymity such that a buyer can purchase digital content anonymously but the buyer's anonymity can be revoked as soon as he/she is proven guilty. The buyer is only required to interact with the third party (certification authority) prior to transactions and with the merchant during multiple transactions. Using the additive homomorphism concept, an encrypted watermark is embedded in an encrypted content by adapting the quantized DCT coefficients. For collusion resistance, the authors assume that the adopted watermarking strategy is required to be collusion resistant. The security of the underlying homomorphic cryptosystem requires the use of very large algebraic structures that results in a high-data expansion from the plain-text to the encrypted signals. As a result, an encryption of a large-size data such as video, would require high computational and bandwidth requirements.

  S. He and Wu (2006) proposed a joint coding and embedding scheme based on Reed-Solomon codes for a large-scale video fingerprinting. The proposed design can accommodate more than ten million users resisting collusions performed by hundreds of users. The permutation subsequent embedding technique is applied to enhance collusion resistance. In permuted sub-segment embedding, each segment of the fingerprint sequence is partitioned into β sub-segments and these sub-segments are then randomly permuted according to a secret key. The permuted fingerprint sequence is added to the host signal through spread-spectrum embedding with perceptual scaling to form the final fingerprinted signal. A trimming-based detection algorithm is proposed that significantly speeds up the detection while maintaining good detection performance. The experimental results show that

the joint coding and embedding strategy substantially improves the collusion resistance at affordable computational complexity. Due to the fact that system can accommodate 16 million users, the length of the code is very long, i.e. 260 Mbits.

2.2.4.5 Research Challenges

The existing anonymous fingerprinting algorithms share a common drawback, i.e. the computational and communicational costs are quite high, due to the use of at least one of the following highly demanding technologies: homomorphic encryption, bit-commitment or secure multi-party computation schemes, thus making the schemes impractical in a real-world scenario. In addition, some of the schemes restrict the use of specific watermarking technologies which are not among the most robust and secure ones or even rely on a watermarking system for which no proof of existence has been provided yet. Additionally, merging collusion-resistant fingerprinting schemes and secure embedding is a difficult task. In most of the existing fingerprinting schemes, it is assumed that the use of anti-collusion codes can make the schemes resistant against collusion attacks without giving any proof of concept. Recently, two asymmetric fingerprinting schemes based on c-secure codes were proposed. Charpentier et al. (2011) proposed a solution that allows a buyer to pick up fingerprint bits from a list controlled by the merchant, in such a way that the he/she does not know the chosen elements. However, the proposed scheme requires heavy computation due to use of an oblivious transfer protocol. Also, the number of communication rounds between a buyer and a merchant is impracticable as it has a linear relation with the length of the code. Pehlivanoglu (2013) proposed an asymmetric fingerprinting scheme based on B-S code with constant communication round but at a cost of a longer codeword.

Moreover, all the collusion-resistant codes have a trade-off among the size of user base N, the collusion-resilience c_0, and codeword length m. As N or c_0 increases, m grows abruptly and vice versa. This trade-off makes the traceability code impractical because a large user base and collusion resistance are needed in many applications, but these requirements will make the codeword very long.

An ideal fingerprinting system should therefore, be able to provide high collusion resistance, low embedding computational complexity, high robustness against common signal processing attacks, low communicational cost and large user base with small length codewords.

2.3 Privacy Protection Techniques

P2P has become an integrated part of the Internet traffic by attracting millions of users. The most popular P2P applications remain file sharing and content distribution. P2P environments for data-centered applications offer valuable characteristics (e.g. scalability, distribution, autonomy) but limited guarantees concerning data privacy. They can be considered as hostile because data can be accessed by everyone (by potentially malicious peers) and used for everything. For example, in a recent study (Vijayan, 2010), Eric Johnson described how university researchers discovered thousands of documents containing sensitive patient information on popular P2P networks. One of the 3,000 files discovered by the researchers was a spreadsheet containing insurance details, personally identifying information, names and diagnosis codes on more than 28,000 individuals. Thus, P2P systems must take into account data privacy which is not the case in current P2P systems.

The other major issue is that the P2P systems are currently associated with an illegal sharing of copyrighted materials, especially music, movies, videos and software. In recent years, the copyright infringement problem has motivated the researchers to develop content protection techniques to prevent piracy in P2P networks. However, these content protection techniques have been criticized for implicating user's privacy by collecting information about the user, such as transaction history, usage habits, purchasing behaviors or other profiling information. These constraints instigate an adversarial relation of a user with the content provider. Hence, an incorporation of content protection mechanisms in P2P system can have serious effects on the privacy interests of users: the fact that a tracing mechanism makes use of a systematic record which details what multimedia files are downloaded through a specific IP address, history of files shared or downloaded, or a list of the peers with whom a user has interacted in the past, ultimately disrespects the private space of the user. In this context, providing privacy to a user of P2P system is a challenge.

In our modern days, the interest in the right of information privacy is increasing with the advent of information technology. The surveillance potential of powerful computer systems prompted demands for specific rules governing the collection and handling of personal information. In the literature, there is no consensus on a definition of data privacy. It can vary depending on the domain in which it is applied. In information technology, R. Clarke (1998) defines data privacy as *the right of individuals to claim that data about themselves should not be automatically available to other individuals and organizations, and that, even if data is possessed by another party, the individual must be able to exercise a substantial degree of control over that*

data and its use. Similarly, Westin (1967) defines data privacy as *the right of individuals to determine for themselves when, how and to what extent information about them is communicated to others*. Four basic functions of privacy are outlined by Westin (1967), namely personal autonomy, emotional release, self-evaluation, and limited and protected communication. A taxonomy of privacy has been proposed by Solove (2006) as a framework for understanding privacy from a social and legal perspective. According to Solove's taxonomy, privacy encapsulates information collection (surveillance), information processing (aggregation and identification), information dissemination (breach of confidentiality and disclosure exposure) and invasion (intrusion).

2.4 Classification of Privacy Protection Techniques

Privacy deals with the control on dispersion of one's personal information. This can be achieved with the help of anonymity. In P2P systems, encrypting the communication between two users can only hide the contents of their transaction. The malicious entities can get various details like IP addresses, duration of communication, etc. that can reveal their identity. Thus, there is a necessity to hide this information to enhance the privacy of users in a system. In communication perspective, there exist three types of anonymity: receiver, sender and mutual anonymity (cf. Section 1.2.2). Various anonymity mechanisms have been proposed that serve as tools for the protection of data and user privacy in content distribution applications. In this section, various existing privacy protection technique are briefly described.

2.4.1 Anonymity Techniques

Anonymity techniques are mostly used to make a user indistinguishable from other users, thus providing anonymity among a group of users. In the following part, a brief overview of existing anonymity techniques is presented:

2.4.1.1 Pseudonymity

Pseudonyms are generally dynamic identifiers, or names of the users that are hard to be linked to the real identities. In other words, a pseudonym is an identifier of a user other than one of the users' real names. From the perspective of a user, different levels of anonymity are important, e.g. ranging from latent identification to complete indistinguishability from other users.

1. **Latent identification:** The user identifies himself to a trustee and adopts a unique pseudonym that becomes registered with his/her identity. Using this pseudonym, he/she is subsequently able to interact with the system without revealing his/her identity. Additionally, it allows the system to determine the identities behind the pseudonyms, from the trustee who issued the pseudonym. This revocation of pseudonyms may be desirable in cases of misuse or when the identification of the user becomes necessary for other reasons, such as guilty of copyright violation and payment issues. Since the users' true identities are kept private, in the context of secure P2P content distribution systems, this is useful for honest users who could communicate with the merchant without a fear of identity disclosure in case of tracing a copyright violator.

2. **Full Anonymity:** The user initially chooses a unique but otherwise uncontrolled pseudonym, which cannot be traced by an authority. This type of anonymity provides an excellent opportunity to malicious users to act maliciously without being traced.

A pseudonym can be reversible and irreversible: reversible pseudonyms allow the re-identification of the individual and irreversible pseudonyms cannot be reversed but allow record linkage. The creation of a reversible pseudonym generally involves encryption and requires that the re-identification depends on a secret key. Whereas, the primary mechanism for creating irreversible pseudonyms is through the application of a cryptographic hash function These are one-way functions that take a string of any length as input and produce a fixed-length hash value or digest.

2.4.1.2 Anonymous Communication

Another form of anonymity technique is anonymous communication which is largely used in distributed systems. Anonymous communication aims to preserve communication privacy within the shared network setting. While end-to-end encryption can protect the data content from adversarial access, it does not conceal all the relevant information that two users are communicating. Adversaries can still learn significant information about the entities and the traffic carried on the network. The research on privacy preserving communication was initiated by Chaum (1981). Since then, research in anonymous communication has been extended to many areas such as routing mechanisms and P2P communication systems. Works in this domain (e.g. onion routing, mix networks, crowds, etc.) generally aim to make communication ambiguous in

order to make it difficult for malicious users to collect information about the system entities and the shared data.

- **Onion Routing:** Onion routing is a distributed P2P mechanism that allows two users to communicate anonymously over the network (Reed et al., 1998). The main aim of onion routing is to prevent intermediary nodes from knowing the source, destination and contents of the message. It strongly resists traffic analysis, eavesdropping and other attacks both by outsiders and insiders. Onion routing works in the following way: An onion routing network consists of a set of onion routers and users. To send data, a user chooses a sequence of routers, called a circuit, in which each node or onion router (OR) in the path only knows about his/her predecessor and successor node, but no other node in the circuit. The messages in onion routing are encrypted with symmetric keys. The user generates two symmetric keys: a forward key and a backward key for each OR on his/her path; and forward and backward cryptographic functions which correspond with these keys. These two pairs of function keys are responsible for encrypting and decrypting the message along the path. When a node receives the message, he/she decrypts the outer encryption layer with his/her own symmetric key, obtaining the pair function key and the next node in the path. Then, the node encrypts the message using the new key and forwards the message to the next node. This process is repeated until the message arrives to its destination. Once the circuit is completed, the reply traffic is sent encrypted in the opposite manner: each router encrypts and forwards the result of its predecessor onion router. Onion routing preserves the unlinkability in the communication, however it does not offer resistance to timing attacks in case a dishonest node owns the first and last node of a circuit.

- **Chaum Mixes:** A mix enables anonymous communication by means of public cryptography, scrambling the messages, and unifying them (padding to constant size, fixing a constant sending rate by sending dummy messages, etc.). Chaum (1981) mixes support sender anonymity and protect from traffic analysis. It requires public-key cryptography, which is computationally expensive. Chaum mixes works as follows: When a user wants to send a message in a Chuam mixing network, he/she must first choose a route through a series of Chaum mixes $(M_1, \ldots, M_N)$ as shown in Fig. 2.7, to the intended recipient, and then prepare a layered message for delivery. The first layer includes the name of the recipient and the message encrypted with the public key K_{pC} of the recipient. The second layer includes M_N and the first layer encrypted with the public key of M_N. This

continues until the last layer includes M_1 and the last one layer encrypted with the public key of M_1 (K_{pM_1}). This last layer represents the message that is actually sent out. For example, if $N = 2$ and the recipient is C, the message that is sent out may look as follows: $M_1, \left\{M_2, \left\{C, \{\text{message}\}\, K_{pC}\right\} K_{pM_2}\right\} K_{pM_1}$. When this message reaches M_1, the Chaum mix uses its private key K_{sM_1} to decrypt it. The result is split into two parts: M_2 and $\left\{C, \{\text{message}\}\, K_{pC}\right\} K_{pM_2}$. The first part is used to route the second part to M_2. M_2, in turn, decrypts $\left\{C, \{\text{message}\}\, K_{pC}\right\} K_{pM_2}$ using its private key K_{sM_2}. The result is C and $\{\text{message}\}\, K_{pC}$. It is then forwarded to C, who can apply his/her private key K_{sC} to decrypt the message.

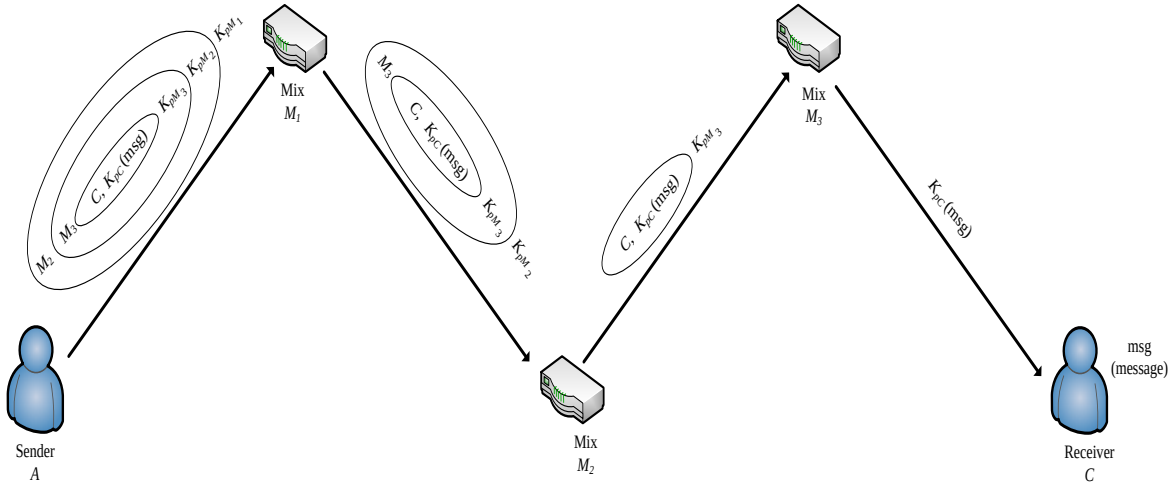


FIGURE 2.7: Chaum Mix

- **Dining Cryptographers network:** The Dining Cryptographers (DC) problem is the basis of a kind of network that gives absolute sender anonymity for messages, which is called a DC-net (Chaum, 1988). In DC-nets, each participant can send messages to all others and none can tell from whom this message is. If a participant wishes to send a message to a specific recipient, he/she can encrypt it in a way that only the intended recipient can decrypt. A DC network allows both the sender and the recipient to remain anonymous. The working of a DC-net is as follows: The DC-net protocol has two stages: First, N users set up pair-wise shared secrets through secret channels. Next, each user P_i transmits a one bit message x_i, which is the XOR of all the shared one-bit secrets that P_i holds if he/she has no message to send, or the opposite bit otherwise. Repeating this protocol allows users to send full messages. After complete transmission, the sent message is decoded by all participants through computing the XOR of the transmitted bits.

2.4.1.3 Anonymous Authentication

It is impractical to pursue user privacy without taking accountability into consideration. Without the fear of being identified and punished when they abuse the services, users are likely to misbehave due to malice, thereby disrupting system operations and harming everyone else. Accountability has traditionally been achieved through authentication mechanisms which verify the identity of a user who requests a service. In the pursuit of authentication schemes that balance privacy and accountability, numerous anonymous authentication techniques (group signatures, traceable signatures, authenticated key exchange protocols) have been constructed.

- **Group Signatures:** A group signature is a privacy-preserving signature scheme introduced by Chaum and Van-Heyst (1991). In this scheme, a group member can sign a message on behalf of the group without revealing his/her identity. Only a specific authority (group manager) can open a signature and find its originator. Signatures by the same user cannot be identified as from the same source. The security properties of group signature scheme are the following:

  a) **Correctness:** Correctness means that valid signatures by group members always verify correctly, and invalid signatures always fail verification.

  b) **Unforgeability:** Only members of the group can create valid group signatures.

  c) **Anonymity:** Anonymity implies that given a message and its signature, the identity of the signer cannot be determined without the group manager's secret key.

  d) **Unlinkability:** Given two messages and their signatures, one cannot tell if the signatures were from the same signer or not.

  In a group signature scheme, the group manager issues group certificates to group members. The group certificate is a special construction jointly produced by the group manager and a group member through a Join protocol. The Join protocol ensures that a valid group membership certificate can only be produced with the help of the group manager, and a group member knows a secret corresponding to this certificate. The authentication can be performed by a group member (prover) placing a group signature on a challenge (nonce) sent by the user (verifier) requiring authentication.

- **Traceable Signatures:** Group signatures come with a mechanism which allows the group manager to open a signature and reveal the true signer by the group manager's decision.

To identify any signatures previously generated by a misbehaving user, the group manager is required to open all the signatures. This incurs two problems: it penalizes the privacy of all other honest users and imposes a high computational overhead on the group manager. In view of these shortcomings, Kiayias, Tsiounis, and Yung (2004) proposed the concept of traceable signatures. Traceable signature schemes extend a group signature scheme with an enhanced anonymity management mechanism. The group manager can compute a tracing trapdoor which enables anyone to test if a signature is signed by a given misbehaving user, while the only way to do so for group signatures requires revealing the signer of all signatures. The group manager may delegate the trapdoor to many tracing agents (TAs) to check whether a signature was issued by a given user.

- **Password-Authenticated Key Exchange:** Password-Authenticated key exchange (PAKE) is a form of Diffie-Hellman key exchange protocol. PAKE allows each party to authenticate the other's identity based solely on their knowledge of a short password, without revealing any useful information about the password to any other party. Moreover, the two parties can also agree on a shared key suitable for other cryptographic purposes such as bulk encryption. Two forms of PAKE are the Balanced and Augmented methods.

  a) **Balanced PAKE:** Balanced PAKE allows parties using the same password to negotiate and authenticate a shared key. One example of PAKE is the Encrypted Key Exchange (EKE) protocol introduced by Bellovin and Merrit (1992). Encrypted key exchange is a protocol that allows two parties sharing a common password to communicate over an insecure network without exposing that password. EKE involves a combination of asymmetric (or public-key encryption) and symmetric (or secret-key encryption). Each party holds a pair of public/private keys. The public key is known by all the parties and the private key is kept secret. In EKE, a secret key, or a password, is derived from one party's public key and another party's private key. The shared secret key is then used to encrypt subsequent communications between the parties, who may have no prior knowledge of each other, using a symmetric-key cipher.

  b) **Augmented PAKE:** Augmented PAKE is applicable to client/server scenarios, in which the server does not store password-equivalent data. An example of Augmented PAKE is the Secure Remote Password protocol (SRP). SRP (T. Wu, 1998) is a secure password-based authentication and key-exchange protocol that solves the

problem of authenticating clients to servers securely. Additionally, the SRP protocol performs a secure key-exchange during the authentication process. SRP does not require trusted key servers or certificate infrastructures and clients are not required to store or manage any long-term keys. On the server, a mathematically generated number is stored. This number is based on a user-chosen password and a randomly generated salt (a random value that is used in one-way hashing algorithms). Both the client and server maintain a pre-determined prime number and a primitive root based on that prime number. The nature of all these numbers allows an authentication without the server needing to save the password. The client asks for the salt that was created, and then a series of calculations are performed with the client and server exchanging the calculated values.

2.4.2 Trust Techniques

These are techniques that provide privacy protection by handling the trustworthiness of users without revealing their true identities (i.e. users with pseudonyms could be assigned trust levels even if their identities are kept private). Trust management techniques have been proposed as mechanisms that allow potentially unknown parties to decide whom is trusted enough to provide or access requested data. They allow unknown parties to access resources by showing appropriate credentials that prove their qualifications to acquire data. The concept of trust is closely linked to reputation. Reputation is considered as a collective measure of trustworthiness based on ratings from parties in a community and can be used to establish trust relationships. Reputation can be used by parties in making a decision whether or not to work with the other party in the future. In general, trust management techniques are classified into two categories: reputation-based trust management and credential-based trust management.

2.4.2.1 Reputation-based Trust Management Systems

Reputation-based trust management systems provide a mechanism by which a user evaluates his/her trust in the reliability of the data and the data provider. Users in such systems establish trust relationships with other users and assign trust values to these relationships. A trust level assigned to a trust relationship is a function of the combination of the user global reputation and the evaluating user perception of that user. A good reputation system should be able to:

- Identify malicious peers in order not to be selected as transactions partners.
- Spread information regarding a malicious peer in case that a negative transaction occurred. This is a retaliation measure after ending a transaction to help other users in future interactions.

Some reputation systems that have been proposed in the literature are XRep (Damiani et al., 2002), EigenTrust (Kamvar et al., 2003) and FGTrust (Zhang & Fang, 2007).

Different approaches have been proposed to aggregate trust values received from recommended users and synthesize them to generate a reputation value for a providing user.

1. **Deterministic Approach:** In this approach, a user's reputation is based on a simple summation or average of collected ratings. For example, the reputation scheme used in *ebay* (1995) is based on the sum of the number of positive and negative ratings.
2. **Probabilistic Approach:** Bayesian networks (Y. Wang & Vassileva, 2003) and Maximum likelihood estimations (MLE) (Despotovic & Aberer, 2006) are two examples of reputation systems based on a probabilistic approach. Bayesian systems take binary ratings as inputs and are based on computing reputation scores by statistical updating of beta probability density functions. On the other hand, in MLE, the reputation value is the probability of a user to cooperate and it is chosen to maximize the probability of the available ratings.
3. **Flow Models:** Flow models are systems that compute trust or reputation based on transitive iteration through looped or arbitrarily long chains.

2.4.2.2 Credentials-based Trust Management Systems

In credential-based trust management systems, users use credential verification to establish a trust relationship with other users. The primary goal of such systems is to enable access control. Therefore, their concept of trust management is limited to verifying credentials and restricting access to resources according to application-defined policies. A user (i.e. data owner) provides access to restricted data to other users (i.e. data requesters) only if he/she can verify the credentials of the requester. This is useful by itself only for those applications that assume implicit trust in data owners. Examples of these systems are PeerTrust (2004) and OpenTrust (2004). Since the credential-based trust mechanisms do not incorporate the need of the user to establish trust in the data owner by themselves they do not provide a complete generic trust management solution for decentralized applications. Some trust management systems, such as KeyNote (Blaze,

1999), and PolicyMaker (Blaze, Feigenbaum, & Lacy, 1996) attempt to provide trust in large-scale distributed networks through the use of credentials that delegate permissions. However, these systems are based on the assumption that data owners and their services are fully trusted and data requesters are not trusted. The requesters have to be verified each time.

2.4.3 Cryptography-based Techniques

These techniques are used to provide data security and privacy by making data unreadable for unauthorized users. Besides providing data security, a few cryptographic protocols (encryption algorithms, hash function, zero-knowledge proof of identity) also form an integral part of anonymous communication and authentication protocols. Section 2.2.1.2 describes the symmetric-key, asymmetric-key algorithms and hybrid encryption. Thus, in the following, the other cryptographic techniques, namely, cryptographic hash functions and zero-knowledge proofs-of-identity are described.

2.4.3.1 Cryptographic Hash Function

Cryptographic hash functions are one-way functions that take variable-length data as input, and output a fixed length hash value. The hash value is a summary of the original data and is substantially smaller than the original data. Cryptographic hash functions are used notably in digital signatures and authentication schemes. There are several well-known hash functions used in cryptography. These include MD4, MD5 and the Secure Hash Algorithm (SHA). MD4 (Rivest, 1990) is a long-used hash function whose security has been severely compromised. The first full collision attack against MD4 was published in 1995 and several newer attacks have been published since then. MD5 (Rivest, 1992), a strengthened variant of MD4, is also widely used but is broken with regards to collisions. The USA National Security Agency (NSA) developed the SHA series. SHA (1992) specifies four secure hash algorithms- SHA-1, SHA-256, SHA-384 and SHA-512. SHA-1 is currently the most widely deployed hash function. It forms a part of several widely used security applications and protocols. The SHA-1 (1995) hash is called secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest.

2.4.3.2 Zero-Knowledge Proof-of-Identity

A Zero-knowledge proof-of-identity (ZKPI) system (Feige et al., 1998) is a cryptographic protocol between two parties whereby the first party wants to prove his/her identity to the second party without revealing anything about his/her identity. Following are the three main properties of ZKPI system:

a) **Completeness:** The honest prover convinces the honest verifier that the secret statement is true.

b) **Soundness:** A cheating prover cannot convince the honest verifier that a statement is true (if the statement is really false).

c) **Zero-knowledge:** A cheating verifier cannot obtain anything other than prover's public data sent from the honest prover.

A large class of zero-knowledge protocols consists in repeating n times the following three message rounds:

1. Prover to verifier: Witness
2. Verifier to prover: Challenge
3. Prover to verifier: Response

The prover selects a random element from a pre-defined set as its secret commitment and from this computes a public witness. The prover basically asserts that it can answer a number of questions. The verifier probabilistically tests this by asking one of these questions. If the prover is the one it claims to be, then it can answer all questions successfully. The answer to any one of these questions does not provide information about the secret commitment. The verifier checks the answer for accuracy. The protocol is repeated n times. Zero-knowledge proof protocol can be applied in many cryptographic applications and operations, such as anonymous authentication and key-exchange protocols.

2.5 Conclusions

In this chapter, an overview of the techniques used for content and privacy protection is presented. In the first part, the state-of-the-art content protection techniques is introduced, which

can help content providers to prevent copyright infringement. Encryption is presented as a first line of defence against unauthorized access to digital content. As a second line of defence, DRM is discussed. A DRM technology promises the content owner persistent control over the content even when the data leaves the owner's machine. However, DRM technologies only provide partial solutions to a copyright protection problem and thus, are still not effective at combating piracy. Then, as the third line of defence, digital watermarking is discussed. Digital watermarking is a core technique that can be used to solve the problems of copyright protection, content integrity or tamper detection. Finally, as an application of watermarking, digital fingerprinting is presented that provides a traitor-tracing mechanism. Fingerprinting provides content protection by allowing a user specific identification mark (fingerprint) to be embedded in digital content such that if a content owner finds a pirated copy, he/she can identify a person responsible for illegal re-distribution.

In the second part, privacy protection techniques are presented, which are important to protect data and user privacy. Among privacy protection techniques, first the anonymity techniques are discussed that are mostly used to make a user indistinguishable from other users. Then, trust and reputation techniques are described. These can be used to predict the behavior of users in the system in some way and, thus, protect data privacy from malicious users. Finally, cryptographic schemes are discussed, which can protect data and user privacy. Neither of these techniques alone can ensure privacy in distributed environments. Thus, combining these techniques provide more privacy guaranties.

The main characteristics of the content and privacy protection techniques presented in this chapter can be used together to achieve multimedia security and user privacy in decentralized systems. The next chapter shows how these techniques are used in current P2P systems to provide content security and privacy preservation.

Chapter 3

Security and Privacy in P2P Content Distribution Systems

The previous chapter presented content and privacy protection techniques that can be used to provide multimedia security and user privacy in a distributed system. This chapter describes how these techniques are used in P2P content distribution systems to provide content security and privacy preservation.

3.1 Introduction

A P2P network is a newly emerging paradigm in the communication era. The popularity of P2P networks has increased tremendously in recent years. The reason for this popularity lies in the services provided by these networks by using the resources of their end users. Theotokis and Spinellis (2004) provide an elaborated definition of P2P Networks:

"Peer-to-Peer systems are distributed systems consisting of interconnected nodes able to self-organize into network topologies with the purpose of sharing resources such as content, CPU cycles, storage and bandwidth, capable of adapting to failures and accommodating transient populations of nodes while maintaining acceptable connectivity and performance without requiring the intermediation or support of a global centralized server or authority".

P2P systems have been implemented more and more in diverse applications and services. These systems have been successfully used in services for sharing computation *Seti@home* (1999), data *BitTorrent* (2000), communication *Skype* (2003) and real-time multimedia services *PeerCast* (2006). However, most of the current P2P systems fall within the category of content distribution, which includes systems and infrastructures designed for the sharing of digital content between end users. P2P content distribution has received considerable research attention in recent years. In this thesis, the focus is on content distribution application of P2P systems.

There are several features of P2P data-centered systems that differentiate them from traditional centralized data-centered systems.

- End users (peers) are autonomous and free to join and leave the system any time.
- P2P systems are highly scalable and can accommodate millions of peers.
- Peers have symmetric roles. Any peer can store objects on behalf of other peers, provide content on demand, support queries and perform routing of messages.
- A P2P system provides a shared resource pool. The resources a peer contributes include compute cycles, disk storage and network bandwidth. Peers offer and consume resources in a fair and balanced manner.
- The data is available any time and everywhere due to geographical scalability.
- The peers are provided with an efficient content search mechanisms to easily locate the content they desire.
- There is no centralized control over the data shared in the system.
- P2P systems are reliable and robust, since there is no single point of failure of the system.

The low-cost, scalability and ease of content dissemination presents a lucrative opportunity for multimedia companies to generate revenues through P2P systems. However, content providers have been reluctant to adopt P2P systems as a distribution vehicle to monetize digital content since these systems are plagued with piracy. The ability to make perfect copies and the ease with which these copies can then be distributed has given rise to significant problems regarding the misuse, illegal copying and re-distribution and misappropriation of copyright-protected content (Lee & Chen, 2002; E. Lin et al., 2005). Consequently, content providers feel threatened by the broad and unregulated exchange of the content in P2P systems. They apparently fear losing

control of content ownership in the sense that they are no longer in control of the content distribution and worry about the promotion of illegal activities. Also, the decentralized nature of the P2P technology makes them more resistive to its adoption due to absence of a central authority that could regulate how and what kind of files are distributed within the system. Moreover, tracing a copyright violator in a P2P system with millions of connected users is an immense task.

The open nature of P2P systems leads to the vulnerability of the whole system since it is easy for a malicious peer to join the system without having his/her identity verified. The modes of attacks on P2P systems can be classified into three types:

1. **Attack on anonymity of peers:** These attacks are used to reveal the identity of the peers that are sharing information in the system.

2. **Attack on a communication channel:** These attacks try to weaken the communication between the two communicating peers in the system by injecting malicious messages into the network. These attacks may also try to flood the network with multiple requests to prevent legitimate users from using the network.

3. **Attack on exchanged data:** In these attacks, an attacker tries to learn about the content of the exchanged data.

In recent years, researchers have proposed new solutions to add privacy aspects to P2P systems. The goal is to ensure data and peers' privacy without affecting the P2P advantages. In this chapter, a survey of P2P systems proposed for providing security to the content providers and privacy to the end users is presented. These systems are compared on the basis of multimedia security and privacy protection properties. The literature survey of P2P content distribution systems conducted in this chapter with the intention of describing the challenges and solutions associated with content and privacy protection in P2P systems is published as a conference paper (Qureshi, Rifà-Pous, & Megías, 2013a) and a working paper (Qureshi, Rifà-Pous, & Megías, 2013b).

The rest of the chapter is organized as follows. Section 3.2 presents P2P systems, the types of P2P systems and applications of P2P. In Section 3.3, current mechanisms are discussed which provide security in P2P content distribution systems. Section 3.4 discusses privacy-preserving P2P content distribution systems. Section 3.5 compares these systems with respect to security and privacy properties. The chapter is concluded in Section 3.6.

3.2 P2P Paradigm

P2P is often described as a type of decentralized computing system in which nodes, referred to as peers, use the Internet to communicate with each other directly. The P2P paradigm is a way to manage vast amounts of computing power, data (storage and content) and connectivity from personal computers distributed around the world (Milojicic et al., 2002). P2P systems are attractive because they do not require any special administrative arrangements, unlike centralized facilities, and their decentralized and distributed nature make them scalable, bandwidth-efficient and fault-tolerant.

3.2.1 Classification of P2P Systems

All P2P systems rely on a P2P network to operate. Such a network is built on top of the physical network (typically the Internet) and, thus, referred to as overlay networks (Fig. 3.1). The degree of centralization and the topology of overlay networks have significant influence on properties such as scalability, fault-tolerance, efficiency and security. Currently, P2P systems can be classified as unstructured and structured, depending on the overlay structures (Theotokis & Spinellis, 2004; Lua et al., 2005).

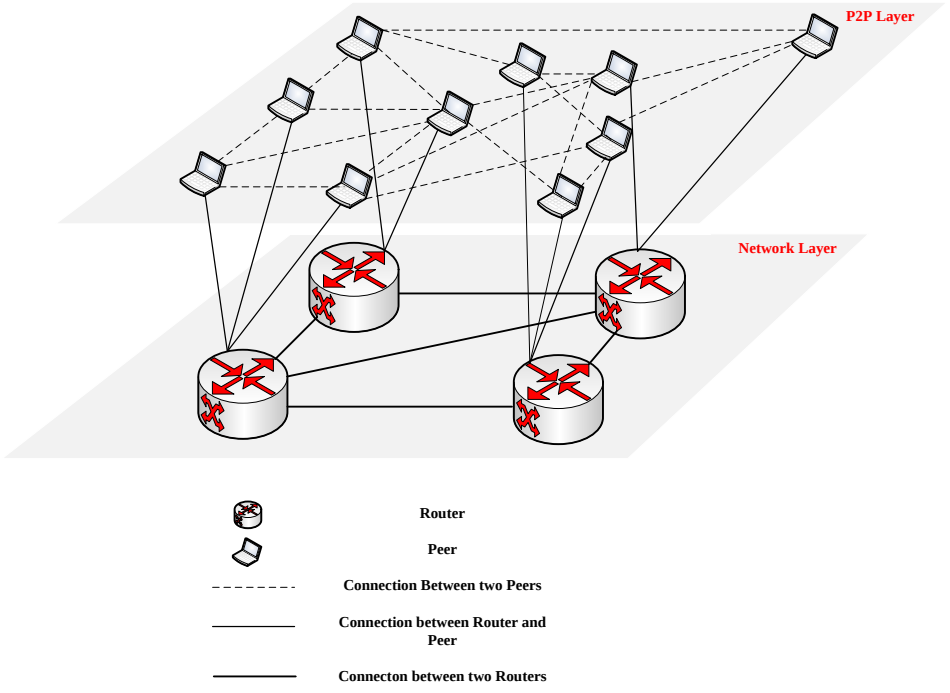
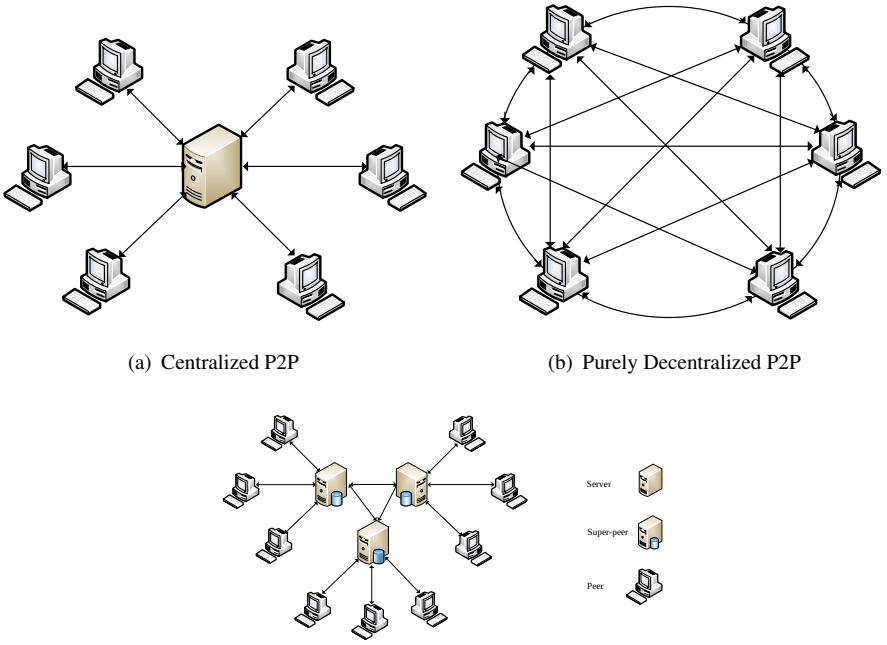


FIGURE 3.1: P2P overlay network on top of the Internet

3.2.1.1 Unstructured P2P

Unstructured systems do not impose any structure on the overlay networks. The overlay network is created in a non-deterministic way and data placement is completely unrelated to the overlay topology. Each peer knows his/her neighbours, but does not know the resources they have. Query routing is typically done by flooding the queries across the limited neighbourhood. Unstructured systems are simple to implement and incur virtually no overhead in topology maintenance. Consequently, most popular large-scale P2P systems are unstructured. However, the lack of structure makes it difficult to locate shared resources in the system. Examples of P2P systems supported by unstructured networks include *FreeHaven* (1999) and *gtk-Gnutella* (2000).

Although P2P systems are supposed to be fully decentralized, in practice, three categories of unstructured networks can be encountered: centralized, purely decentralized and hybrid decentralized.



(a) Centralized P2P

(b) Purely Decentralized P2P

(c) Hybrid P2P

FIGURE 3.2: Types of unstructured P2P Systems

1. **Centralized P2P:** In centralized P2P systems, there is a central server facilitating coordination between peers (Fig. 3.2(a)). Although an end-to-end interaction and file exchanges may take place directly between two peers, the central server facilitates this interaction by performing the lookups and identifying the peers storing the files. Certainly, this approach provides an efficient data searching. However, the central server, which is a single point of

failure, renders this architecture inherently unscalable and vulnerable to malicious attacks. A well-known example of hybrid decentralized system is *Napster* (2011).

2. **Purely Decentralized:** In these networks (Fig. 3.2(b)), there is no central coordination and all peers have equal rights and responsibilities. Each peer can issue queries, serve and forward queries to his/her neighbours using a technique known as query flooding. This technique requires that a peer forwards a request to his/her neighbouring peers, who then forward that request to their neighbours, and so on. This approach provides peers dynamicity, and there is no single point of failure. However, guaranties on lookup efficiency cannot be provided since peers' knowledge about the system is limited to their neighbours. Moreover, the queries flood the system, and leads to scalability problems. Representative examples of pure decentralized P2P systems are *FreeHaven* (1999) and *gtk-Gnutella* (2000).

3. **Hybrid P2P:** In hybrid P2P systems (also referred to as super peer P2P systems), the concept of super peer is used to introduce hierarchy into the system. The hybrid P2P systems employ some super peers with higher capabilities to act as locally centralized index servers to their surrounding peers (leaf peers) and proxy content on behalf of these peers (Fig. 3.2(c)). The super peers are connected to each other in a decentralized fashion. This approach provides an efficient content lookup. The super peers do not constitute a single point of failure for a system, since they are dynamically assigned and, if they fail, the system automatically takes action to replace them with other super peers. However, the management, specifically, the assignments of a peer's rank (whether as a super peer or a leaf peer), must be managed by a bootstrap peer. Thus, construction and maintenance of the overlay network is left in the hand of the bootstrapping peer to account for the dynamic flow of peers. A representative example of partially centralized system is Gia (Chawathe, Ratnasamy, and Breslau (2003)).

Much research work is done by the researchers in order to improve the performance of unstructured P2P systems by insuring availability (Cuenca-Acuna, Martin, & Nguyen, 2003), reducing lookup costs (Cholvi, Felber, & Biersack, 2004) and reducing network traffic (Akbarinia, Pacitti, & Valduriez, 2006).

3.2.1.2 Structured P2P

In order to solve the problems that come across in unstructured P2P systems, a structure is introduced into a P2P network. The structure is introduced by controlling the overlay topology and the content placement. The content placement is under the control of certain predefined strategies (generally a distributed hash table, DHT). In other words, there is a mapping between a content (e.g. file identifier) and a location (e.g. peer address). These systems provide a guarantee (precise or probabilistic) on search cost. However, this is typically obtained at the expense of maintaining certain additional information (Vu & Ooi, 2010).

A DHT provides a hash table interface with primitives *put*(*key*, *value*) and *get*(*key*), where *key* is an object identifier, and each peer is responsible for storing the values (object contents) corresponding to a certain range of *key*s. Each peer also knows a certain number of other peers, called neighbours, and holds a routing table that associates his/her neighbours' identifiers to the corresponding addresses. Most DHT data access operations consist of a lookup for finding the address of the peer that holds the requested data, followed by a direct communication with that peer. Since a peer is responsible of storing the values corresponding to a range of keys, the autonomy is limited. Furthermore, DHT queries are limited to exact keyword search (e.g. file identifiers). Some examples of DHTs are Chord (Stoica et al., 2001), Pastry (Rowstron & Druschel, 2001) and Content-addressable network (CAN) (Ratnasamy et al., 2001). Few examples of P2P systems that employ DHT are Freenet (I. Clarke et al., 2002) and OneSwarm (Isdal et al., 2009).

Improving the DHT functionalities such as complex querying and availability is an active research area. For example, Roncancio et al. (2009) describe solutions for declarative querying support and query optimization in DHT-based P2P systems and identify important future research trends in these systems.

3.2.2 P2P Applications

In this section, different types of P2P applications are presented.

3.2.2.1 Content Distribution

Most of the current P2P systems fall within the category of content distribution, which includes systems and infrastructures designed for the sharing of digital content between peers (users).

P2P content distribution systems range from simple file sharing applications to more sophisticated systems that create a distributed storage medium for secure and efficient publishing, indexing, searching, updating and retrieving data. Some examples of P2P content distribution systems are: *FreeHaven* (1999), *gtk-Gnutella* (2000) and *eDonkey2000* (2000).

3.2.2.2 Content Streaming

Streaming is a technology used for delivering multimedia content among users on the Internet. With this technology, the user can playback the media content without waiting for the entire media file to be downloaded. Thus, streaming allows real-time transmission of multimedia over the Internet. Streaming applications over P2P systems have gained an enormous popularity. The power to accommodate large amounts of users, together with resilience to churn, reliability and low cost, are some of the reasons why P2P content streaming systems are preferred over dedicated servers or content distribution systems. P2P systems are considered a promising solution for video streaming to a large number of users (Alstrup & Rauhe, 2005; Bracciale et al., 2008). Examples include *SopCast* (2008) and *PearStreamer* (2013). These approaches typically build an overlay on the peers, but they differ in the techniques of constructing and managing the distribution network. The distribution network can be a tree (Castro et al., 2002), a mesh (Kostić et al., 2003) or a forest of trees (Castro et al., 2003).

3.2.2.3 Distributed Computing

In distributed computation applications, the resource of interest is the CPU processing power. The peers independently process pieces of a huge computational problem that requires an enormous amount of CPU processing. This is achieved by breaking down a computer intensive task into small work units, distributing them to different peer computers, which execute their corresponding work unit, and return the results. Several distributed cycle sharing projects are currently running over the Internet. For instance, the *Seti@home* (1999) (Search for Extraterrestrial Intelligence) project aims at analyzing signals received by radio telescopes to determine whether an intelligent life exists outside the Earth. Another example is the Genome project proposed by Larson, Snow, and Pande (2003), which tries to study and understand the human genetic information. In each of these projects, there is a centralized manager that distributes work to, and collects results from, peers.

3.2.2.4 Communication

The systems under this category provide the infrastructure for facilitating direct, usually real-time, communication and collaboration between peer computers. Some representative examples are *Skype* (2003) and *Jabber* (2008).

3.3 Security in P2P Content Distribution Systems

In the literature, many systems can be found that propose to solve the copyright infringement problem in P2P systems. In this section, these systems are categorized into two main classes: systems focusing on content protection (cf. Section 3.3.2) and systems focusing on traceability mechanisms (cf. Section 3.3.3).

3.3.1 Guaranteed Security Properties for a Content Provider

Internet content distribution first commenced with web caching and caching infrastructures. Then, *Akamai* (1998) turned caching into a service for content providers, and Content Distribution Networks (CDNs) became one of the most important advances in Internet technologies. In recent years, P2P systems have emerged as a new paradigm for content distribution. The popularity of P2P systems is attested by the fact that, in some countries, P2P traffic accounts for more than 60% of the overall Internet traffic, and a great deal of this traffic is generated by P2P content distribution systems (García-Dorado et al., 2012). However, today's P2P content distribution systems are severely abused by illegal re-distributions. Thus, in order to counteract the threat of unauthorized copying and distribution of media files over P2P systems, the media industry has sought protection from technological solutions, which are designed to prevent copyright infringement and illicit dissemination of protected works. Technological protections could take many forms and serve many related purposes. Some of these protection mechanisms are multimedia encryption, DRM, watermarking and fingerprinting.

Many P2P content distribution systems are proposed by the researchers to satisfy the needs of the content providers. The content providers typically search for the following guarantees in these systems:

- **Copy Prevention:** No additional replication is allowed other than the permitted copies.

- **Usage Monitoring:** All usage information of content by the users is recorded or communicated to the content owner.
- **Data Confidentiality:** Limited information access and disclosure to authorized users only.
- **Data Integrity:** Data is not altered during transmission between users due to malicious attacks.
- **Copyright Protection:** Associating digital rights with the content by embedding metadata or a watermark into the contents so as to express the rights of a party for that content.
- **Resistance against attacks:** The inserted watermark or a fingerprint into the content, provides robustness against signal processing or collusion attacks, respectively.
- **Traceability:** Ability to trace and identify the copyright violator.

The existing P2P content distribution systems can be categorized into two types: systems that focus on content protection without traceability and systems that provide content protection with tracing mechanisms.

3.3.2 P2P Content Distribution Systems with Content Protection

Various researchers have devised mechanisms to address content protection in P2P systems. Following are some P2P content distribution systems to protect digital copyrights.

Y. Y. Chen et al. (2009) proposed a DRM mechanism for a Bit-Torrent-like P2P system, which provides an end-to-end content secrecy and a transaction mechanism with confidentiality of data communication. In their scheme, the original files are divided into many pieces, and each piece is encrypted and licensed to defend illegal access. The users obtain the decryption keys after paying the content owners though a trusted payment gateway. The security analysis of the system shows that the content distributed via Bit-Torrent is securely protected and provides resistance against colluding and passive attacks.

Tsolis et al. (2011) proposed a P2P sharing system, which not only allows digital content exchange, but also supports copyright protection and management through a watermarking technology. The system is proposed mainly for digital images, and tracks all the watermarked image files, which are distributed and copied through the P2P network. The detection of multiple watermarking keys is managed by a novel decentralized lookup algorithm which allows effective

key detection in an optimal number of hops.

Stenborg, Herberthson, and Forchheimer (2011) proposed two schemes for the distribution of individually watermarked content to P2P users. The first method, shared fragment watermarking (SFW), uses encrypted shared fragment packets and recipient unique sets of decryption keys to achieve the individual watermarks, thus enabling the content distributor to efficiently transmit the content to many recipients. The second method, client-based watermarking based on Modified Transform Watermark (MTW), uses transformation scrambled content for the distribution and recipient unique keys to simultaneously descramble and watermark the content on the client side. MTW is presented as a secure scheme against collaborating attacks as the same information is always distributed. Similarly, SFW is also considered secure against a collaborating attack, since the extraction of a watermark requires more details about the embedding method. The robustness of both SFW and MTW is discussed without experimental details.

Inamura and Iwamura (2014) proposed a license management system for a content delivery over a P2P network. In the proposed scheme, entities of license administrator class do not need to administer a user key and a content key. The license administrator issues the license with a small amount of network resources and computational power. The system is based on a separate delivery model, in which a user can send encrypted content to another user over the P2P network, and the other user can decrypt the received content by using the license information obtained from the license administrator. The license administrator only manages one master secret and only re-binds a content key to the target user on the request basis. The security analysis of the system shows that the proposed system is resistant to communication attacks, namely, replay attack and passive wiretapping.

3.3.3 P2P Content Distribution Systems with Traceability

Megías and Domingo-Ferrer (2014) introduced a novel concept of a recombination fingerprinting mechanism for P2P content distribution. The proposed scheme uses the fingerprinting concept to provide identification to the copyright owner, offers collusion resistance against dishonest buyers trying to create a forged copy and detects copyright violators. Also, the users can preserve their privacy as long as they do not get involved in illegal content re-distribution. The security analysis of the system demonstrates that the communication between the buyers in the system is secure against man-in-the middle attack, and the utilized watermarking scheme for embedding a fingerprint is robust against common signal processing attacks.

Megías (2014) proposed an improved version of the automatic recombined fingerprinting

TABLE 3.1: Summary of presented P2P content distribution systems

P2P Sytems	Focus on:	Depends on:	Objectives
Chen et al. (2009)	Copyright protection	Bit-Torrent	Confidentiality of data communication, end-to-end content secrecy, immune to passive attacks, compromised peers and collusion between peers.
Tsolis et al. (2011)		ART-P2P	Efficient digital rights protection and management.
Stenborg et al. (2011)		Own routing protocol	Efficient distribution of individually watermarked content and confidentiality of data during transmission.
Inamura & Iwamura (2014)		Own routing protocol	Lightweight, scalable and secure content delivery system.
Megías & Domingo-Ferrer (2014)	Copyright protection and tracing of an illegal re-distributor	Own routing protocol	Digital content protection, tracing unlawful re-distributors, revocable privacy to buyers, collusion resistance against dishonest buyers, buyer frameproofness and communication confidentiality.
Megías (2014)		Own routing protocol	Digital content protection, efficient traitor tracing of illegal re-distributors, revocable privacy to buyers, collusion resistance, buyer frameproofness, mutual anonymity for merchant and buyers and communication confidentiality.
Domingo-Ferrer & Megías (2013)		Own routing protocol	Digital content protection, tracing illegal re-distributor, anonymity of honest users and co-utility.
Li et al. (2010)		Bit-Torrent	Secure, scalable, practical and robust music dissemination system and identification of copyright violator.
Li et al. (2010)		Own routing protocol	Secure, practical, scalable and robust system with efficient distribution of a large-sized multimedia content.
Gao et al. (2010)		Own routing protocol	Protection against unauthorized access and identification of copyright violator.

mechanism in which malicious proxies are considered in the fingerprinting protocol. A four-party anonymous communication protocol is proposed to prevent malicious proxies to access clear-text fingerprinted content. The proposed scheme provides a convenient solution for the legal distribution of multimedia contents with copyright protection whilst preserving the privacy of buyers, whose identities are only revealed in case of illegal re-distribution. Moreover, the scheme uses standard database search for traitor tracing unlike the automatic recombined fingerprint-based P2P content distribution system, which requires an expensive graph search to identify an illegal re-distributor. The experimental results of the proposed scheme show that the

system provides an efficient and scalable content distribution in P2P networks, collusion resistance and an efficient traitor tracing of illegal re-distributors.

Domingo-Ferrer and Megías (2013) proposed a P2P protocol for distributed multicast of fingerprinted content in which cryptographic primitives and a robust watermarking technique are used to produce different marked copies of the content for the requesting user such that it can help the provider to trace re-distributors. Moreover, the authors used rewards and punishment concepts of game theory to ensure that peers rationally cooperate in P2P fashion for fingerprint embedding and content distribution. The fingerprinting scheme used in the distribution of fingerprinted content guarantees buyer frameproofness and collusion resistance.

J. S. Li et al. (2010) proposed a DRM enabled P2P architecture, which provides secure distribution of copyright-protected music contents and efficient tracing of unauthorized users. The RSA public-key cryptosystem is used to generate a unique digital fingerprint for every user within the network. The fingerprint is embedded within the music file in a protected form, such that the music provider can establish the identification of any user performing an unauthorized distribution of the file. Moreover, the robustness of the fingerprint toward deliberate attack by a malicious user is improved via the use of an error-correcting code polling technique. The experimental results confirm the ability of the system to achieve an efficient and robust distribution of MP3 music files with no discernible degradation in the quality of the music content.

A fingerprint generation and embedding method is proposed by X. Li et al. (2010) for complex P2P file sharing networks for copyright protection. In this system, wavelet transforms and principal component analysis (PCA) techniques are used for fingerprint generation. The wavelet technique provides a scalable approximation matrix that contains the most important low-frequency information and the PCA technique determines the orthogonal eigenvectors, which makes it possible to maximally distinguish the different fingerprints. The proposed scheme is scalable since it is able to generate a large number of unique fingerprints. Furthermore, the media producer keeps the mapping between the fingerprint and the user, and therefore, is able to trace the fingerprint for a pirated content. The experimental results of the scheme prove the robustness of the unique fingerprint against most common attacks such as Gaussian white noise, lossy compression, median filter and border cropping.

Gao et al. (2010) proposed a fingerprinting scheme that is suitable for a P2P network. In the proposed scheme, the distributor divides the media content into two parts: a demo clip that is not encrypted, so that the users can know that the content they are requesting is indeed what they want, and the other is an encrypted part. Both parts are published in a P2P system, where each

user obtains a unique decryption key from the distributor according to their peer IDs. When decryption with different decryption keys are performed on the same encrypted content, the result is a slight different fingerprinted copy for each user. The embedded ID in the decrypted digital copy is used to trace the traitors. The experimental results show that the scheme is robust and resistant against average collusion attacks.

Table 3.1 gives a summary of the presented P2P systems. In Section 3.5, these systems are compared according to the security techniques used and the guaranteed security properties.

3.4 Privacy in P2P Content Distribution Systems

Providing privacy in P2P systems is a challenging task due to their open and autonomous nature. In addition, an incorporation of content protection mechanisms in P2P systems affects the privacy interests of users: the fact that a tracing mechanism makes use of a systematic record which details what multimedia files are downloaded through a specific IP address, the history of files shared or downloaded, or a list of the peers with whom a user has interacted in the past, ultimately disrespects the private space of the user. In this context, providing privacy to data owners' is a challenge. The literature review shows that a few researchers have worked on P2P content distribution systems that provide preservation of content providers' ownership properties and end users' privacy so far. The existing systems are classified into two categories: systems with content protection, data and user privacy (anonymity) (cf. Section 3.4.2) and systems with a focus on either user privacy or data privacy (cf. Section 3.4.3).

3.4.1 Guaranteed Privacy Properties for End Users

Grodzinsky and Tavani (2005) highlight the absence of privacy in P2P systems by noting that the user reveals his/her details, such as plain-text queries and IP addresses, to another user (provider) that provides the services when downloading files. Furthermore, Walkowiak and Przewozniczek (2011) and Y. Wang et al. (2011) highlight that a great deal of information regarding the user preferences can be collected in a content distribution by tracking the user activities at the provider side, thus compromising the user's anonymity. For example, in 2009, it was discovered through a P2P sharing system, that an IP address in Iran had obtained blueprints

and the avionics package for Marine One (the U.S.s president's helicopter) (Mack, 2009). Consequently, the end users search for two types of privacy guarantees in P2P content distribution systems:

1. **Data privacy guarantees**
    - Data is only available to authorized requesting users.
    - The communication channel used for transferring the data between two users is protected against malicious attacks.
2. **User privacy guarantees**
    - The real-identity of a user is protected from being revealed to any other entity in the system, except in case of being found involved in a malicious act.
    - The online behavior of the users is unlinkable to their real identities.
    - Users are protected against identity theft.

Data privacy can be protected by techniques such as anonymous communication, anonymous authentication, data encryption and digital checksums. On the other hand, user privacy can be protected by using different anonymity methods such as pseudonymity, trust management and cryptographic techniques.

3.4.2 Secure and Privacy-Preserving P2P Content Distribution Systems

In this section, P2P content distribution systems are presented, that focus on multimedia security, data privacy and anonymity.

Megías and Domingo-Ferrer (2014) proposed a secure and privacy-preserving P2P content distribution system based on an automatic fingerprint recombination. In the system, the buyers (users) can preserve their privacy as long as they do not get involved in illegal content re-distribution. A P2P proxy (or set of proxies) uses an onion routing-like solution to create anonymous connections between the buyers, such that source (parent) and destination (child) buyers do not lose their anonymity. Pseudo-identities are used to protect the real identities of the end users, and public-key cryptography is used to protect the hash of the fingerprint and the public keys. Symmetric cryptography is used to provide data confidentiality during an anonymous file transfer from the parent buyer to the child buyer.

In the system based on automatic recombined fingerprinting (Megías & Domingo-Ferrer, 2014), the traitor-tracing protocol requires an expensive graph search and disturbs a few honest buyers who must co-operate with the authority to identify the source of an illegal re-distribution. Moreover, the scheme of Megías and Domingo-Ferrer (2014) is vulnerable against malicious proxies, who may even collude with other parties, such as, the merchant, to frame an innocent buyer. Megías (2014) proposed an improved version of an automatic fingerprint recombination scheme to overcome the above mentioned drawbacks. The improved system achieves a more efficient and practical system, especially as traitor tracing is concerned, since it avoids the situations in which illegal re-distributors cannot be traced with the proposal of Megías and Domingo-Ferrer (2014). Furthermore, with the proposed four-party anonymous communication protocol, better security properties against potentially malicious proxies are obtained. In addition, a P2P proxy (or set of proxies) uses an onion routing-like approach likewise to Megías and Domingo-Ferrer (2014) to create anonymous communication between the merchant and the seed buyers, and between peer buyers within the P2P distribution system. The scheme avoids homomorphic (or any type of public key) encryption of the multimedia content and restricts the usage of the public-key cryptography to the encryption of short binary strings such as fingerprint segments or hashes. The fragments of the content transferred from the parent buyer to the child buyer in a distribution protocol are protected with a symmetric-key encryption.

Domingo-Ferrer and Megías (2013) proposed a P2P protocol for distributed multicast of fingerprinted content. In the proposed framework, the content provider can trace re-distributors without affecting the privacy of honest users. The group signatures used in the system provide anonymity and unlinkability to the end users. In order to preserve the privacy of the input and output information in each execution of the fingerprinting scheme, a secure two-party computation protocol, is used as a building block of the anonymous fingerprinting protocol.

M. K. Sun et al. (2009) proposed an identity-based DRM system with privacy enhancement. Their DRM system retains user privacy by hiding the relationship information between users and the digital contents the users own. In order to provide strong privacy and anonymous consumption, restrictive partial blind signatures are adopted in the system. Moreover, a content key management protocol is proposed in the system to protect the users against malicious servers, and prevent them from obtaining a complete content key.

Win and Emmanuel (2011) proposed a privacy-preserving content distribution mechanism without requiring trust over any third party by using the mechanisms of blind decryption and one-way hash chains. In the system, a privacy-preserving revocation mechanism preserves a user's anonymity, even if the user has been blocked for his/her misbehavior. The proposed

scheme provides privacy protection to the users by generating an anonymous token set. The user only interacts in the system with other entities with these anonymous tokens. The security analysis of the system shows that the proposed scheme is resistant to collusion of the involved parties. In addition, the system is not prone to the oracle problem of the blind decryption mechanism.

3.4.3 Privacy-Preserving P2P Content Distribution Systems

In this section, P2P content distribution systems focusing either on data privacy or user privacy are presented.

Yu, Lee, and Ramakrishnan (2011) proposed a P2P protocol, Nemor, which not only allows a requesting user and a serving user (provider) to communicate anonymously with each other and from other participating users, but also protects the content being exchanged. Nemor relies on a trusted intermediary, such as a provider-managed tracker, to identify a potential serving peer. The tracker tracks the membership information of a peer and the objects stored by that peer. It uses a combination of random walks and flooding to build a path between the requesting peer and the serving peer. The experimental results show that Nemor is resilient to traffic analysis attacks that are aimed to break the anonymity of the users.

The Peer-to-Peer Personal Privacy Protocol (P^5) proposed by Sherwood, Bhattacharjee, and Srinivasan (2002) uses a hierarchical broadcasting technique to achieve mutual anonymity between users. For different levels of the hierarchy, different levels of anonymity are provided. The user has the flexibility to decrease his/her level of anonymity in order to increase his/her performance. The proposed system also provides sender and receiver anonymity by transmitting encrypted packets at a constant rate to all participants. The security analysis shows that P^5 is invulnerable to common communication attacks such as denial-of-service, correlation and coalition attacks.

The protocol proposed by Lu et al. (2007) uses an anonymous zero-knowledge authentication protocol to support trust management such that users can use unforgeable and verifiable pseudonyms instead of their real identities. The PseudoTrust model enables pseudonym-based trust management so that the real identities of peers are protected during the authentication. It also anonymizes the communication between two peers by adopting an anonymous communication technique (onion routing) within the model. In the authentication protocol, the Diffie-Hellman key exchange protocol is incorporated to provide confidentiality and integrity to data exchanges such that, after authentication, both peers can share a session key for encrypting the

TABLE 3.2: Summary of presented P2P content distribution systems

P2P Sytems	Focus on:	Depends on:	Objectives
Megías (2014)	Multimedia security and privacy protection	Own routing protocol	An efficient and scalable digital content protection, tracing unlawful re-distributors, revocable privacy, mutual anonymity, collusion resistance, buyer frameproofness, real identity protection, data integrity and communication confidentiality.
Megías & Domingo-Ferrer (2014)		Own routing protocol	Digital content protection, tracing unlawful re-distributors, revocable privacy to buyers, collusion-resistance against dishonest buyers, buyer frameproofness, real identity protection, data integrity and communication confidentiality.
Domingo-Ferrer & Megías (2013)		Own routing protocol	Digital content protection, tracing illegal re-distributors, anonymity of honest users, co-utility and data confidentiality.
Win et al. (2011)		Own routing protocol	Digital rights protection, revocable anonymity, accountability without relying on any third party and resistance against collusion of malicious users.
Sun et al. (2009)		Own routing protocol	Digital rights and user privacy protection, regional content restriction, anonymous content consumption and protection against malicious servers.
Yu et al. (2011)	Privacy protection	Own routing protocol	Efficient content search and delivery, anonymous communication, data protection, congestion-avoidance mechanism and resistance against communication attacks.
Lu et al. (2007)		Gnutella	Identity protection, anonymous communication, anonymous authentication, trust management and resistance against communication attacks.
Sherwood et al. (2002)		Own routing protocol	Scalable anonymous communication and protection against communication attacks.

exchanged data. The security analysis of the PseudoTrust model demonstrates its ability to defend against communication attacks, namely, man-in-the-middle attacks, replay attacks, denial of service and collaborated attacks.

Table 3.2 gives a summary of the presented P2P systems.

3.5 Comparison of P2P Content Distribution Systems

In this section, the content distribution systems presented in Section 3.3 and Section 3.4 are compared. The comparison focuses on the used security (cf. Section 3.5.1) and privacy (cf. Section 3.5.2) techniques, and the guaranteed security and privacy properties (cf. Section 3.5.3).

3.5.1 Comparison in Terms of Security Techniques

Here, the P2P content distribution systems presented in Section 3.3 and Section 3.4 are compared with respect to DRM, watermarking and fingerprinting techniques.

3.5.1.1 DRM

The use of a DRM is to guarantee that the content is only accessible by an authorized user with a valid license issued from the license creator.

A system proposed by Y. Y. Chen et al. (2009) enables a large-scale distribution of copyrighted digital content in P2P networks. There are four entities in the proposed system: peers, original peers (the content provider and the license issuer), a tracker site and a customer-to-content provider (C2C) payment gateway. In the system, there are four stages: initial, content blocks distribution, purchase and content decryption. In the initial phase, the content is first divided into many pieces, and each piece is encrypted with a content encryption key by the original peer. Then, the serial number of the content, the total length of all the cipher blocks, the hash value of all cipher blocks, the URLs of the selected tracker sites and the original peer's identity are recorded in the seed. The seed is forwarded to anyone in the system to download this content with an assistant of the tracker site. In the content block distribution phase, the peer with a seed can follow the instruction to find the tracker site. The tracker site provides the list of the peers which hold some blocks of the content. Then, the peer can download each block of the content through P2P interaction. In the purchase phase, after downloading all the cipher blocks, the downloading peer obtains the corresponding license from the original peer through a trusted C2C payment gateway. As the C2C payment gateway confirms the payment from the downloading peer, it acknowledges the original peer to release the corresponding license. The downloading peer can then use this license to decrypt the cipher blocks in the decryption phase.

In the DRM system proposed by Inamura and Iwamura (2014), a user can send the encrypted content to the other user over a P2P network, and the other user can decrypt the received content by using the license information obtained from the license administrator. There are two classes in the system: a license administration class and a user class. The administration class is a group of three servers: content provider, content key issue server and content administration server, and the user class is a network of connected users. A user can obtain the content file in two ways: either directly from the content provider or from the other peers in the system (a content file consists of a content key, a license data (user key and meta-data) and content). In

the content file, each data is encrypted with an encryption key, where only the entity that has a master key can decrypt and obtains all data in the content file. In case a user obtains a content file directly from the content provider, he/she obtains the content key from the content key issue server after a payment. The user can then decrypt the content key with his/her user key and decrypts the content with the decrypted content key. For distribution of the content in the system, a user sends the content file to the other users, after re-encrypting the content file with a temporal user key. The other user who receives the content file sends a request to a content key issue server to re-encrypt the content key in the content file with his/her user key for the purpose of using the content. Once the receiving user receives the re-encrypted content key from the server, he/she is allowed to use the content.

The DRM-based P2P system proposed by J. S. Li et al. (2010) comprises a single music player (MP) and multiple peers. In the DRM module of the system, the peer application embeds a unique digital fingerprint (FP) generated by an RSA cryptosystem into the music file compiled by each user. As a result, if a user subsequently disseminates the music file without first obtaining authorization to do so, MP can identify the user by retrieving the fingerprint from the music file and can then take the appropriate legal action. The DRM framework comprises two major phases, namely, the FP generation phase and the FP embedding/protection/checking phase. In the FP generation phase, the downloading peer (DP) first sends a request for a protection key pair to MP, who responds with two key pairs: the FP-encryption key pair and the password PW-encryption key pair. Once DP receives the two key pairs, he/she uses his/her seed to produce the FP-generated key. The FP-generated key consists of a public key and a private key. The user uses his/her private key to generate FP and encrypts FP using his/her public key. The user then embeds the encrypted FP into the MP3 file using his/her PW. PW is encrypted by a public key of the user. The user then sends the encrypted FP, the encrypted PW, the user ID and a public key to MP. MP then uses the private keys of the FP-encryption key pair and the PW-encryption key pair to decrypt the encrypted FP and PW. MP then stores FP, PW, the public key and the user ID of the DP in the database for DRM purposes.

The ticket-based DRM system proposed by M. K. Sun et al. (2009) provides a flexible and secure DRM model, in which the user is able to play the digital content following the usage rules enforced by the DRM client-controller. In this system, there are three major roles and one major item, namely, the producer (creator of the content), the distributor (collection, license, content and subscription servers), the user and a ticket (used for enabling anonymous consumption). There are three phases in the proposed DRM model: upload, ticket and play. In the upload phase, a raw content is uploaded by the producer to the collection server encrypted

by the content keys. Then, the collection server management system uploads the usage rules to the license server, which is responsible for generating the content license to respond to the requests from the subscription server. The collection server is also responsible for uploading the encrypted contents to the content server. The content server responds to content download requests from the subscription server. In the ticket phase, the user builds a session key with the subscription server and sends the request for purchasing a ticket, which can be used to obtain the content licenses. In the last phase, i.e. the play phase, the user with the ticket sends a request to the license server for the corresponding content key to play the digital content. The content key is divided into two parts: CK_1 and CK_2. CK_1 is encrypted with the public key of the content server and CK_2 is encrypted with the public key of the license server. The content server decrypts the encrypted CK_1 and encrypts it with the user's temporary public key. Similarly, the license server decrypts CK_2 and encrypts it again with a user's temporary public key. The license server provides both encrypted CK_1 and CK_2 to the user according to the ticket value. In addition, the user can only obtain a regional license associated with the region encoded on the ticket. The region code ensures that DRM players can only play protected content embedded with the correct region code.

The DRM-based content distribution system proposed by Win and Emmanuel (2011) provides security, revocable privacy and accountability without a need of any trusted third party. The system consists of three entities: the content owner, multiple levels of content providers and the end users. The content owner generates anonymous token sets and is responsible for the registration of legitimate users and the revocation of malicious users. The content provider is a software agent at the distributor side that performs the content purchase transactions with the users and tracks the usage patterns of the users. Before registration with the content owner, each user obtains a token set from the owner anonymously by paying, and uses it for the content purchase. However, the obtained token cannot be used unless the user obtains the decryption key K_j. A user performs the authentication with the content owner as a qualified user with his/her real identity credentials at the registration stage, since the content owner only performs a blind decryption protocol with registered users. Through a blind decryption protocol, the user obtains K_j to decrypt the encrypted anonymous token sets. The user then uses the anonymous token set for each transaction with the content provider. The tokens are bounded with the DRM agent at the user side using the seal storage function of the Trusted Platform Module of the client device. A user first downloads the content from the content server of the content provider, and then obtains the license using an anonymous token.

3.5.1.2 Digital Watermarking

The goal of digital watermarking is to evade other parties from claiming the copyright by embedding the information that identifies the copyright owner of the multimedia data, and also to provide assurance that the origin of the content is authentic, and its integrity can be proved.

In a decentralized P2P system proposed by Tsolis et al. (2011), the copyright protection and management is obtained through a watermarking technology. A robust multi-bit watermark is embedded into an image by casting several zero-bit watermarks onto specified coefficients obtained by a frequency transform technique. Thus, the watermark that is embedded into the image is not a single sequence but many different sequences generated with different seeds. These sequences are casted, one after the other, on the mid coefficients of the image using the additive rule. Every single random sequence of Gaussian distribution is generated using a different number as the seed for the Gaussian sequence generator. It is important to differentiate the sequences in order not to mislead the detection mechanism, since it is based on the correlation between the extracted sequence and the sequence produced with the watermark key. At the same time as of casting watermarks to the images, the watermarking keys are being stored in the independent network peers of Autonomous Range Tree (ART) system. The copyright status of each digital image can be retrieved and evaluated rapidly via the ART P2P system.

In one of the two schemes proposed by Stenborg et al. (2011) for distribution of individually watermarked content, there can be two scenarios in distribution of the watermarked copies to the end users. In the first scenario, the recipient A accesses the data through the P2P network and receives all the watermarked packets (both the ones he/she needs and those that he/she cannot decrypt). All the watermarked packets are stored in an encrypted form at A, so that the other recipients in the P2P network can access them without obtaining the original source. In the second case, the recipient A accesses the data through the P2P network, but he/she is only given the watermarked packets that he/she is able to decrypt. The transmitted content data is twice as much in scenario 1 as compared to case 2. If bandwidth is not a problem in the network, case 1 can be preferred. If the access to the distributor is limited, then also case 1 is the best. If, instead, bandwidth in the network is limited, then case 2 might be preferred, particularly when the distributor is easy to be accessed. The second method is a client-based watermarking method originally created for video broadcast distribution. In a client-based watermarking, the content is scrambled by the distributor. The same information is distributed to all the recipients. For P2P distribution, it is necessary to store the scrambled video at the recipient that has accessed

the video. Since all recipients use the same scrambled video content, the data are easily distributed from one recipient to another. For the distribution of the individual descramble keys, a direct secure connection between each recipient and the distributor is used. The size of a set of descramble keys, the key length and frequency of changing a key, are changed often due to security and transmission issues.

3.5.1.3 Digital Fingerprinting

The watermarking algorithm can be used to prove content ownership but it is unable to deal with content leakages, i.e. cases where a buyer may re-distribute the received content to other unauthorized customers. This deficiency of watermarking scheme inspires a lot of research works in digital fingerprinting. In digital fingerprinting, if an unauthorized copy of the content is recovered, an extraction of the fingerprint will reveal the identity of the copyright violator.

The P2P content distribution system proposed by Megías and Domingo-Ferrer (2014) uses the fingerprinting concept to provide identification to the copyright owner. In the system, the merchant originates only a set of M seed copies of the content with different pseudo-random binary fingerprints and sends them to M seed buyers. The merchant or some trusted authority keeps the association of the first M fingerprints with the identities (or maybe some pseudonyms) of the first M buyers. All subsequent copies are generated from the seed copies. Each non-seed buyer obtains his/her copy of the content by running a P2P purchase software tool. The copy obtained by each buyer is a combination of the copies provided by his/her sources (parents). The fingerprint of each buyer is thus a binary sequence formed as the combination of the sequences of his/her parents. Whenever a buyer obtains fragments of the content from another buyer, the transaction record is sent to a third party "transaction monitor". The purpose of storing the transaction records at the transaction monitor is to enable tracing of illegal re-distributors. The transaction record does not specify which fragments come from which buyer, so that the privacy of the buyers' fingerprints is preserved. In addition, the transaction monitor only records an encrypted hash of the whole fingerprint of each buyer, thus preventing a possible coalition of the transaction monitor with the merchant or other buyers. The graph-based backtracking algorithm is designed to identify an illegal re-distributor. In the proposed system, collusion-resistance is obtained by a 2-layer collusion-resistant coding of the fingerprints: segment-level code (the anti-collusion code is used for the segments of the fingerprint) and hash-level code (the anti-collusion code is used for the hash of the fingerprint).

The system proposed by Megías (2014) is derived from a privacy-preserving P2P system

based on recombined fingerprints of Megías and Domingo-Ferrer (2014). This new version incorporates significant improvements in the distribution and traitor-tracing protocols to achieve an efficient and practical system. The merchant distributes the copies of the content legally to the M seed buyers. All subsequent copies within the system are generated from M seed copies. The non-seed buyers obtain their copies through a P2P purchase software application. The copy obtained by each non-seed buyer is a combination of the copies provided by his/her sources (parents). The fingerprint of each buyer is thus a binary sequence formed as the combination of the sequences of his/her parents. To provide collusion resistance, a two-layer anti-collusion code (segment level and fingerprint level) is used. The modified distribution protocol involves four parties, namely, a parent buyer, a child buyer, a transaction monitor and a proxy. The proxies are used to provide anonymous communication between a parent and child buyer. In the modified distribution protocol, a transaction monitor acts as a temporary key database to prevent the proxies from accessing the symmetric keys used for encrypting the distributed content. The transaction monitor stores the symmetric session keys shared by each parent and a child buyer. The session key can be retrieved only once from its database implying that only a child buyer can access it. Once the key is retrieved, the transaction monitor blocks the register and eventually removes it. Thus, a malicious proxy trying to access the database in order to retrieve the key would be detected since the register containing the key would be blocked either to the proxy or to the child buyer, raising an investigation. The improved traitor tracing protocol of Megías (2014) does not require a cleartext of the fingerprints of honest buyers and is based on a standard database search, which is different from the graph-based backtracking algorithm of the system proposed by Megías and Domingo-Ferrer (2014).

Domingo-Ferrer and Megías (2013) proposed a P2P content distribution system based on an anonymous fingerprinting and game theory concept. The proposed fingerprinting scheme guarantees correctness, anonymity, unlinkability, buyer frameproofness, revocability and collusion resistance. In the fingerprinting scheme, there are three main entities: a registration center (RC), a merchant and a set of buyers (P^i). Each buyer (P^i) in the system engages in an anonymous fingerprinting with other buyer (P^{i+1}) such that P^{i+1} obtains a fingerprinted version ($D_{012\ldots i+1}$) of the original content D_0, and P^i obtains a transaction record $t_{i,i+1}$. P^i sends $t_{i,i+1}$ to P^0 (the buyer with the content D_0). Thus, P^0 has all the transaction records. In case a peer P^i fails to send the transaction record to P^0, P^0 together with RC, is able to obtain P^i's identity and thus P^i can be found guilty. When P^0 detects a re-distributed copy, he/she along with RC runs a re-distributor identification protocol to output an identity of the illegal re-distributor. In the distribution protocol, only P^0 has access to the original content D_0, whereas only P^1 knows

the secret information y_1, which is embedded in D_{01}. The same applies for the subsequent executions of the fingerprinting protocol. Thus, to preserve the privacy of the input and output information ($t_{i,i+1}$, $D_{01\ldots i}$, y_{i+1}, $D_{01\ldots i+1}$) in each execution of the fingerprinting scheme, a secure two-party computation protocol is required as a building block of the anonymous fingerprinting protocol. In the system, rewards and punishments based on game theory are introduced so that the buyers rationally co-operate in a P2P fashion and loyally follow the prescribed P2P multicast protocol to make the scheme co-utile.

A fingerprint generation and embedding method is proposed by X. Li et al. (2010) for complex P2P file sharing networks. In the system, the source file is first decomposed into two parts: a small-sized file and a large-sized file. The small-sized file carries the embedded unique fingerprint for each peer and is distributed using the traditional client-server mode, while the large-sized file is freely distributed in P2P networks. The P2P fingerprinting method employs a wavelet transform to model the low-frequency features of the image (obtained by using an Inter-frame of a DVD quality video), and PCA to further decompose it into eigenvectors. After the preprocessing, any vector can be adopted to generate one fingerprint. The approximation coefficients obtained by L-level wavelet transform are used to form a small-sized file, while the detail coefficients constitute a large-sized file. The small-sized file is then used to calculate the eigenvectors using PCA. A fingerprint matrix is calculated by multiplication of a product of the eigenvectors and a scale vector with a matrix (company logo or small part of a host image) provided by a source owner. The matrix provided by the source owner is used to prove the right ownership of the fingerprint. The content owner keeps the mapping between the fingerprint and the customer, and is therefore able to successfully trace back the fingerprint for a pirated content. To identify the embedded fingerprint, the owner decomposes the fingerprinted image using inverse L-level wavelet transform to obtain a fingerprint matrix. Then, the signs of the columns in this matrix are compared to the signs of each eigenvector using the Hamming distance. The eigenvector that has the minimum Hamming distance to the matrix is claimed as the embedded fingerprint.

In the copyright protection system proposed by Gao et al. (2010), each user of the P2P system obtains a slightly different version of the same content. In the system, the distributor divides the media content into two parts: unencrypted content to be used as a demo clip and an encrypted content. The distributor generates the encrypted content by embedding a high strength watermarking signal into the original content. Then both parts are published to a P2P system. An AND anti-collusion code is used to represent the corresponding peer ID. A unique

decryption that is generated from the peer ID and the corresponding watermarking signal is assigned to each user. When decryption with different decryption keys are performed on the same encrypted content, the result is a slight different fingerprinted copy for each user. In case that the content provider finds a pirated copy, he/she adopts hard-detection algorithm based on a correlation method to trace the traitors.

Table 3.3 presents the multimedia security techniques used in the compared P2P systems. In the table, a cell is marked with "No" when a security technique is not used by the P2P content distribution system.

3.5.2 Comparison in Terms of Privacy Techniques

In this section, the P2P content distribution systems presented in Section 3.3 and Section 3.4 are compared with respect to privacy protection techniques such anonymity, trust management and cryptographic techniques.

3.5.2.1 Anonymity Techniques

Anonymity techniques (cf. Section 2.4.1) are mostly used to make a user indistinguishable from other users, thus providing anonymity among a group of users.

In the system proposed by Inamura and Iwamura (2014), an anonymous communication channel is used for delivering the content to the users in such a way that the entities of license administrator class (content provider, content key issue server and content administration server), and third parties cannot learn anything about the channel. In the distribution phase, the seed of a user key is sent through the channel instead of a real identity of the user. Thus, user privacy about a channel is protected against entities of the license administrator class.

In the system of J. S. Li et al. (2010), revocable privacy is provided by either using a smart card or a combination of user name, MAC and IP addresses. In case that MP finds an unauthorized music file in the system, he/she can determine the identity of the misbehaving peer by looking into his/her database which contains the FP, the user ID and password of the users.

In the system proposed by M. K. Sun et al. (2009), full anonymity is provided by using restrictive partial-blind signature. In the restrictive partial-blind signature scheme, a temporary public key is embedded into the blind message, which contains partial information about the user. This provides anonymous consumption to the users. In the proposed model, a user can obtain a ticket from online stores, friends or any other distribution channel, and can present it to

TABLE 3.3: Comparison of presented P2P systems based on used security techniques

P2P Systems	DRM-based protection	Watermarking-based rotection	Fingerprinting-based protection
Megías (2014)	No	No	Yes, due to automatic recombination and collusion-resistant fingerprinted content distribution
Megías & Domingo-Ferrer (2014)	No	No	Yes, due to automatic recombination and collusion-resistant fingerprinted content distribution
Inamura & Iwamura (2014)	Yes	No	No
Domingo-Ferrer & Megías (2013)	No	No	Yes, through distributed multicast of collusion-resistant fingerprinted content
Win et al. (2011)	Yes	No	No
Tsolis et al. (2011)	No	Yes	No
Stenborg et al. (2011)	No	Yes	No
Yu et al. (2011)	No	No	No
Li et al. (2010)	No	No	Yes, through RSA-based-fingerprinted content distribution
Gao et al. (2010)	No	No	Yes, through collusion-resistant fingerprinted content distribution
Li et al. (2010)	Yes	No	No
Chen et al. (2009)	Yes	No	No
Sun et al. (2009)	Yes	No	No
Lu et al. (2007)	No	No	No
Sherwood et al. (2002)	No	No	No

the subscription server to obtain a license without revealing his/her real identity.

In the system proposed by Win and Emmanuel (2011), revocable privacy is provided to the

end users. In the registration phase, the user registers with the content owner using his/her real name. After registration, the user obtains an anonymous token set from the content owner which can be used in other transactions such as license acquisition. The users are accountable for the licenses they had purchased, and the usages of the license are tracked by the content owner with anonymous tokens. If a misuse of a license by a user is found, the anonymous token set of the user is retrieved and revoked by the content owner. In order to obtain a license from the content provider, a user sends his/her anonymous token set to the content provider, who verifies it and then sends the license anonymously to the user.

In the system proposed by Megías and Domingo-Ferrer (2014), anonymity is provided through pseudonyms. The merchant has access to the buyers' database only, which relates a given pseudonym to real identity data. Thus, a true identity of a buyer can be revealed by the merchant in case a user is found guilty of illegal re-distribution. In the proposed system, privacy is also maintained by using anonymous communications. Rather than transferring content directly from the parent buyer to the child buyer, data travels through proxy peers.

The system proposed by Megías (2014) uses pseudonyms to protect the real-identity of the buyer. The real identities of buyers are known only by the merchant. Thus, in case a buyer is found guilty of illegal re-distribution, a true identity of him/her can be revealed by the merchant. A proxy (or a set of proxies) provide anonymous communication between the parent and the child buyer by means of a specific protocol analogous to Chaum's mix networks. The content that is transferred over the proxy is encrypted using symmetric cryptography. The session key used for encrypting the content is shared by the parent and the child buyer using the transaction monitor as a temporary key database.

Domingo-Ferrer and Megías (2013) proposed to preserve revocable anonymity in the system using a group-signature scheme. Group signatures allows members of a group to create signatures anonymously, such that it is hard for an adversary, not in possession of the registration center's secret key, to recover the identity of the signer. The registration center is in charge of adding group members and has the ability to reveal the original signer in the event of disputes. Before obtaining the fingerprinted content from the merchant, a buyer undergoes a two-party protocol with a registration center to obtain a secret input against his/her identity. The registration center stores the secret information and the identity of the buyer, and thus can reveal the identity in case a buyer is found guilty of illegal re-distribution.

Nemor (Yu et al., 2011) allows a requesting peer and a corresponding serving peer to communicate anonymously with each other and from other participating peers, while protecting the identity of the content being exchanged. Nemor also relies on a trusted intermediary, such as a

provider-managed tracker, to identify a potential serving peer. A peer in Nemor can join one or more trees. The requesting and serving peer can dynamically construct an overlay path between them using a combination of a random walk, a probabilistic jump from one tree to another and constrained flooding.

In P^5 (Sherwood et al., 2002), full privacy is maintained by using anonymous communications. P^5 relies on users to generate and broadcast cover traffic to hide the real traffic. It groups users in a logical hierarchy of broadcast groups, arranged as a binary tree. P^5 logical broadcast hierarchy is a binary tree constructed using the public keys of each user. Each node of the tree consists of a bit-string of a specific length to represent its hierarchy level (in horizontal) and also the group (in vertical). When a message is sent to a broadcast group G_1, the message is also sent to all groups that are descendants of G_1 as well as all groups that have G_1 as a descendant. To send a message, a user first encrypts the message with the intended recipient's public key and then broadcasts the ciphertext to one of the broadcast groups the sender has joined. If the recipient is not in one of the sender's broadcast groups, the message can be anonymously broadcast across other groups in the binary tree. When a user does not have data packets to send, he/she sends noise, which is then propagated throughout the network in the same manner as data packets. This approach provides strong anonymity, since a passive eavesdropper cannot tell which packets contain data and which packets are noise.

In the PseudoTrust model (Lu et al., 2007), full anonymity is maintained by using pseudo-identities and anonymous communications. Each peer in the system generates a pseudo-identity (*PI*) and a pseudo-identity certificate (*PIC*). A *PI* is used to identify and replace the real identity of a peer in a P2P system. A *PIC* is generated to authenticate the *PI* holder. On joining the system, each peer constructs an anonymous onion route and finds tail nodes based on the Anonymous P2P File-Sharing (APFS) protocol. The APFS protocol provides mutual anonymity of the initiator and the responder in a connection. In APFS, each peer chooses a tail node and creates an onion route to it. This tail node serves as an entry point to the anonymous network for that peer. A tail node and other peers in the onion route do not know about the peer at the end point positions.

3.5.2.2 Trust Management Techniques

Trust techniques (cf. Section 2.4.2) provide privacy protection by handling the trustworthiness of users without revealing their true identities. The right to access data is given to peers who are trustworthy and forbidden to peers who are untrustworthy.

In the PseudoTrust model (Lu et al., 2007), a pseudonym-based trust management is proposed. The PseudoTrust model allows users to generate their pseudo names individually, and users do not depend on any third party to authenticate each other. The reputation of a peer is connected with the peer's pseudo-identity instead of his/her real ID or IP address. When an initiating peer *A* requests a particular content from a peer *B*, *B* obtains *A*'s credit based on the trust management mechanism to help him/her decide whether to act as a responder and provide the requested content. Similarly, when *A* obtains the content, he/she evaluates the content, and provides comments to the peers who provided the resource.

3.5.2.3 Cryptographic Techniques

Cryptography is largely used by P2P content distribution system in order to protect data from unauthorized access. As discussed in Section 2.4.3, cryptographic techniques include encryption algorithms, hash functions and zero-knowledge proofs. The encryption techniques are used to prevent unauthorized parties from reading private data. Hash functions are used to provide data integrity in conjunction with a digital signature scheme, and to protect real identities of a user by generating pseudo-identities. Zero-knowledge proofs can be used for identity verification in various authentication protocols.

- **Data Encryption:** Usually symmetric-key encryption is used to protect data content, since symmetric-key generation is less expensive than asymmetric-key generation. In P2P systems, hybrid encryption can prove to be useful for a large-sized data. Using hybrid encryption algorithms, the major portion of the work in encryption/decryption can be done by the more efficient symmetric-key scheme, while the asymmetric-key scheme can be used only to encrypt/decrypt a symmetric key value.

  In the system proposed by Y. Y. Chen et al. (2009), data are divided into blocks and each block is encrypted using a content encryption key CEK. CEK is generated by using the public key of the peer and a random number. Once the peer obtains a license and decryption parameters, he/she generates a content decryption key by using his/her correct private key and the random decryption parameters.

  In the system proposed by Inamura and Iwamura (2014), a content file, which a user obtains from a content provider directly or through P2P, is encrypted with a user key (symmetric) for content distribution. Once the user obtains the content file, he/she undergoes content key issuance protocol with content key issue server, to obtain an encrypted

content key. The user decrypts a content key with his/her user key and decrypts the content with the decrypted content key.

The system proposed by J. S. Li et al. (2010) employs a symmetric P2P key K_E based on AES-128 to encrypt pieces of the music file and a user ID. The RSA cryptosystem is used to generate a digital fingerprint. For the protection of the digital fingerprint, MP generates two public-private key pairs, namely the FP-protection key pair and the PW-protection key pair. These two keys enable the digital fingerprint and password to be conveyed in the network in a ciphered form. With these two keys, the downloading peer can generate his/her own FP-generate key pair. The *MP3stego Tool* (1997) is used for embedding digital fingerprint into the content. This MP3Stego tool uses 3DES encryption technique to protect the fingerprint.

In the system of M. K. Sun et al. (2009), the content stored at the content server and collection server is encrypted with the content key CK. The content key is divided into two parts: CK_1 and CK_2. CK_1 and CK_1 are placed at the license server in an encrypted form. Once the user obtains his/her ticket from the subscription server, he/she can requests the license from the license server. Once the license server receives the request of CK from the user, he/she sends the encrypted CK_1 and CK_2 to the user. The user uses his/her temp private key to decrypt CK_1, and a public key of the license server to obtain CK_2. The user creates CK from CK_1 and CK_2, and then decrypts the content using CK. Also, to provide data protection during communication between subscription server and the user, a session key is used to encrypt the private data.

In the system of Win and Emmanuel (2011), the content owner generates a transaction ID for each anonymous token set to be delivered to the registered users of the system. The owner encrypts the transaction ID with his/her public key, signs the encrypted token and the token expiry time with his/her private key. The owner generates a symmetric key K_j for the encryption of anonymous token set. To use the anonymous token set, the user needs the decryption of K_j. The user requests the decryption of encrypted K_j using the blind decryption protocol. Once the user obtains K_j, he/she can use the anonymous token set to interact with the content providers of the system.

In the system proposed by Tsolis et al. (2011), the watermark that is embedded into the content is encrypted with a watermark key. The watermark key is a positive integer value that plays a vital role in the overall watermarking procedure. It corresponds to the private information that must be shared between the embedder and the detector of the watermark. The encryption of the watermark to be embedded into the content is performed according

to a private key. The encryption is accomplished by using the private key as the seed for the pseudo-random sequence of Gaussian distribution generator.

In the shared fragment method proposed by Stenborg et al. (2011), each packet of the content is duplicated, watermarked, encrypted with different keys, and then distributed to P2P system. Each recipient has a unique set of decryption keys. These keys can only decrypt one of the two copies of every packet. In client-based watermarking method, the content is first encoded, and then scrambled with a secret key W_s. To decrypt the content, an individual key W_i is used at the user end. W_i is the inverse of W_s combined with a small individual transform alteration w_i.

In the system proposed by Megías and Domingo-Ferrer (2014), public-key cryptography is used to encrypt the hash of a fingerprint to be placed at the transaction monitor, so that no single proxy has access to the complete clear-text of the fingerprint hash. Also, the hash of the fingerprint sent from the proxy peers to the transaction monitor contains another encrypted hash (E_1). E_1 is obtained by encrypting the fingerprint hash with the public key of the parent buyer (chosen by the proxy peer to transfer the content to the child buyer). In the content distribution phase, the content transferred from the parent buyer to the child buyer through proxy peers is encrypted with one-time symmetric session keys to restrict intermediate routers to see the clear-text.

The system proposed by Megías (2014) uses public-key cryptography in the distribution and traitor-tracing protocols. The public-key encryption is only applied to fingerprints and hashes of the fingerprints. The binary fingerprint and a hash of the fingerprint are stored in an encrypted form in the transaction monitor. The segments of the binary fingerprint and their hashes are encrypted with the public key of the transaction monitor. The hash of a fingerprint is encrypted to prevent a proxy from accessing the complete clear-text of the fingerprint hash. The binary fingerprint is encrypted to perform a traitor-tracing protocol without involving any buyer and also without decrypting any single fingerprint. Moreover, in the four-party distribution protocol, the content transferred from the parent buyer to the child buyer through a proxy is encrypted with a symmetric session key that is shared between a parent and a child buyer. The symmetric session keys are stored in the transaction monitor which is used as a temporary key database. This protection of symmetric keys prevent malicious proxies from accessing the stored session keys in order to obtain the decrypted fragments of the content.

In the system of Domingo-Ferrer and Megías (2013), the public key of the registration center is used to encrypt the public identity of the buyer to preserve his/her anonymity,

and enable the registration center to identify the re-distributor. Also, the secret input that the buyer receives in a registration protocol with the registration center is encrypted with the public key of the buyer to provide unlinkability.

In the system proposed by Gao et al. (2010), the content is divided by the content provider into two parts: a demo clip which is not encrypted and the encrypted content. The encryption key is generated by adding a high strength watermark signal and lower-frequency DCT coefficients of the content. The merchant generates N decryption keys, and the end users obtain unique decryption keys according to their peer IDs.

In Nemor (Yu et al., 2011), data passing through a communication channel between the content provider and the peer is encrypted with a session key. Also public-key cryptography is used to encrypt the token that contains the information about the content provider and the relay peer.

In P^5 (Sherwood et al., 2002), the packets transferred from the providing peer to the requesting peer through relay peers are encrypted using public-key cryptography.

In the PseudoTrust model (Lu et al., 2007), an anonymous communication model used for transferring the content from the providing (responder) peer to the requesting (initiator) peer encrypts the requested content with the session keys to prevent tail nodes to access the clear-text.

- **Cryptographic Hash Function:** Hash functions play an important role in building security applications related to digital signatures, authentication and data integrity. They are also used to construct pseudo-random number generators.

  SHA-1 is used in the system of Inamura and Iwamura (2014) for the generation of user key seeds. The seeds of the user key are used to provide anonymity to the user.

  In the system proposed by J. S. Li et al. (2010), the MP3Stego tool uses the SHA-1 function to generate pseudo-random bits in fingerprint embedding process. By adopting SHA-1, it is unlikely that obvious and repetitive patterns are apparent to the embedder.

  A one-way collusion resistant hash function is used in the system of Win and Emmanuel (2011) to generate a hash of a transaction ID. The owner also generates a blind decryption key K_j using a hash function. In case of the user privacy revocation mechanism, the owner computes the transaction ID of the hash chain by repeated hashing of the transaction ID found in the anonymous token of the misbehaving user.

  In the system proposed by Megías and Domingo-Ferrer (2014), a hash function is used to generate the hash of the complete fingerprint and the segments of the fingerprint. The

fingerprint hash placed at the transaction monitor is used by the authority in case a buyer intends to cheat the tracing system by showing a different (modified or borrowed) copy of the content. If the hash of a buyer's fingerprint exactly matches the hash of the re-distributed contents fingerprint, then the buyer is charged with unlawful re-distribution. For indexing in the P2P distribution software, a perceptual hash function for which the same hash value is obtained for different (perceptually identical) versions of the same content is used.

A hash function is used in the system proposed by Megías (2014) to generate the hash of the complete fingerprint and the segments of the fingerprint. In case of collusion, the encrypted fingerprint's hash stored at the transaction monitor is used by the authority in the traitor-tracing protocol instead of the fingerprint itself. The hash collisions are almost negligible with a large enough hash space and thus, a traitor could be identified in the majority of the cases. In addition to cryptographic hash functions, the perceptual hash function is used for indexing in the content database of the system.

In the system of Sherwood et al. (2002), the user is mapped to a node and a group by applying a hash function on the public key of each user to form a logical broadcast hierarchy.

The PseudoTrust model (Lu et al., 2007) employs a one-way hash function to bind users' pseudonyms and the authentication paths together. The peers in the PseudoTrust model also use SHA-1 function to generate a message authentication code as a warrant to convince the opposing party that the file is valid and guarantee the integrity of the data.

- **Other Cryptographic Techniques:** Other cryptographic techniques such as zero-knowledge proof of identity, key-agreement protocols (cf. Section 2.2.1.2), and secure multi-party protocols are used to build secure and privacy-preserving applications.

  In a system proposed by Domingo-Ferrer and Megías (2013), a secure multi-party computation (SMC) protocol is used as a building block of anonymous fingerprinting protocol. The SMC protocol enables multiple parties to jointly compute a function based on individually held secret bits of the information, while at the same time keep these secret inputs private in the process. The SMC protocol is used to preserve the privacy of the input and output information (transaction records and the fingerprinted content) in each execution of the fingerprinting protocol.

  A novel authentication scheme based on zero-knowledge proof of identity is designed by Lu et al. (2007) to help unfamiliar peers' successfully complete authentication procedures

TABLE 3.4: Comparison of presented P2P systems based on used privacy techniques

P2P Systems	**Privacy Protection Techniques**		
	Anonymity Techniques	**Trust Management Techniques**	**Cryptographic Techniques**
Megías (2014)	Latent identification and anonymous communications	No	Symmetric encryption, public-key encryption and hash functions
Megías & Domingo-Ferrer (2014)	Latent identification and anonymous communications	No	Symmetric encryption, public-key encryption and hash functions
Inamura & Iwamura (2014)	Anonymous communication	No	Symmetric encryption and SHA-1 hash function
Domingo-Ferrer & Megías (2013)	Anonymous authentication	No	Public-key encryption and secure multi-party computation protocol
Win et al. (2011)	Latent identification and blind decryption	No	RSA cryptosystem and SHA-1 hash function
Tsolis et al. (2011)	No	No	Symmetric encryption
Stenborg et al. (2011)	No	No	Symmetric encryption
Yu et al. (2011)	Anonymous communication	No	Symmetric encryption and public-key encryption
Li et al. (2010)	Latent identification	No	Symmetric encryption, RSA cryptosystem and SHA-1 hash function
Gao et al. (2010)	No	No	Symmetric encryption
Li et al. (2010)	No	No	No
Chen et al. (2009)	No	No	Symmetric block encryption
Sun et al. (2009)	Anonymous authentication	No	Hybrid encryption
Lu et al. (2007)	Full anonymity and anonymous communications	Pseudonym-based trust management	Symmetric encryption, SHA-1 function and zero-knowledge-based authentication

during transactions without revealing any sensitive information.

Table 3.4 presents the privacy techniques used in the compared P2P systems. In the table, a cell is marked with "No" when a privacy technique is not used by the P2P content distribution system.

3.5.3 Guaranteed Security and Privacy Properties

The use of multimedia security, data privacy and anonymity techniques as presented in the previous section allows P2P systems to guarantee the following security and privacy properties:

1. **Content protection:** Content protection incorporates basic security properties such as copyright protection, conditional access and traceability.

2. **Privacy:** The privacy property is categorized into two types: user privacy and data privacy. User privacy implies protection of user-related information and linkability of the users' identities with their online activities. Data privacy implies protection of data against unauthorized entities.

3. **Revocable privacy:** Revocable privacy is a balance between security and privacy needs. It implies that users can enjoy full anonymity unless he/she violates a pre-defined set of rules of the system. This property incorporates both accountability and authentication.

4. **Robustness and security against attacks:** This property is divided into three categories:

   (a) Robustness against signal processing attacks

   (b) Security against collusion attacks

   (c) Security against communication attacks

   The first category is applied to the systems that provide copyright protection by embedding a watermark or fingerprint into the content. The watermark/fingerprint embedded into the content must be resistant against common signal processing attacks such that the extracted information from the attacked content resembles the original embedded information.

   The second category can be sub-divided into two types:

- **Collusion resistance in content protection systems:** The collusion-resistance property in content protection systems that employ fingerprinting techniques implies that the scheme can tolerate a collusion of buyers up to a certain size by preventing colluding buyers from creating a copy that cannot be traced back to one of the colluders. For systems that employ DRM, the collusion-resistance property implies that collusion between any number of malicious peers cannot allow any of them to obtain more than they have and prevent them to make a counterfeit content license.

- **Collusion resistance in privacy protection systems:** This category applies to privacy protection systems that focus on preserving the anonymity of the users against collaborated attacks.

In the third category, the communication channel used for transferring the data between two users must be protected against malicious attacks such as man-in-the-middle attacks, denial of service and replay attacks.

Table 3.5 and Table 3.6 present the privacy and security properties guaranteed in the compared P2P systems. In these tables, a cell is marked with "No" when any of the security and privacy properties is not guaranteed by the P2P content distribution system.

3.6 Conclusions

In this chapter, security and privacy techniques used by the presented P2P content distribution systems are discussed.

First, P2P systems are defined followed by the types of P2P networks with respect to network topologies. A brief review of a few applications of P2P systems is also presented. Secondly, current mechanisms are discussed that are proposed for providing security in P2P content distribution systems. The systems are classified into two types: P2P content distribution systems that provide content protection only, and P2P content distribution systems that provide content protections and traceability. These systems are compared on the basis of the security techniques used to provide content protection (cf. Table 3.3) and the security properties guaranteed (cf. Tables 3.5 and 3.6). Thirdly, privacy-preserving P2P content distribution systems are discussed. Here, again the existing systems are categorized into two types: the systems with copyright protection, data privacy and anonymity, and the systems with a focus on either user privacy or data

TABLE 3.5: Comparison of P2P systems based on guaranteed security and privacy properties

P2P Systems	Content Protection			Privacy		Revocable Privacy
	Copyright Protection	Copy Prevention	Traceability	User	Data	
Megías (2014)	Yes, due to fingerprinting	No	Yes	Yes, due to pseudonymity and anonymous communications	Yes, due to encryption	Yes
Megías & Domingo-Ferrer (2014)	Yes, due to fingerprinting	No	Yes	Yes, due to pseudonymity and anonymous communications	Yes, due to encryption	Yes
Inamura & Iwamura (2014)	No	Yes, due to DRM	No	Yes, due to anonymous communications	Yes, due to encryption	No
Domingo-Ferrer & Megías (2013)	Yes, due to fingerprinting	No	Yes	Yes due to group-signature	Yes, due to encryption	Yes
Yu et al. (2011)	No	No	No	Yes, due to anonymous communications	Yes, due to encryption	No
Win et al. (2011)	No	Yes, due to DRM	Yes	Yes, due to pseudonymity and blind decryption	Yes, due to encryption	Yes
Tsolis et al. (2011)	Yes, due to watermarking	No	No	No	Yes, due to encryption	No
Stenborg et al. (2011)	Yes, due to watermarking	No	No	No	Yes, due to encryption	No
Li et al. (2010)	Yes, due to fingerprinting	Yes, due to DRM	Yes	No	Yes, due to encryption	No
Gao at al. (2010)	Yes, due to fingerprinting	No	Yes	No	Yes, due to encryption	No
Li et al. (2010)	Yes, due to fingerprinting	No	Yes	No	No	No
Chen et al. (2009)	No	Yes, due to DRM	No	No	Yes, due to encryption	No
Sun et al. (2009)	No	Yes, due to DRM	No	Yes, due to anonymous authentication	Yes, due to encryption	No
Lu et al. (2007)	No	No	No	Yes, due to pseudonymity and anonymous communications	Yes, due to encryption	No
Sherwood et al. (2002)	No	No	No	Yes, due to anonymous communications	Yes, due to encryption	No

privacy. The systems are compared based on the privacy techniques used to protect privacy (cf. Table 3.4), and the guaranteed privacy properties (cf. Tables 3.5 and 3.6).

TABLE 3.6: Comparison of P2P systems based on guaranteed security and privacy properties

P2P Systems	**Robustness and Security against Attacks**			
	Signal Processing Attacks	**Collusion and Malicious Attacks**		**Communication Attacks**
		Content Protection Systems	**Privacy Protection Systems**	
Megías (2014)	Yes	Yes	Yes	Yes
Megías & Domingo-Ferrer (2014)	Yes	Yes	Yes	Yes
Inamura and Iwamura (2014)	No	No	No	No
Domingo-Ferrer & Megías (2013)	Yes	Yes	Yes	No
Yu et al. (2011)	No	No	Yes	Yes
Win et al. (2011)	No	No	Yes	No
Tsolis et al. (2011)	No	No	No	No
Stenborg et al. (2011)	Yes	Yes	No	No
Li et al. (2010)	Yes	Yes	No	No
Gao et al. (2010)	Yes	Yes	No	No
Li et al. (2010)	Yes	No	No	No
Chen et al. (2009)	No	Yes	No	Yes
Sun et al. (2009)	No	Yes	No	No
Lu et al. (2007)	No	No	Yes	Yes
Sherwood et al. (2002)	No	No	Yes	Yes

Then, it is shown that the use of content protection and privacy techniques allows P2P content distribution systems to guarantee security and privacy properties. In addition, the presented P2P content distribution systems are compared on the basis of these guaranteed security and privacy properties. The comparison shows that most of the presented systems either focus on content protection or privacy preservation. Except for two systems (Megías & Domingo-Ferrer, 2014; Megías, 2014), all other systems fail to provide the guaranteed security and privacy properties simultaneously. This comparison illustrates that P2P systems face serious challenges in

terms of combining security and privacy properties.

As discussed in Sections 2.2 and 2.3, the implementation of content and privacy protection technologies have some complexities and trade-offs. Thus, the integration of content protection and privacy protection techniques is a challenging task. While designing a secure and privacy-preserving P2P content distribution system, P2P developers and researchers must often face the following challenges:

- Considerable amount of research work has been carried out by researchers to provide an appropriate balance between distributing content on a large-scale and preserving the right of copyright owners. Much of the work has been done by using applications of DRM, watermarking and fingerprinting mechanisms. However, an implementation of a system with the above mentioned techniques face some challenges. For example, most of the proposed works on DRM mechanisms in P2P systems have not been able to effectively prevent copyright infringement and privacy breach of end users at the same time. Similarly, research work for developing robust and secure watermarking schemes is still in progress. The trade-off between robustness, capacity and imperceptibility of watermarking schemes are yet to be achieved. Also, most of the research work involving fingerprinting protocols for copyright protection incur high computational and communicational burdens due to the use of public-key encryption of the contents, secure multi-party protocols, zero-knowledge proofs and other techniques.

- Most of the content protection mechanisms focus on the protection mechanisms for digital contents and pay less attention to the users' privacy rights. Thus, there is a need to design such multimedia techniques that should be sensitive to user's privacy.

- There is a need to develop such security protocols that can help against copyright infringement, protect the privacy of honest users and provide traceability for misbehaving users in the system. Efficient traitor-tracing algorithms must be developed to prevent privacy breach of honest users.

- In order to achieve user privacy in P2P systems, there is always a performance overhead due to encryption and decryption, insertion of fake traffic and an increased routing path to provide anonymity between two communicating users. Thus, a better anonymity and efficiency trade-off is of primary importance for these systems.

- In P2P systems with anonymous authentication, if the privacy of peers is increased, the difficulties of ensuring authenticity and security are increased too. Thus, there is a need to achieve a better trade-off between the authentication and anonymity properties. Also the use of a trusted party for authentication can be risky. Hence, there is a trade-off between accountability and the use of a trusted party for authentication.

It is apparent from the presented open problems that the integration of security and privacy mechanisms in P2P networks is a challenging task, and needs a critical attention of researchers to improve the efficiency of these systems. Research efforts in addressing these concerns are mostly unsuccessful because of the intricacy of each mechanism. Often, exertion for addressing one of these factors may increase the severity of the other, i.e. the strategies with the intention of enhancing security in P2P systems are often characterized with privacy concerns and vice versa. Thus, there is a need to design a P2P content distribution system that can satisfy the needs of both the content providers and the end users. The next chapter presents FPSUM-HE, a framework aimed at assuring copyright and privacy protection to the content providers and the end users, respectively, in a P2P content distribution system.

Chapter 4

Framework for preserving Privacy and Security of User and Merchant based on Homomorphic Encryption

In the previous chapter, the P2P content distribution systems were compared on the basis of guaranteed security and privacy properties, and discussed the design issues faced by the researchers that motivate the proposal of a secure and privacy-preserving content distribution framework. This chapter presents Framework for preserving Privacy and Security of User and Merchant based on Homomorphic Encryption (FPSUM-HE), a P2P content distribution framework for preserving privacy and security of the user and the merchant based on homomorphic encryption. The design goal of FPSUM-HE is to guarantee the content protection, conditional anonymity to the user, privacy of user-related information, resistance against signal processing, collusion and communication attacks.

4.1 Introduction

Recent years have drawn increasing attention from both industry and research communities towards the preservation of content providers' ownership properties, content receivers' privacy and accountability in P2P content distribution systems. The goals of these systems are threefold. A first goal is to provide digital copyright and ownership protection to content owners by

using content protection techniques. A second goal is to preserve user privacy in his/her transactions with the content owner or any other third party by using privacy protection techniques. A third goal is to provide accountability in the system such that the system ensures anonymity for honest users and traceability for misbehaving users. Combining these three goals facilitate content providers, while guaranteeing the privacy of the end users.

However, the major technical challenges that researchers face in designing secure and privacy-preserving P2P content distribution systems are the integration of security and privacy protection techniques that facilitate efficiency in terms of computational and communicational costs, and fulfilment of the guaranteed security and privacy properties (cf. Section 3.5.3). To date, a few P2P distribution systems have been proposed that provide copyright and privacy protection, but at a cost of high computational burden at the merchant's and/or at the buyer's end. Thus, these systems are impractical to distribute multimedia content.

This chapter presents a secure and privacy-preserving P2P content distribution system FPSUM-HE that provides copyright protection to the merchant at a reduced computational cost, and also offers privacy to an end user until he/she is found guilty of illegal re-distribution. In the proposed system, the original multimedia file is partitioned by the merchant into a base and a supplementary file. The base file is much smaller than the original file and contains the most important information. Without this information, the supplementary file is unusable. The base file is dispensed by the merchant on payment from the user, and a supplementary file is sent to the P2P network to be distributed in P2P fashion. This solution reduces the burden of the merchant by only sending the small-sized base file to the buyer and making use of the P2P network infrastructure to support most of the file transfer process. Thus, this scheme enables the merchant to save bandwidth and CPU time. Asymmetric fingerprinting and collusion-resistant codes are used to form a base file in order to provide buyer frameproofness against a malicious merchant and traitor tracing, respectively. The asymmetric fingerprinting protocol is performed by the merchant and the buyer in the presence of a trusted party in such a way that the merchant does not know the fingerprint and the fingerprinted content, while the buyer receives fingerprinted content with his/her unique identity. Collusion-resistant fingerprinting codes are embedded by the merchant into the content so as to identify an illegal re-distributor(s) from an unlawfully re-distributed content. The proposed framework also enables buyers to obtain digital contents anonymously, but this anonymity can be revoked as soon as he/she is found guilty for copyright violation. To ensure anonymous communication between buyers, onion routing is used for an anonymous data transfer. A symmetric-key encryption is performed on the supplementary file to prevent the onion routers (or middle nodes) from observing any similarity between the incoming

and outgoing content. The implementation with a software solution of the proposed system is discussed with a detailed security and performance analysis.

The work described in this chapter has been published as a conference paper (Qureshi, Megías, & Rifà-Pous, 2014), and is accepted in an international journal (Qureshi, Megías, & Rifà-Pous, 2015).

This chapter is organized as follows. Section 4.2 presents an overview of a framework which describes an environment and the design fundamentals of the framework. In Section 4.3, the components of the environment are described. Section 4.4 discusses the design fundamentals of the framework. In Section 4.5, the architecture of FPSUM-HE is presented followed by an explanation of the protocols of the framework designed to address the security and privacy concerns of the merchant and the user, respectively. In Section 4.6, the security analysis of the framework's protocols are discussed through a number of attack scenarios. In this section, the performance and efficiency analysis of the framework are also presented. Finally, a conclusion is provided in Section 4.7.

4.2 Overview of the Framework

The proposed framework consists of two components, namely, environment and design fundamentals, which provide general guidelines on the FPSUM-HE architecture. This section describes an overview of these two main components.

- **Environment:** The environment consists of the following components: P2P network, trust infrastructure and building blocks.
  - The network identifies the P2P network (cf. Section 3.2.1) that is to be used as a platform for running FPSUM-HE.
  - The trust infrastructure identifies the trusted third parties that are used in FPSUM-HE.
  - The building blocks identify the underlying components (cf. Sections 2.2 and 2.4) that are used in the construction of different components of FPSUM-HE.

  The environment is further analysed in Section 4.3.

- **Design Fundamentals:** The design fundamentals aim to provide a clear view on the objectives of FPSUM-HE. It consists of the parties involved, the assumptions, the design requirements and the threat model.
  - The parties involved identify the role of each player in FPSUM-HE.
  - The assumptions identify the general and security assumptions made in the construction of different protocols of FPSUM-HE.
  - The design requirements are the security and privacy requirements that identify precisely what the different protocols of FPSUM-HE should achieve.
  - The threat model establishes a basic attack model that identifies different attacks targeting the involved parties and the protocols of FPSUM-HE.

  The design fundamentals are explained in Section 4.4.

Fig. 4.1 illustrates the environment and design fundamentals of FPSUM-HE.

4.3 Environment

In this section, the first component of FPSUM-HE is described. The main purpose of this component is to provide the design modules that are chosen to construct the protocols of FPSUM-HE.

4.3.1 P2P Network

In FPSUM-HE, a hybrid P2P network is opted as a platform for content distribution. Hybrid P2P systems are presented in Section 3.2.1 in detail. Therefore, this section briefly discusses the reason of the choice of this network, followed by a concise explanation of the network functioning. The main reason for selecting a hybrid P2P is its ability to provide an efficient data search with the help of multiple coordinators, called super peers. The super peers are assigned with responsibilities like maintaining a central index of the files shared by peers, helping a peer in establishing a relationship with another peer for file sharing, etc. Whenever a peer (buyer) connects to the network, he/she directly connects to a single super peer, who gathers information about this buyer and the available content for sharing. When a super peer receives a query from

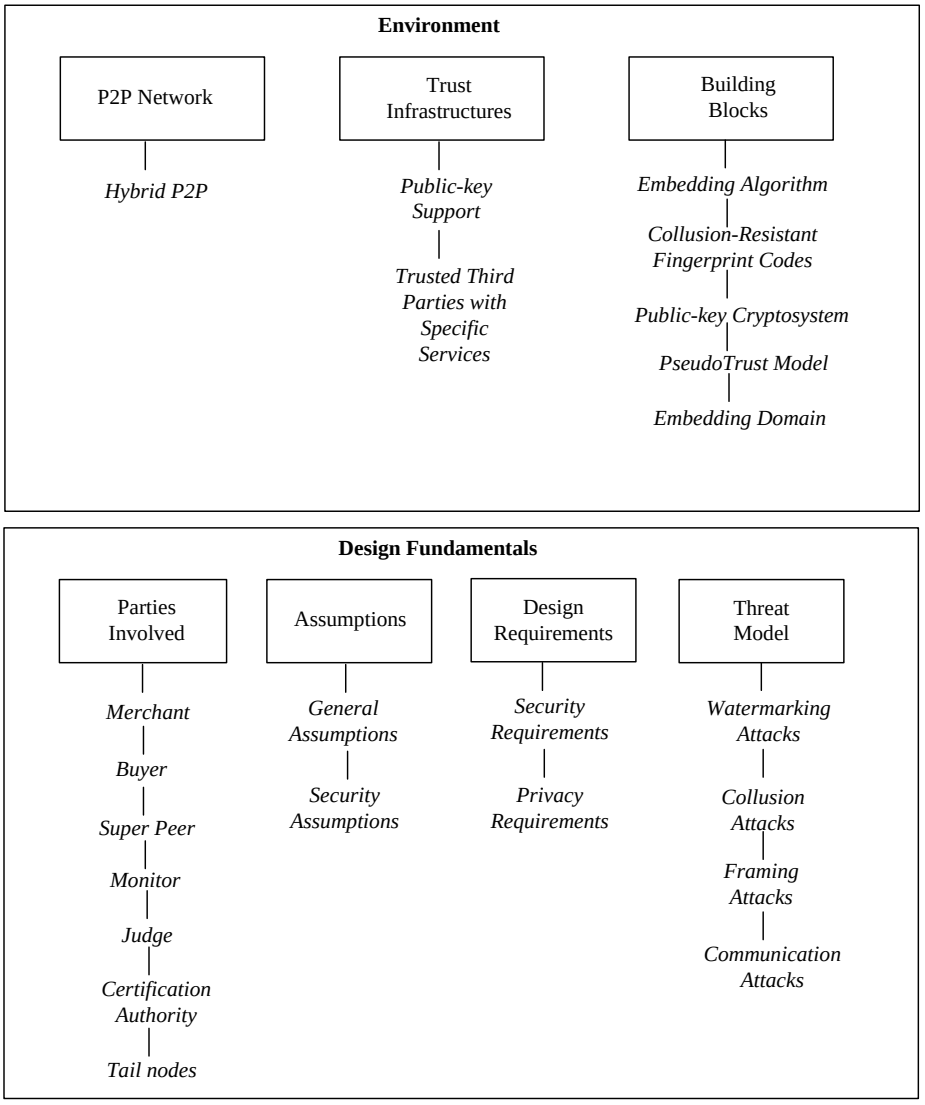


FIGURE 4.1: An overview of FPSUM-HE

a peer within his/her group, he/she first checks his/her local content index. If the query cannot be satisfied, then the super peer forwards the query to other linked super peers. The query response is eventually handed over to the initiating peer.

4.3.2 Trust Infrastructure

Trust infrastructures can be thought of as central points of contact for providing security services. These represent a crucial element of secure and privacy-preserving protocols. For example, in e-commerce applications, there must be a certain trust infrastructure to ensure the authenticity of the service providing website as well as the authenticity of the user. The trust

infrastructure in this case is commonly provided by a public key infrastructure (PKI) supporting the use of a digital certificate, which acts as an electronic equivalent to a witness acknowledging and authenticating the identity of both the parties involved in a transaction. These certificates are generated by a trusted party known as a certificate authority (*CA*).

FPSUM-HE involves the following trust infrastructures:

- **Public key Support:** The security requirements presented in Section 2.2.4.2, notably non-repudiation of re-distribution, requires the generation of non-repudiable proofs (e.g. a digital signature) so that it is possible to prove the guilt of a dishonest buyer. A standard way of providing this proof is by employing a digital signature scheme, which is defined in Section 2.2.1.2. In addition, many existing asymmetric fingerprinting protocols use homomorphic encryption schemes as tools for providing framing resistance. Both digital signatures and homomorphic encryption schemes require the distribution of public verification and encryption keys prior to the content distribution process. These keys must be authenticated so that a party who uses these keys knows that they belong to the legitimate parties. To achieve this, it is assumed the existence of a PKI. Moreover, the anonymity requirement of the secure content distribution protocol, as stated in Section 2.2.4.2, requires that a buyer's real identity remains anonymous to the merchant during the transaction except when he/she is found guilty of illegal re-distribution. An appropriate way of providing revocable anonymity is to use anonymous key pairs validated by a certification authority. For this purpose, one offline external *CA* and one online internal *CA* are assumed to be present in FPSUM-HE. The offline *CA* is only responsible for validating the real identity of a buyer by providing a signed public-key certificate to the buyer. On the other hand, the internal *CA* validates the anonymous key pairs used by the authenticated buyer during the anonymous content distribution protocol.

- **Trusted Third Parties (TTPs) with Specific Services:** In addition to public key support, the presence of other trusted third parties is also necessary to satisfy the security requirements, namely buyer frameproofness, traceability and dispute resolution. (cf. Section 2.2.4.2). The roles of each trusted third party are defined as follows:

  1. **Monitor:** A monitor is a trusted party used to provide framing resistance to a buyer from the merchant in a content distribution protocol. If the monitor is not considered in a content distribution protocol, then the merchant is solely responsible for generation and embedding a user-specific identification mark, known as a fingerprint,

into the content requested by the buyer. However, this creates a customer's right problem. Similarly, if the buyer generates his/her unique fingerprint and sends it securely to the merchant for embedding into the content, it causes a repudiation issue, since a guilty buyer producing unauthorized copies could be able to repudiate the fact and claim that these copies were possibly made by the merchant. In case both the merchant and the buyer generate their own fingerprint, and the jointly computed fingerprint is embedded into the content by the merchant creates a problem of quality degradation or ambiguity attacks. Therefore, the existence of the monitor ensures that the fingerprint embedded into the content is not revealed to either the merchant or the buyer. The monitor is not involved in the embedding operation; it is only used to provide the merchant unique buyer-specific information, and traceability of a buyer involved in an illegal re-distribution. Since it is possible that many buyers request content from the merchant at anytime, the monitor must be always online during the content distribution protocol.

2. **Judge:** The judge is a trusted third party that is not involved in any other protocol of FPSUM-HE, except the identification and dispute resolution protocol. The goal of the identification and dispute resolution protocol is to reveal the real identity of the copyright violator or reject the claim of illegal re-distribution made by the merchant with the help of a certification authority. The presence of a judge in FPSUM-HE ensures that the buyer does not need to participate in the dispute resolution protocol, and the identity of the buyer is not exposed until he/she is found guilty of re-distribution. The judge is only called in case a merchant finds a pirated copy, thus the judge does not need to be online during the content distribution protocol.

4.3.3 Building Blocks

The building blocks are the technical means to fulfill the core security and privacy properties needed by FPSUM-HE. These building blocks are selected from the security and the privacy protection techniques discussed in Sections 2.2 and 2.4. In this section, the role and working of each of these building blocks are detailed.

4.3.3.1 Embedding Algorithm

An embedding algorithm is an important building block in producing a marked copy with user-related information. An embedding algorithm is used to insert a fingerprint into different copies of the same content. Multimedia fingerprinting requires the use of robust data embedding methods that are capable of withstanding attacks that the malicious users might apply to remove the fingerprint. Quantization index modulation (QIM) (Section 2.2.3.3) is a popular watermark embedding technique that provides high watermarking capacity, ease of implementation and blind extraction. However, the basic QIM algorithm uses a fixed quantization step Δ which leads to decreased security, since the buyer can easily observe the even-spaced spikes of the signal due to a constant difference value Δ, and identify the embedding positions. For improving the QIM algorithm, dither modulation (DM) is produced based on the basic QIM. In DM quantization, the host signal is dithered using the watermark information. The dither is a pseudo-random signal that serves as the key to provide security to the scheme. Then, the watermark information is embedded by quantizing the dithered host signal using quantizers selected from a set of possibilities. DM quantization has a convenient performance in terms of imperceptibility, data payload, robustness and blind extraction.

A basic dither modulation technique, called subtractive-dither QIM (SD-QIM) scheme (Prins et al., 2007), is used in the proposed system for embedding a collusion-resistant fingerprint code into the multimedia content. In SD-QIM, a small amount of dither d_j is added prior to quantizing the signal amplitude x_i to an odd or even value depending on the information bit $f_{i,j}$. After the quantization of $x_i + d_j$, the same amount of dither d_j is subtracted. The dither is used in cooperation with the QIM uniform quantizers $Q_{\Delta-odd}(\bullet)$ and $Q_{\Delta-even}(\bullet)$, which use a quantization step size of 2Δ. The output of the SD-QIM operation obtained as the following:

$$y_i = Q_{2\Delta}(x_i + d_j) - d_j.$$

A suitable choice for the PDF of the random dither d_j is a uniform distribution on $[-\Delta, \Delta]$. Fig. 4.2 illustrates the working of SD-QIM watermarking scheme.

4.3.3.2 Embedding Domain

In the scientific literature, a large number of watermarking methods and algorithms are found. These algorithms can be divided into two large groups of algorithms based on embedding a

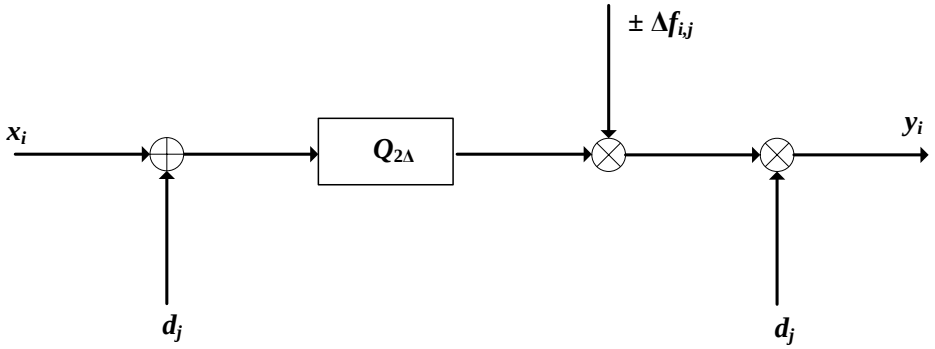


FIGURE 4.2: Subtractive-dither QIM

watermark directly into the spatial (or time) or the frequency (transform) domain (cf. Section 2.2.3.3). Spatial-domain watermarking techniques are often not robust to signal processing attacks, although they are efficient in computing. Compared to spatial-domain watermarking, watermarking in the frequency domain is more robust. Amongst the transforms used for the purpose of copyright protection, the wavelet transform is a popular embedding domain method. Wavelets are a mathematical tool for hierarchically decomposing signals. They can be applied to extract information from the signal in different resolution levels. They allow a function to be described in terms of a coarse overall shape plus a family of details, which correspond to the complementary information necessary to recover the original data from one level to the other, until the finest resolution level is achieved. Because of their inherent multi-resolution nature, wavelet schemes are suitable for applications where scalability and tolerable degradation are important.

In FPSUM-HE, the discrete wavelet transform (DWT) is used to embed the fingerprint into the multimedia content. The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into approximation and detail coefficients. The DWT employs two sets of functions, called scaling functions and wavelet functions, which are associated with low-pass and high-pass filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive high-pass and low-pass filtering of the spatial (or time) domain signal. For one-level DWT decomposition, the original signal passes through a half-band high-pass filter and a low-pass filter. After the filtering, half of the samples are eliminated and the signal can therefore be sub-sampled by 2, simply by discarding every other sample. This decomposition halves the time resolution since only half the number of samples now characterizes the entire signal. However, this operation doubles the frequency resolution. The above procedure can be repeated for further levels of decomposition. At every level, the filtering and sub-sampling results in half the number of samples and half the frequency

band spanned (and hence double the frequency resolution).

4.3.3.3 Collusion-resistant Fingerprinting Codes

The security requirement presented in Section 2.2.4.2, namely collusion resistance, requires the fingerprinting scheme to be able to defy collusion attacks. Collusion resistance can be provided in the scheme by employing collusion-resistant fingerprint codes, which are defined in Section 2.2.4.3. Many collusion-secure (c-secure) codes are proposed in the literature and, amongst these, Tardos (2003) codes is a much studied collusion-resistant fingerprinting code. Tardos codes were the first to achieve the asymptotically optimal property $m \propto c_0^2$, where m is the length of the code and c_0 is the number of colluders that can be resisted, since the previous collusion-resistant codes had higher powers of c_0 or required an alphabet size that is unrealistically large in the context of multimedia watermarking. This optimality has generated much interest and many researchers have proposed new c-secure codes to further improve the code length m or provide better traceability through encoder or decoder modifications.

In FPSUM-HE, a variation of Tardos codes, i.e. Nuida et al. (2007) c_0-secure codes, are used for generating collusion-resistant codes. These codes are based on a relaxed marking assumption called δ-marking assumption, i.e. the number of undetectable bits that are either erased or flipped is bounded by a δ-fraction of the total code length m. The length of Nuida's δ-robust c_0-secure codes is approximately 5.35% shorter than Tardos codes, which is the smallest value so far provable without requiring any additional assumption. Nuida et al.'s codes are based on a discrete bias distribution that depends on c_0. In the tracing algorithm of Tardos codes, a score is assigned to each user which measures how his/her codeword is similar to the pirated codeword, and then all users whose score exceeds a suitably determined threshold are output as pirates. However, if the score of a pirate is much higher than the threshold and that of an innocent user is only slightly higher than the threshold, then the latter user is also accused, though the most suspect user is obviously the former. Nuida et al.'s improved the Tardos' code tracing algorithm by outputting just one user with the highest score. The number of users N, the error probability ε and the coalition size c_0 are inputs of the Nuida et al.'s fingerprint generation algorithm, and the fingerprinting code F and a secret bias vector p are the outputs of this algorithm. In the tracing algorithm, a pirated codeword pc, a bias vector p and the original fingerprint are the inputs, and the output is a pirate with the highest score. The details of the Nuida et al.'s fingerprint codes generation and the traitor-tracing algorithm can be found in Sections 4.5.1.1 and 4.5.1.5.

4.3.3.4 Homomorphic Encryption

For some asymmetric fingerprinting protocols, homomorphic encryption schemes together with digital watermarking schemes are used to provide watermarking in the encrypted domain (cf. Section 2.2.4.4). This is used to provide buyer the frameproofness and non-repudiation security properties (Section 2.2.4.2).

A homomorphic cryptosystem (or privacy homomorphism) refers to a cryptosystem $\mathcal{E}$ which is homomorphic with respect to some binary operators $\odot_{\mathcal{M}}$ in the plaintext space $\mathcal{M}$ and $\odot_{\mathcal{C}}$ in the ciphertext space $\mathcal{C}$, such that $\forall m_1, m_2 \in \mathcal{M} : \mathcal{E}(m_1 \odot_{\mathcal{M}} m_2) = \mathcal{E}(m_1) \odot_{\mathcal{C}} \mathcal{E}(m_2)$. Homomorphic cryptosystems can be classified as two groups, namely the ones whose security relies on the "decisional composite residuosity assumption" (DCRA), and the ones of the ElGamal class based on the "decisional Diffie-Hellman assumption" (DDHA). The indistinguishability under chosen plain-text attacks (Goldwasser & Micali, 1984) guarantees that an adversary does not obtain any knowledge about the plain-text m_1 from the cipher-text $\hat{c}$. For instance, the deterministic RSA cryptosystem (Rivest et al., 1978) and the ElGamal (El-Gamal, 1985) are multiplicative privacy homomorphisms. The Goldwasser-Micali cryptosystem (Goldwasser & Micali, 1984), and the Paillier cryptosystem (Paillier, 1999) are additive privacy homomorphisms.

In FPSUM-HE, a Paillier cryptosystem (Paillier, 1999) is employed, which is homomorphic with respect to the addition operation, i.e. there exists a map between an addition in the plain-text domain m_1 and an operation in the cipher-text domain $\hat{c}$. Paillier is a probabilistic asymmetric algorithm for public-key cryptography and inherits additive homomorphic properties. It is a semantically secure cryptosystem based on the problem of deciding whether a number is an $\mathcal{N}$-th residue modulo $\mathcal{N}^2$, whose computation is believed to be computationally difficult, and is linked to the hardness of factorization $\mathcal{N}$, if $\mathcal{N}$ is a product of two large prime numbers. An $\mathcal{N}$-th residue is defined below followed by an explanation of its data encryption operation.

Given the product of two large primes $\mathcal{N} = \mathcal{PQ}$, the set $\mathbb{Z}_{\mathcal{N}}$ of the integer numbers modulo $\mathcal{N}$, and the set $\mathbb{Z}^*_{\mathcal{N}}$ representing the integer numbers belonging to $\mathbb{Z}_{\mathcal{N}}$ that are relatively prime with $\mathcal{N}$, $z_1 \in \mathbb{Z}^*_{\mathcal{N}^2}$ is said to be an $\mathcal{N}$-th residue modulo $\mathcal{N}^2$ if there exists a number $z_2 \in \mathbb{Z}^*_{\mathcal{N}^2}$ such that $z_1 = z_2^{\mathcal{N}} \bmod \mathcal{N}^2$.

The set-up, encryption and decryption procedures are explained as following:

- **Setup:** Select $\mathcal{P}$, $\mathcal{Q}$ large primes. The private key is the least common multiple of ($\mathcal{P} - 1$,

$\mathcal{Q}-1$), denoted as $\lambda = \text{lcm}(\mathcal{P}-1, \mathcal{Q}-1)$. Let $\mathcal{N} = \mathcal{P}\mathcal{Q}$ and $\mathcal{G} \in \mathbb{Z}^*_{\mathcal{N}^2}$ an element of order $\beta \cdot \mathcal{N}$ for some $\beta \neq 0$ ($\mathcal{G} = \mathcal{N}+1$ is usually a convenient choice). $(\mathcal{N}, \mathcal{G})$ is the public key.

- **Encryption:** Let $m_1 < \mathcal{N}$ be the plain-text and $\kappa < \mathcal{N}$ a random value. The encryption $\hat{c}$ of m_1 is:

$$\hat{c} = \mathcal{E}(m_1, \kappa) = \mathcal{G}^{m_1}\kappa^{\mathcal{N}} \bmod \mathcal{N}^2.$$

- **Decryption:** Let $\hat{c} < \mathcal{N}^2$ be the cipher-text. The plain-text m_1 hidden in $\hat{c}$ is:

$$m_1 = \mathcal{D}(\hat{c}) = \frac{L(\hat{c}^{\lambda} \bmod \mathcal{N}^2)}{L(\mathcal{G}^{\lambda} \bmod \mathcal{N}^2)} \bmod \mathcal{N},$$

where, $L(\nu) = (\nu-1)/\mathcal{N}$.

From the above equations, it can be easily verified that the Paillier cryptosystem is additively homomorphic, since: $\mathcal{E}(m_1, \kappa_1) \cdot \mathcal{E}(m_2, \kappa_2) = \mathcal{G}^{m_1+m_2}(\kappa_1\kappa_2)^N = \mathcal{E}(m_1+m_2, \kappa_1\kappa_2)$ and $\mathcal{E}(m_1, \kappa)^{\zeta} = (\mathcal{G}^{m_1}(\kappa)^{\mathcal{N}})^{\zeta} = (\mathcal{G}^{\zeta m_1}(\kappa)^{\zeta\mathcal{N}}) = \mathcal{E}(\zeta m_1, \kappa^{\zeta})$.

4.3.3.5 PseudoTrust Model

In order to provide the revocable anonymity and unlinkability properties (cf. Section 2.2.4.2) in a P2P content distribution system for privacy protection of buyers (peers), the PseudoTrust model (cf. Section 3.4.3) based on a zero-knowledge proof-of-identity (cf. Section 2.4.3.2) proposed by Lu et al. (2007) is employed. In the PseudoTrust model, the peers authenticate each other with their pseudo-identities (cf. Section 2.4.1.1) without a presence of a trusted third party (certification authority). It enables pseudonym-based trust management so that the real identities of peers are protected during the authentication. The PseudoTrust model anonymizes the communication between two peers by adopting onion routing (cf. Section 2.4.1.2) within the model. In the authentication protocol, the Diffie-Hellman key exchange protocol (cf. Section 2.2.1.2) is incorporated to provide confidentiality, unlinkability and integrity to data exchanges such that, after authentication, both peers can share a session key for encrypting the exchanged data.

Each peer is required to generate a pseudo-identity (*PI*), and a pseudo-identity certificate (*PIC*) using the SHA-1 function (cf. Section 2.4.3.1). A *PI* is used to identify and replace the real identity of a peer in a P2P system. A *PIC* is generated to authenticate the *PI* holder. Since the PseudoTrust model allows peers to generate their pseudo-identities individually and peers do

not depend on any trusted third party to authenticate with each other, it creates an accountability problem. Without a trusted third party, it would be impossible to find a person responsible for mischievous activities. Thus, to add accountability to the system, an internal certificate authority (CA_R) is incorporated in the PseudoTrust model. Each peer is authenticated by CA_R before he/she joins the network. Thus, each peer has a private key, a public key and a public-key certificate signed by CA_R. The pseudo-identities and certificates are used by the peers for anonymous communication within the P2P system. The detail of the anonymous communication process used in the PseudoTrust model is discussed in Section 3.5.2.1. The details of the generation of PI are presented below.

CA_R selects a finite cyclic group $\mathbb{G}$ with P elements, and g as a generator of $\mathbb{G}$. The parameters g and P are made public by CA_R. CA_R then selects a secret random number $r \in [1, \ldots, P-1)$ and sends r encrypted with the peer's public key to the peer. Thus, CA_R and all the peers share a secret number r. When a new peer joins the network or an old peer leaves the network, the secret number r should be updated by CA_R.

A. PI **Generation:**

1. ID_{P_a} denotes the real identity of a peer P_a.
2. $\text{Cert}_{CA_R}(P_a)$ denotes the public-key certificate of peer P_a.
3. P_a chooses a number $\upsilon_1 \in \{0, 1, \ldots, P-1\}$.
4. P_a uses his/her private key K_{sP_a} to sign $\{ID_{P_a}, \text{Cert}_{CA_R}(P_a), r, \upsilon_1\}$.
5. P_a computes a PI using a hash function. PI_{P_a} is defined as following:

$$PI_{P_a} = h(ID_{P_a}, \text{Cert}_{CA_R}(P_a), r, \upsilon_1, \text{Sign}_{P_a}\{ID_{P_a}, \text{Cert}_{CA_R}(P_a), r, \upsilon_1\}).$$

Without CA_R in FPSUM-HE, the peers can use their self-generated pseudo-identities, but then there would be no way of tracing a malicious peer, since each peer could use multiple pseudo-identities and can even impersonate other peers. Thus, there is always a trade-off between anonymity and accountability. Increased anonymity can cause problem in the identification of a copyright violator, which in turn could be a problem for the content provider. Thus, to ensure accountability and revocable anonymity, the presence of CA_R is worth it. Moreover, in FPSUM-HE, the authentication between two peers does not involve CA_R. The role of CA_R is limited

to one-time generation of public-certificates and secret number r for generation of a pseudo-identity.

4.4 Design Fundamentals

The design fundamentals aim to provide a proper definition of the objectives of FPSUM-HE.

4.4.1 Parties Involved

FPSUM-HE involves six entities, and the function of each entity is defined as follows:

- A merchant M is an entity that distributes the copyrighted content to the buyers (peers) in the P2P system. It is involved in the generation and distribution of the base and supplementary files, traitor tracing and dispute resolution protocols.
- A buyer (peer) B_i is an entity that can either play a role of a data requester or provider. A buyer is involved in the acquisition of a base file from the merchant, obtaining and distributing a supplementary file in the P2P system and a dispute resolution if he/she is found guilty of copyright violation.
- A super peer SP (a.k.a. index server) is a reputed peer with additional facilities who is assigned the role of the coordinator for a small portion of the group of peers. Each SP maintains a list of the peers connected to the network and acts as a central coordinator. However, SP store peers' pseudo-identities instead of their real identities or IP addresses. The peers send their queries to SP for downloading their files of interest. Initially, SPs are provided with the supplementary file from M at the system start-up. SP divides the supplementary file into multiple fragments, and on a request from a buyer for the content, he/she transmits these fragments to the requesting buyer.
- A certification authority CA_R is a trusted party that is responsible for issuing certificates to the buyer for acquisition of the base file from M, and the supplementary file from the peers. The certificate is used to guarantee that the pseudo-identity of a buyer is correctly registered to CA_R, and only CA_R knows about the real identity of the buyer.
- A monitor MO functions as a trusted third party which is responsible for the generation of collusion-resistant fingerprint codes. The existence of MO ensures that the generated

fingerprints are not revealed to M and the buyer. It also keeps the record of transactions between M and the buyer. MO is also responsible for executing the traitor-tracing algorithm in case of a piracy claim by M. In case of dispute resolution between M, a buyer and judge, MO provides the pseudonym of the accused buyer to the judge.

- A judge J is assumed to be a trusted party which resolves disputes between M and a peer with the cooperation of MO and CA_R.

4.4.2 Assumptions

In this subsection, the general and security assumptions of FPSUM-HE are described.

4.4.2.1 General Assumptions

In the following, the general assumptions related to the construction of FPSUM-HE are defined.

- There are six major players involved: merchant M, buyer (peer B_i), super peer SP, monitor MO, certification authority CA_R, and judge J.
- Each entity is supposed to have a public key K_p and a private key K_s.
- At the start-up of FPSUM-HE, the bootstrapping is carried out via a well-known booting peer.
- The real identity of each entity is validated by an external (offline) certification authority CA_{ext}. Thus, each entity has a public key certificate signed by CA_{ext}. CA_{ext} keeps track of all the identities to be sure that they remain unique, and also to revoke an identity of a malicious entity. The generation of a public key certificate is a one-time process.
- Before joining the system, each buyer is authenticated by an internal certification authority CA_R of the system. CA_R validates the identity of a buyer from CA_{ext}. After successful verification, each buyer has a private key and a public key certified by CA_R. CA_R generates a random number r and shares it with an authenticated buyer for the generation of a pseudo-identity.
- Each peer can have multiple pseudo-identities.

- On joining the P2P system, each peer finds its tail node and builts anonymous links with each other via onion routing. A tail node T_A is a message transferring agent that manages anonymous communication on behalf of a peer P_a. Each peer within the P2P network has one such agent. The tail node forwards the query of a requesting peer to the providing peer through an anonymous path and returns the reply back to the requesting peer.

- The reconstruction of the original file from the base and supplementary files should be performed at the buyer's end. The base file cannot be shared within the end users of the system.

4.4.2.2 Security Assumptions

The security assumptions of FPSUM-HE are defined in this section.

- *M* and the buyer do not trust each other but they both trust *MO*. Because of the anonymity of the embedding procedure, *MO* generates the collusion-secure fingerprints as this is the only party that is trusted by both *M* and the buyer to generate a valid fingerprint.

- The SHA-1 function used in the system to generate unforgeable and verifiable pseudo-identities for each entity is secure and cannot be reversed.

- The communication between the peers is anonymous due to the use of onion routing within the system.

- *SP* is selected on the basis of his/her reputation and resources. *SP*s that manage the content distribution activities honestly gain more reputation among peers and the merchants. More peers shall connect with a well-reputed peer and obtain the intended data through that trusted *SP*.

4.4.3 Design Requirements

For FPSUM-HE, the following design requirements are defined in terms of content protection (security) and privacy protection.

4.4.3.1 Security Requirements

- M should be able to trace and identify an illegal re-distributor in case of finding a pirated copy with the help of MO, J and CA_R.
- The scheme should be collusion resistant against a given coalition size c_0 as specified by Nuida et al. (2007).
- M should not be able to frame an honest buyer of illegal re-distribution.
- The buyer accused of re-distributing an unauthorized copy should not be able to claim that the copy was created by M.
- The embedding process should be blind and the embedded fingerprint should be imperceptible and robust against common signal processing attacks.
- The data expands on conversion from a plain-text to an encrypted representation of signals due to the use of an additive homomorphic cryptosystem. The homomorphic encryption should be performed in such a way that the size of the encrypted base file remains small.

4.4.3.2 Privacy Requirements

- The identity of a buyer should remain anonymous during transactions until he/she is proven to be guilty of copyright violation.
- The identity of a buyer should not be linked to his/her activities such as purchasing, transferring of file and so on.
- The real identity of a buyer should be protected during the authentication process, thus enabling each buyer to verify the authenticity of each other anonymously.
- None of the tail nodes should know about the requesting buyer's and source provider buyer's identity or an item being exchanged. Thus, the supplementary file transfer between the requesting buyer and the providing buyer must be encrypted to prevent linkability of the content.
- J, with the help of MO, should be able to resolve the disputes without involving the buyer in the process.

4.4.4 Threat Model

This section highlights an attack model for FPSUM-HE related to the robustness of a fingerprint, resistance of a fingerprint against collusion attacks, buyer's security from malicious entities and communication attacks.

4.4.4.1 Watermarking Attacks

A fingerprint embedding scheme used for copyright protection must have a capability to survive attacks such as signal enhancement, geometrical operations and noise filtering. The inserted fingerprint must be highly robust against these attacks such that the retrieved fingerprint unambiguously identifies the copyright owner. The robustness of a fingerprint can be evaluated by simultaneously considering fingerprint impairment and the distortion of the attacked content. An attack succeeds in defeating a fingerprint embedding scheme if it impairs the fingerprint beyond acceptable limits while maintaining the perceptual quality of the attacked data. Thus, an effective attack handling is required during evaluation of embedding techniques. The attacks on fingerprint embedding schemes are categorized into two groups: attacks on audio and video fingerprints.

1. ***Attacks on an Audio Fingerprint***

   (a) ***Re-quantization:*** The fingerprinted audio signal is re-quantized from original bit-rate down to half the bit-rate and then back to original number of bits/sample. An increased incoherent background noise is heard in the audio track due to the rounding errors produced by the re-quantization process.

   (b) ***Re-sampling:*** Under this attack, fingerprinted audio signals are down-sampled and then up-sampled (or vice versa) back to its original sampling rate. This attack affects audibility and produces distortions especially in audio tracks carrying high frequencies.

   (c) ***MP***3 ***Compression:*** MP3 compresses data by discarding some parts of it. The fingerprinted audio signal can be compressed at different bit rates (e.g. 256, 128, 64, or 32 kbps) and then decompressed back to the wave format. This attack reduces the file size but at the cost of a lower sound quality. The lower the bit-rate, the lower is the sound quality.

(d) ***Additive White Gaussian Noise:*** The Additive White Gaussian Noise (AWGN) attack adds an additive Gaussian noise of zero mean, constant variance, and controlled value of signal-to-noise ratio (SNR) to the fingerprinted signal. The SNR is a metric that determines the strength of this attack. An addition of noise to a signal results in quality degradation of that signal.

1. ***Attacks on a Video Watermark***

   (a) ***Median Filtering:*** Under this attack, a window of $[N' \times N']$ pixels is moved onto a fingerprinted signal. It returns the median pixel value in the moving window. The lower the value of N', the smoother image is produced. On the other hand, an increase in N''s value considerably blurs the image.

   (b) ***Re-sizing:*** In re-sizing, a fingerprinted signal is either re-sized to double or down-scaled to half the size of its original size, and it is then reduced back to its original size. However, in downscaling an image to the desired size, there is a loss of information.

   (c) ***H.***264 ***Compression:*** H.264 compression is one of the common lossy compression attacks on a video content. With H.264 compression, there is a trade-off between video quality, processing cost of compression/decompression, and file size. This trade-off is determined by specifying a bit rate.

   (d) ***AWGN:*** Gaussian noise insertion is a signal processing attack in which the amount of noise to be added into a signal is controlled by its mean, variance and SNR value.

4.4.4.2 Collusion Attacks

Collusion attacks are a challenging issue for digital fingerprinting. The main concern for a fingerprinting system is the resistance of a fingerprint to colluders' attacks. Collusion occurs when different buyers recombine their marked copies to obtain a new copy of the content such that they cannot be accused of copyright violation. The collusion attacks are defined as follows:

1. ***Averaging Attack:*** In an averaging attack, attackers with a total of K fingerprinted copies of the same content collude to produce a colluded version Y'. The fingerprinted signals are typically averaged with an equal weight for each user. It can be defined mathematically

as following:

$$Y'(i) = \frac{y_0^{'}(i) + y_1^{'}(i) + \ldots y_{K-1}^{'}(i)}{K}.$$

2. ***Minimum Attack:*** Under this attack, the attackers create a copy Y' whose i^{th} ($i = 1, \ldots, m$, where m is the length of a fingerprint), component is the minimum of the i^{th} components of the observed marked copies. Mathematically, it is defined as follows:

$$Y'(i) = \min(y_0^{'}(i), y_1^{'}(i), \ldots, y_{K-1}^{'}(i)).$$

3. ***Maximum Attack:*** The colluders create an attacked copy Y' by considering the maximum value of the i^{th} components of their individual marked copies. It can be defined mathematically as follows:

$$Y'(i) = \max(y_0^{'}(i), y_1^{'}(i), \ldots, y_{K-1}^{'}(i)).$$

4. ***Median Attack:*** In the median attack, the attackers take the median of the values of the corresponding components of the individual marked copies to create a pirated copy Y'. Mathematically, it is defined as the following:

$$Y'(i) = \text{median}(y_0^{'}(i), y_1^{'}(i), \ldots, y_{K-1}^{'}(i)).$$

4.4.4.3 Framing Attacks

Framing attacks are the type of attacks that are aimed to de-anonymize a buyer and accuse an innocent buyer of illegal re-distribution of the purchased content. The framing attacks are defined as follows:

1. When the fingerprint is inserted solely by M, M may benefit from framing attacks on an innocent buyer. This attack is successful if M is able to prove to the judge J that illegal copies of the marked content belong to a particular buyer even though a buyer has not bought this content, or had bought this content but did not re-distribute copies of it illegally.

2. Different transactions carried out by a buyer with the same pseudo-identity are linkable to one another and an attacker could infer some private information of a buyer through data mining techniques.

3. A malicious entity may try to find two different but real identities such that the two identities have the same pseudo-identity. It might then use one of the two identities to impersonate the buyer with the other identity.

4.4.4.4 Communication Attacks

The following strategies allow attackers to exploit the communication between two buyers interacting in a P2P fashion:

1. ***Replay Attack:*** This approach allows attackers to exploit the authentication process of FPSUM-HE. Under this attack, the attacker may eavesdrop and collect some previous proofs of an initiator buyer P_a, and at a later time reuses this information in an attempt to falsely authenticate to the responder buyer P_b.

2. ***Leakage of the secret number r:*** The secret number r is a key generated by CA_R for sharing it with an authenticated buyer in the pseudo-identity generation step. However, if r is leaked then any malicious node can use it to impersonate other buyers.

3. ***Man-in-the-Middle Attack (MIMA):*** During a communication between an initiator buyer P_a and a responder buyer P_b, an eavesdropper $\hat{E}$ may access and modify messages between these buyers without either buyer knowing that the link between them has been exposed.

The security of the system against these attacks is discussed in Section 4.6.

4.5 Model

This section describes the architecture of FPSUM-HE. Fig. 4.3 shows the structure of FPSUM-HE that contains six main entities: merchant, buyer, super peer, monitor, judge and certification authority. These entities are involved in six key protocols: fingerprint generation, file partitioning into BF and SF, distribution of BF and SF, traitor tracing and dispute resolution of the system.

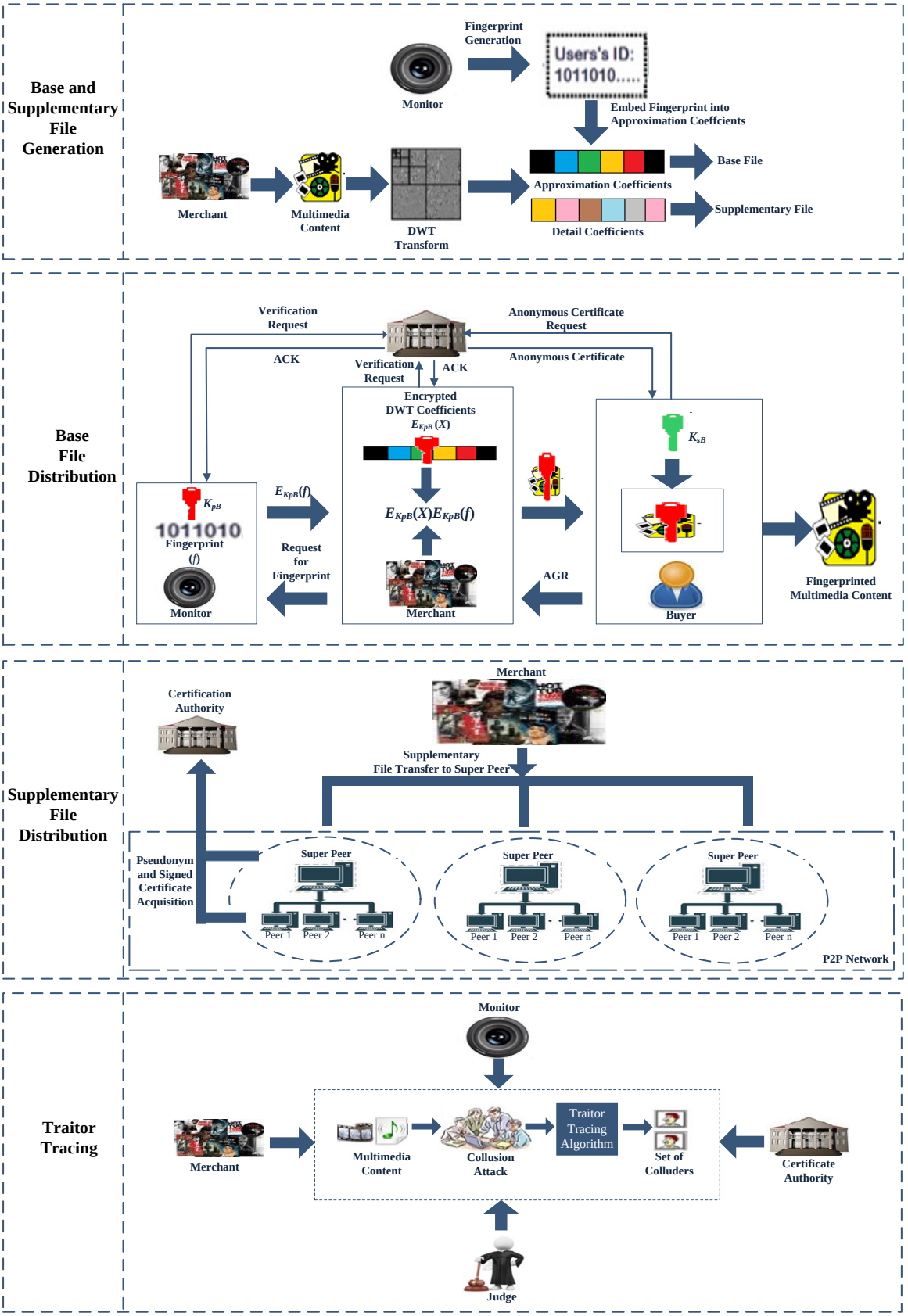


FIGURE 4.3: An overview of FPSUM-HE

4.5.1 Protocols

The protocols of the proposed framework are reviewed in the following sections:

4.5.1.1 Generation of a Collusion-resistant Fingerprint

The fingerprint f_i is generated by *MO* using the Nuida et al. (2007) codes algorithm. The fingerprint generation algorithm takes ε, N and c_0 as inputs, and outputs a collection $F = (f_1, \ldots, f_N)$ of binary codewords (f_i) of size m and a secret bias vector p, as shown in Algorithm 1. The codeword f_i is meant to be embedded into the content of a buyer.

Algorithm 1 Fingerprint Generation

procedure NUIDA ET AL.'S CODES

Input parameters: c_0, N $(N \geq c_0)$, ε

Output parameters: F, p

begin

$m \leftarrow (c_0^2 K \log(N/\varepsilon))$ ▷ where the value of K is 4.245

Select p independently by picking uniformly at random for all $1 \leq j \leq m$

for all $1 \leq i \leq N$ **do** ▷ a loop over all users

for all $1 \leq j \leq m$ **do** ▷ a loop over the bits of the codeword of a user

$P(u_{i,j} = 1) \leftarrow p_j$

$P(u_{i,j} = 0) \leftarrow 1 - p_j$ ▷ with probability 1/2 each to obtain $W_{N \times m}$

end for

end for

return F, p ▷ Fingerprint $F = (f_{i,j})$ where $i \in [1, \ldots, N]$, $j \in [1, \ldots, m]$ and secret vector (p)

end procedure

4.5.1.2 File Partitioning

This section discusses the partitioning of a multimedia file X into a small-sized BF and a large-sized SF. The proposed method employs the DWT to split a multimedia content into low-frequency (approximation coefficients) and high-frequency (detail coefficients) components. An approximation coefficient is then itself split into a second-level approximation and detail coefficients, and the process is repeated as many times as desired (levels of decomposition). The

approximation coefficients are used to form *BF* and detail coefficients are used in *SF* creation. *BF* contains a collusion-resistant fingerprint f_i and is dispensed by the merchant on a payment from the buyer, and *SF* is sent to the P2P network to be distributed in a P2P fashion. The asymmetric fingerprinting protocol is performed between M and a buyer in the presence of *MO* in such a way that M does not know a fingerprint and the fingerprinted *BF*, while the buyer receives *BF* with his/her unique pseudo-identity.

MO generates Nuida et al. (2007) c_0-secure codes, and encrypts each bit of the fingerprint f_i with the public key of a buyer B_i using the Paillier cryptosystem to obtain the following equation:

$$\mathcal{E}_{K_{pB_i}}(f) = (\mathcal{E}_{K_{pB_i}}(f_{i,1})|\mathcal{E}_{K_{pB_i}}(f_{i,2}), | \ldots, |\mathcal{E}_{K_{pB_i}}(f_{i,m})).$$

where m is the length of the fingerprint f_i, i is the user, and "|" is the concatenation operator. *MO* sends the encrypted fingerprint $\mathcal{E}_{K_{pB_i}}(f_i)$ to M. In order to embed an encrypted fingerprint $\mathcal{E}_{K_{pB_i}}(f_i)$ in the approximation coefficients for the formation of *BF*, the additive homomorphic property of public-key cryptosystems is applied. However, additive homomorphic cryptosystems cannot work on real-valued DWT coefficients. Hence, M quantizes the approximation coefficients of the multimedia content that the buyer wishes to obtain, using a quantizer with coarseness 2Δ. Therefore, the approximation coefficients are quantized to integer values. The quantizer step size Δ is a positive integer to ensure that the quantized values can be encrypted. Before quantization, M selects the fingerprint embedding positions by using a unique secret key *sk* which is also used to extract f_i from the re-distributed copies. Based on the embedding algorithm proposed by Prins et al. (2007), a dither vector d_j is generated by M, and it is then added to these selected approximation coefficients a_j (where $j = 1, \ldots, m$). Then, M quantizes these values with 2Δ resulting in $Q_{2\Delta}(a_j + d_j)$. The same dither vector d_j is subtracted from the quantized values resulting in $Q_{2\Delta}(a_j + d_j) - d_j$.

If the values of a_j are sufficiently large, then using integer-valued coefficients is not a restriction at all. For smaller values of a_j, however, using integer values may be too restrictive or may yield too large deviations between an encryption and decryption results. To circumvent this problem, all the quantized approximation coefficients are scaled by a constant factor c' before encryption. c' has to be communicated to the buyer so that the buyer can re-scale *BF* after decryption to the original signal. M then encrypts all the quantized and scaled approximation coefficients with the public key of the buyer. In order to embed a single bit of information $f_{i,j}$ into one of the scaled, quantized and encrypted value approximation coefficients at a particular

embedding position, M performs the following operation:

$$\mathcal{E}_{K_{pB_i}}(y_i) = \begin{cases} \mathcal{E}_{K_{pB_i}}(c' \cdot (Q_{2\Delta}(a_j + d_j) - d_j)) \times \mathcal{E}_{K_{pB_i}}(f_{i,j})^{\Delta}, & \text{if } a_j \geq Q_{2\Delta}(a_j), \\ \mathcal{E}_{K_{pB_i}}(c' \cdot (Q_{2\Delta}(a_j + d_j) - d_j)) \times (\mathcal{E}_{K_{pB_i}}(f_{i,j})^{\Delta})^{-1}, & \text{if } a_j < Q_{2\Delta}(a_j). \end{cases}$$

where $()^{-1}$ denotes the modular inverse in the cyclic group defined by the Paillier's encryption scheme. The encrypted signal $\mathcal{E}_{K_{pB_i}}(y_i)$, with the buyer's identity information embedded into it in the form of a fingerprint, is finally sent to the buyer. Obviously, only the buyer can decrypt the fingerprinted signal values using his/her secret key K_{sB_i}. M also encrypts the remaining scaled and quantized approximation coefficients that do not carry a fingerprint, so as to hide these embedding positions. These approximation coefficients are encrypted in a block form with the public key of a buyer K_{pB_i}, instead of encrypting individual bits. After decryption, the buyer obtains the decrypted BF into which his/her fingerprint is embedded.

In FPSUM-HE both audio and video multimedia files are considered. Therefore, the explanation of the partitioning method for each type of the content is required. In the following, a step-by-step method of the audio and video file partitioning algorithms are explained.

- **Partitioning of an Audio File**

  1. The 3-level DWT is applied to an audio signal X to split it into approximation coefficients and detail coefficients. The reason to select 3-level DWT decomposition for an audio signal is to obtain a convenient trade-off between the robustness, capacity and transparency properties of watermarking.

  2. The level-3 approximation coefficients are divided into non-overlapping frames $\mathcal{F}_k$, with length of each frame equal to m, where m is the length of the fingerprint f_i.

  3. M selects a frame $\mathcal{F}_1$ using a secret key sk for embedding the fingerprint f_i.

  4. The rest of the frames are quantized by 2Δ, scaled by a constant c', and encrypted block-by-block with a buyer's public key K_{pB_i}, with each message block $(\mathcal{F}_k)_i < N$, to obtain $\mathcal{E}_{K_{pB_i}}(y_{\mathcal{F}_k}) = \mathcal{E}_{K_{pB_i}}(c' \cdot (Q_{2\Delta}(\mathcal{F}_k)))$. c' is also communicated to the buyer so that he/she can re-scale the entire content after decryption.

  5. The dither vector d_j is added to the frame selected by M for embedding a fingerprint f_i.

  6. After adding d_j to $\mathcal{F}_1$, $(\mathcal{F}_{1,j} + d_j)$ is quantized using 2Δ.

7. After quantization, the same dither d_j is subtracted from the quantized values $Q_{2\Delta}(\mathcal{F}_{1,j}+d_j)$ to yield $Q_{2\Delta}(\mathcal{F}_{1,j}+d_j)-d_j$.

8. $Q_{2\Delta}(\mathcal{F}_{1,j}+d_j)-d_j$ is then scaled by a factor c'.

9. The quantized and scaled coefficients $c'\cdot(Q_{2\Delta}(\mathcal{F}_{1,j}+d_j)-d_j)$ are then encrypted with K_{pB_i}, yielding $\mathcal{E}_{K_{pB_i}}(c'\cdot(Q_{2\Delta}(\mathcal{F}_{1,j}+d_j)-d_j))$.

10. The fingerprint f_i generated in Algorithm 1 is encrypted by MO with K_{pB_i}, to obtain $\mathcal{E}_{K_{pB_i}}(f_i)$.

11. $\mathcal{E}_{K_{pB_i}}(f_i)^{\Delta}$ or its modular inverse $(\mathcal{E}_{K_{pB_i}}(f_i)^{\Delta})^{-1}$ is multiplied with $\mathcal{E}_{K_{pB_i}}(c'\cdot(Q_{2\Delta}(\mathcal{F}_{1,j}+d_j)-d_j))$, depending on the value of $(\mathcal{F}_{1,j}+d_j)$. The resulting embedding equation can be summarized as follows:

$$\mathcal{E}_{K_{pB_i}}(y_{\mathcal{F}_{1,j}})=\begin{cases}\mathcal{E}_{K_{pB_i}}(c'\cdot(Q_{2\Delta}(\mathcal{F}_{1,j}+d_j)-d_j))\times\mathcal{E}_{K_{pB_i}}(f_{i,j})^{\Delta}, & \text{if } \mathcal{F}_{1,j}\geq Q_{2\Delta}(\mathcal{F}_{1,j}),\\ \mathcal{E}_{K_{pB_i}}(c'\cdot(Q_{2\Delta}(\mathcal{F}_{1,j}+d_j)-d_j))\times(\mathcal{E}_{K_{pB_i}}(f_{i,j})^{\Delta})^{-1}, & \text{if } \mathcal{F}_{1,j}< Q_{2\Delta}(\mathcal{F}_{1,j}).\end{cases} \tag{4.1}$$

12. The frames $\mathcal{E}_{K_{pB_i}}(y_{\mathcal{F}_{1,j}})$ and $\mathcal{E}_{K_{pB_i}}(y_{\mathcal{F}_k})$ are recombined and saved in a "text" format as BF.

13. An inverse 3-level DWT is performed on the detail coefficients to obtain SF in "wav" form. Other formats, such as binary and text, can also be used for the formation of SF.

- **Partitioning of a Video File**

1. In order to divide a video file into BF and SF, it is necessary to extract the significant frames from a video file, since not all the frames of the video contain relevant information. The video frames are arranged into groups of pictures (GoPs). A GoP includes the intra frames (I-frames) and inter-frames (P and B-frames). GoPs typically have 12 or 15 frames. A typical 15-frame GoP structure has one I-frame, four P-frames, and ten B-frames. The I-frames carry a complete video picture. These are coded without reference to other frames, whereas P and B-frames use pseudo-differences from the previous and next frame. Hence, these frames depend on each other. It is not advisable to analyze both intra and inter-frames, thus only intra-frames (key frames) that contain important information are used. For the detection of a key frame, the Canny-edge detection technique proposed by Khurana and Chandak (2013) is used, in which an edge difference is used to calculate the difference between two consecutive frames. Only when the difference exceeds a threshold, one

of the consecutive frames is considered as a key frame. The remaining frames, i.e. P and B-frames are saved in an original video format. The detailed description for key frames extraction from the video file is described in Algorithm 2.

Algorithm 2 Key Frame Extraction using Canny-Edge Detection Technique

procedure CANNY-EDGE DETECTION
Input: Video X with V frames
Output: Key frames of the video X
begin
  for all $1 \leq k \leq V$ **do** ▷ a loop over all the frames
    Read frame X_k and X_{k+1} as still images.
    Find the edge difference between X_k and X_{k+1} using the Canny-Edge Detector.
    diff(k)= $\sum_i \sum_j (X_k - X_{k+1})$ ▷ where, i, j are row and column index
  end for
Compute the mean M and standard deviation S as follows:

$$M \leftarrow \frac{\sum_{i=1}^{V-1} \text{diff}(i)}{V-1}$$

$$S \leftarrow \frac{\sqrt{\sum_{i=1}^{V-1} (\text{diff}(i)-M)^2}}{V-1}$$

Compute the threshold value T as:
  $T \leftarrow M + a' \times S$ ▷ Where, a' is a constant
Find the key frames as following:
  for all $1 \leq k \leq (V-1)$ **do**
    If diff(k) $\geq T$
    $V_{k+1} \leftarrow$ key frame
  end for
end procedure

2. The key frames obtained in Step 1 are converted from the RGB format to the $\mathcal{YUV}$ format. The $\mathcal{YUV}$ model defines a color space in terms of one luminance ($\mathcal{Y}$) and two chrominance ($\mathcal{UV}$) components. The weighted values of R, G and B are added together to produce a single $\mathcal{Y}$ (luminance) component. The chrominance components $\mathcal{U}$ and $\mathcal{V}$ are created by subtracting $\mathcal{Y}$ from B, and $\mathcal{Y}$ from R, respectively.

3. For each key frame, a $\mathcal{Y}$ component is selected. Typically, 3 or 4-level DWT is applied to $\mathcal{Y}$ to obtain the approximation and detail coefficients.

4. A few key frames ($\mathcal{I}_t$, where t is the number of selected key frames) are selected for embedding the fingerprint f_i. These frames are selected on the basis of time period, i.e. one key frame is selected after a duration of 40 secs.

5. The 3/4-level approximation coefficients a_t of the selected key frames $\mathcal{I}_t$ are divided into non-overlapping blocks $\mathcal{B}_k$, with length of each block equal to m, where m is the length of the fingerprint f_i.

6. All the remaining key frames are quantized by 2Δ, scaled by a constant c', and encrypted block-by-block with a buyer's public key K_{pB_i}.

7. In the selected key frames, a dither d_j is added to the coefficients $a_{t,j}$ to yield $a_{t,j}+d_j$.

8. $a_{t,j} + d_i$ are quantized using 2Δ to produce $Q_{2\Delta}(a_{t,j} + d_j)$.

9. After quantization, the same d_j is subtracted from the quantized values $Q_{2\Delta}(a_{t,j}+d_j)$ to yield $Q_{2\Delta}(a_{t,j} + d_j) - d_j$.

10. $Q_{2\Delta}(a_{t,j} + d_j) - d_j$ is then scaled by a factor c'.

11. The quantized and scaled coefficients $c' \cdot (Q_{2\Delta}(a_{t,j}+d_j)-d_j)$ are then encrypted with K_{pB_i}, yielding $\mathcal{E}_{K_{pB_i}}(c' \cdot (Q_{2\Delta}(a_{t,j} + d_j) - d_j))$.

12. The fingerprint f_i is encrypted by MO with K_{pB_i} to obtain $\mathcal{E}_{K_{pB_i}}(f_i)$.

13. $\mathcal{E}_{K_{pB_i}}(f_i)$ obtained from MO is added to the encrypted approximation coefficients using Equation (4.1) to form BF in "text" form. BF can also be saved in other formats such as binary, and bitmap (bmp) image files.

14. The index of the key frames is also scaled, encrypted and added into BF for file re-construction at the user end.

15. An inverse 3/4-level DWT is applied on the detail coefficients, and then these obtained values, the P and B-frames, and the audio of the original video file X constitute SF in a compressed (ZIP) format.

4.5.1.3 Base File Distribution Protocol

When a buyer B_i is interested in buying a particular content X, his/her associated SP provides him/her the details of the merchant M that has the requested content. In order to obtain a content X from M, B_i follows the following protocol:

1. The buyer negotiates with M to set-up an agreement (AGR) that explicitly states the rights and obligations of both parties and specifies the content X. AGR uniquely binds this

particular transaction to X. During the negotiation process, B_i uses his/her pseudonym P_{B_i} to keep his/her anonymity.

2. After the negotiation, B_i generates a key pair $(K^*_{pB_i}, K^*_{sB_i})$, signs the public key with his/her private key, and sends $\text{Sign}_{\text{B}_\text{i}}(K^*_{pB_i}, P_{B_i})$ to CA_R. CA_R verifies $\text{Sign}_{\text{B}_\text{i}}(K^*_{pB_i}, P_{B_i})$ using the public key of B_i. If valid, he/she generates an anonymous certificate $\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$ and sends it to B_i. Then, B_i sends $\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$, AGR, P_{B_i} and $\text{Sign}_{K^*_{pB_i}}(AGR)$ to M.

3. M verifies the received certificate, using CA_R's public key, and the signature of the agreement using the certified key. If the received data is valid, then M generates a transaction ID (TID) for keeping a record of the transaction between him/her and B_i, and sends a request for a fingerprint to MO by sending $\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$, $\text{Cert}_{CA_R}(M)$, TID, AGR, P_{B_i} and $\text{Sign}_{K^*_{pB_i}}(AGR)$. If the received certificates and signatures are not valid, then the transaction is terminated by M.

4. MO validates the certificates and signatures of M and B_i from CA_R. After successful verification, MO generates a Nuida et al.'s c_0-secure codeword f_i using Algorithm 1 against a TID sent by M. MO then sends $\mathcal{E}_{K^*_{pB_i}}(f_i)$, $E_{K_{pM}}(m)$, and $\text{Sign}_{MO}(\mathcal{E}_{K^*_{pB_i}}(f_i), K^*_{pB_i}, \text{Sign}_{K^*_{pB_i}}(AGR))$ to M. MO stores $K^*_{pB_i}$, $\text{Cert}_{CA_R}(M)$, $\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$, $\text{Sign}_{K^*_{pB_i}}(AGR)$, AGR, and $E_{K^*_{pB_i}}(f_i)$ against TID.

5. After receiving the encrypted fingerprint from MO, M embeds the fingerprint code in the encrypted domain by using the file partitioning algorithm described in Section 4.5.1.3 without knowing about the plain-text fingerprint f_i.

6. M sends $\mathcal{E}_{K^*_{pB_i}}(BF)$ to B_i and stores $K^*_{pB_i}$, $\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$, AGR, $\mathcal{E}_{K^*_{pB_i}}(f_i)$, $\text{Sign}_{K^*_{pB_i}}(AGR)$, and $\text{Sign}_{MO}(\mathcal{E}_{K^*_{pB_i}}(f_i), K^*_{pB_i}, \text{Sign}_{K^*_{pB_i}}(AGR)$ against TID.

7. B_i decrypts $\mathcal{E}_{K^*_{pB_i}}(BF)$ with $K^*_{sB_i}$ and obtains a fingerprinted BF.

4.5.1.4 Supplementary File Distribution Protocol

When a buyer B_i requests a particular content X from his/her associated SP, SP directs him/her to M for BF acquisition, whereas for SF, the following protocol is followed:

1. On receiving a request for X from B_i, SP searches for in his/her own file index. If not found, he/she then searches within his/her group of peers. If the particular content is found

within the group, he/she displays the list of the buyers (peers) having that particular file, and also displays their tail nodes to act as middle nodes between the content providing peer and the requesting peer. If *SP* is unable to find the file within his/her group, he/she sends a request for the file to other connected *SP*s. The other *SP*, on finding the particular content provider, sends the response to the requesting *SP*. *SP* then establishes a path between the receiving peer P_a and the content providing peer P_b.

2. On receiving a file request from P_a, P_b decides whether or not to be the file provider, depending on the reputation of P_a. If P_b decides to be a file provider, then he/she replies to the query of P_a through his/her tail node T_b.

3. P_a, using his/her pseudo-identity PI_{P_a}, initiates the authentication process to verify the pseudo-identity PI_{P_b} of P_b. P_a sends an authentication request to P_b through the anonymous path, $P_a \rightarrow T_a \rightarrow T_b \rightarrow P_b$. Thus, a two-party authenticated key exchange protocol is established between P_a and P_b. Fig. 4.4 describes the authentication process between P_a and P_b.

4. P_a chooses $\Upsilon_1 \in [1, \ldots, P-1)$ randomly. Then he/she uses his/her private key $K_{S_{Pa}}$ to sign $\{ID_{P_a}, \text{Cert}_{CA_R}(P_a), r, \Upsilon_1\}$. PI_{P_a} also computes g^{γ_1} with publicly known parameters P and g for generation of a session key. γ_1 is chosen randomly from $[1, \ldots, Q)$ to generate a session key.

5. g^{γ_1} is calculated as follows:

$$g^{\gamma_1} := g^{\gamma_1} \bmod P.$$

6. To send an authentication request to P_b, P_a calculates u as follows:

$$u = h(PI_{P_a}, \Upsilon_1, g^{\gamma_1}).$$

where, h is a hash function with k bits and is defined as: $h = \mathbb{Z}_n^* \times \{0,1\}^w \times \mathbb{Z}_p^* \rightarrow [0,1]^k$.

7. P_a sends $\{PI_{P_a}, \Upsilon_1, g^{\gamma_1}\}$ to P_b.

8. After receiving the authentication request, P_b computes u' to verify the authentication request. u' is obtained as following:

$$u' = h(PI_{P_a}, \Upsilon_1, g^{\gamma_1}).$$

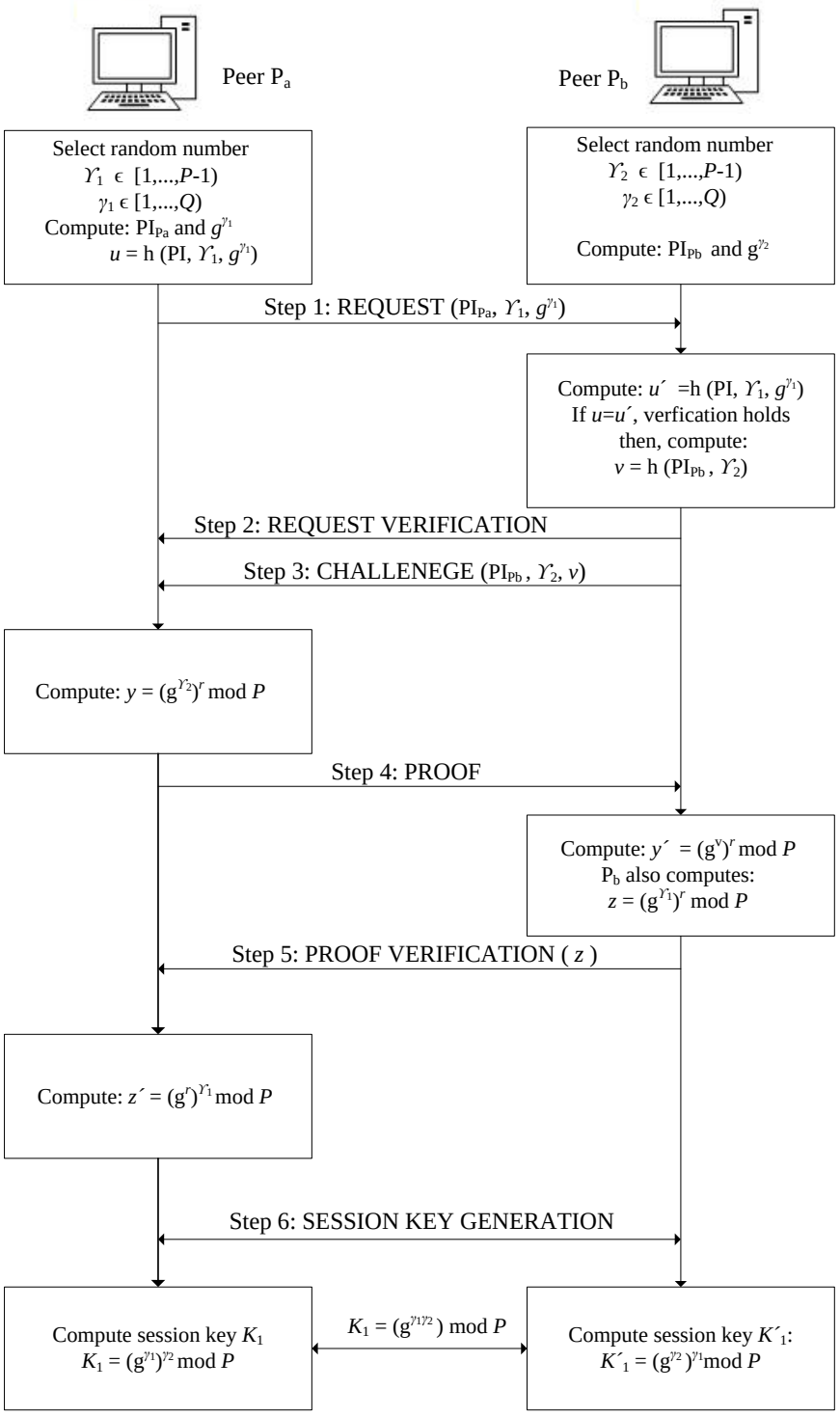


FIGURE 4.4: Two-Party anonymous AKE Protocol

9. Once verified, P_b randomly chooses a number $\Upsilon_2 \in [1, \ldots, P-1)$. Then he/she uses his/her private key $K_{S_{Pb}}$ to sign $\{ID_{P_b}, \text{Cert}_{CA_R}(P_b), r, \Upsilon_2\}$.

10. P_b also chooses a number $\gamma_2 \in [1, \ldots, Q)$ randomly and computes g^{γ_2} for generation of a session key.

11. g^{γ_2} is calculated as following:

$$g^{\gamma_2} := g^{\gamma_2} \bmod P.$$

12. P_b computes v as follows:

$$v = h(PI_{P_b}, \Upsilon_2).$$

and then, sends $\{PI_{P_b}, \Upsilon_2, v\}$ as a challenge to P_a.

13. As a proof, P_a calculates y and sends it to P_b. y is calculated as following:

$$y = (g^{\Upsilon_2})^r \bmod P.$$

14. As a proof verification, P_b calculates y' as follows:

$$y' = (g^r)^{\Upsilon_2} \bmod P.$$

15. If the verification holds, P_b sends $z = (g^{\Upsilon_1})^r \bmod P$ to P_a.

16. P_a then computes $z' = (g^r)^{\Upsilon_1} \bmod P$ to complete the last step of authentication.

17. When the authentication is successfully completed, P_a computes K_1 as follows:

$$K_1 = (g^{\gamma_2})^{\gamma_1} \bmod P,$$

and P_b computes $K_1^{'}$ as follows:

$$K_1^{'} = (g^{\gamma_1})^{\gamma_2} \bmod P.$$

Thus resulting in $K_1 = K_1^{'} = g^{\gamma_1\gamma_2} \bmod P$.

18. P_a and P_b use K_1 as their session key for encryption of SF.

19. P_b encrypts SF using the session key K_1, and sends E_{K_1} (SF) to P_a through T_b and T_a.

20. P_a decrypts $E_{K_1}(SF)$ with K_1 and obtains a decrypted SF.

4.5.1.5 Traitor-Tracing Protocol

A traitor-tracing protocol is executed by MO on receiving a request from M that a codeword pc has been extracted from a pirated copy Y', and the buyer corresponding to pc needs to be identified. Before the execution of the traitor-tracing protocol, M needs to extract pc from Y', and this is achieved using the fingerprint extraction process. The fingerprint extraction is similar to the embedding procedure. It does not require the original multimedia signal. The watermark extraction procedure can be summarized as follows:

1. Let Y be the fingerprinted signal, which is decomposed through DWT with the same wavelet basis used in the fingerprint insertion step described in Section 4.5.1.1.

2. The decomposition gives the approximation coefficient matrix in which the pirated code $pc \in \{0,1\}^*$ is embedded.

3. The code pc is extracted by applying the secret key sk that was used to specify the embedding positions.

4. Each approximation coefficient a_{pc} in the embedding position is quantized using the corresponding quantization step size Δ.

5. If the value is even, the information bit is regarded as 0, else 1. The fingerprint extraction process can be summarized as follows:

$$y_i^{'} = \begin{cases} 0, & \text{if } Q_\Delta(a_{pc} + d_j) - d_j \text{ is even,} \\ 1, & \text{if } Q_\Delta(a_{pc} + d_j) - d_j \text{ is odd.} \end{cases}$$

Once pc is extracted by M from Y', he/she sends pc to MO. MO performs the tracing algorithm of Nuida et al.'s codes as described in Algorithm 3 to identify the colluder(s). In the tracing algorithm, pc provided by M, and a bias vector p generated by MO in the fingerprint generation algorithm, are given as inputs. p is used to generate the fingerprint matrix F for the identification of the colluder(s). The score of the pirate is calculated as per Algorithm 3. The output of this tracing algorithm is a user with the highest score S_i. The real identity of a user is not known to MO, only the pseudo-identity of the guilty buyer is revealed. MO retrieves a TID that contains the fingerprint f_i from his/her database for the arbitration and identification protocol.

Algorithm 3 Traitor Tracing

procedure NUIDA ET AL.'S TRACING ALGORITHM
Input parameters: pc, p, F
Output parameter: S_i
begin
$\sigma = \sqrt{\frac{1-p}{p}}$ ▷ Calculate the score S_i of the pirate using $\sigma(p)$
if ($pc = 1$ and $F_{i,j} = 1$) **then**
$S_i^j \leftarrow \sigma(p^{(j)})$
else if ($pc = 1$ and $F_{i,j} = 0$) **then**
$S_i^j \leftarrow -\sigma(1 - p^{(j)})$
else if ($pc \in \{0, ?\}$ and $F_{i,j} = 1$) **then**
$S_i^j \leftarrow -\sigma(p^{(j)})$
else if ($pc \in \{0, ?\}$ and $F_{i,j} = 0$) **then**
$S_i^j \leftarrow \sigma(1 - p^{(j)})$
end if
return S_i
end procedure

4.5.1.6 Dispute Resolution and Identification Protocol

The goal of the dispute resolution and identification protocol is to reveal the real identity of the traitor or reject the claims made by M. The protocol is performed between M, MO, CA_R and J without involving the buyer B_i. The following steps are performed in the identification protocol:

1. MO sends Y', pc, $\text{Cert}_{CA_R}(K^*_{pB_i})$, AGR, $\text{Sign}_{K^*_{pB_i}}(AGR)$, $\mathcal{E}_{K^*_{pB_i}}(f_i)$, $\text{Sign}_{MO}(\mathcal{E}_{K^*_{pB_i}}(f_i), K^*_{pB_i})$, $E_{K_{p_{MO}}}(f_i)$ and $\text{Sign}_{K^*_{pB_i}}(AGR)$ to J.

2. J verifies the validity of all the certificates and the signatures from Cert_{CA_R}.

3. If the certificates and the signatures are valid, J then asks MO to decrypt $E_{K_{p_{MO}}}(f_i)$.

4. MO decrypts $E_{K_{p_{MO}}}(f_i)$ using his/her private key, encrypts f_i with a public key of J and sends $E_{K_{pJ}}(f_i)$ to J.

5. J obtains f_i by decrypting $E_{K_{p_{MO}}}(f_i)$ with his/her secret key.

6. J computes the correlation between pc and f_i to check the similarity between the two codes by using the following equation:

$$\text{Corr}(pc, f_i) = \frac{\sum_{j=1}^{m} (-1)^{pc_j \oplus f_{i,j}}}{m}.$$

7. If pc and f_i match with a high correlation, J then requests CA_R to provide the real identity of the buyer. Otherwise, the buyer is proved innocent.

4.6 Theoretical and Experimental Results

This section examines how the design goals of FPSUM-HE described in Section 4.4.3 are achieved. The security analysis provides a formal and informal analysis concerning the correctness of the protocols of FPSUM-HE in terms of security and privacy. The performance analysis examines the performance of the protocols of FPSUM-HE in terms of robustness, imperceptibility, computational and communicational costs and cryptographic overhead.

4.6.1 Security Analysis

In this section, an formal and informal security analysis are provided about the security and privacy of FPSUM-HE according to the design requirements and the attack model presented in Sections 4.4.3 and 4.4.4. Several attack scenarios presented in Sections 4.4.4.3, 4.4.4.4 and 4.4.4.2 are discussed, which can occur during each phase of the *BF* and *SF* distribution protocols execution.

4.6.1.1 Formal Analysis of the *BF* Distribution Protocol

Formal proofs are provided in this section to analyze the security of the *BF* distribution protocol.

Theorem 4.1. *A framing of an honest buyer B_i by a malicious merchant M is not possible in the BF distribution protocol.*

Proof. M knows only about $\mathcal{E}_{K^*_{pB_i}}(f_i)$ and $\mathcal{E}_{K^*_{pB_i}}(BF)$ and has no knowledge about the buyer's private key $K^*_{sB_i}$. Therefore, M does not know about the fingerprinted copy that B_i obtains after decrypting $\mathcal{E}_{K^*_{pB_i}}(BF)$ with $K^*_{sB_i}$. It means that M cannot frame B_i by distributing forged copies of the content. Furthermore, $\text{Sign}_{MO}(\mathcal{E}_{K^*_{pB_i}}(f_i), K^*_{pB_i}, \text{Sign}_{K^*_{pB_i}}(AGR))$ explicitly binds f_i to AGR, which specifies the content X. Thus, it is impossible for M to frame B_i. Also, B_i generates a one-time anonymous key pair $(K^*_{pB_i}, K^*_{sB_i})$ for the transaction with M that prevents M to frame B_i by sending $\mathcal{E}_{K^*_{pB_i}}(f_i)$ from previous transactions. Therefore, framing an honest buyer by M is not possible since he/she cannot forge any evidence. □

Theorem 4.2. *The buyer B_i accused of an illegal re-distribution cannot claim that a piracy is originated from the merchant M.*

Proof. From the perspective of M, FPSUM-HE is secure and fair because B_i has no idea about the original content and the embedded fingerprint in BF. B_i cannot claim that a pirated copy Y' is created by M since only B_i can decrypt the $\mathcal{E}^*_{K_{pB_i}}(f_i)$ or $\mathcal{E}^*_{K_{pB_i}}(BF)$ with his/her $K^*_{sB_i}$. Also, MO is an entity trusted by both B_i and M (as described in Section 4.4.2.2), thus B_i cannot accuse MO of collaborating with M to frame him/her. Moreover, the fingerprint is embedded into the selected positions of the content. Thus, the probability to find the exact locations of the embedded fingerprint is quiet low. Moreover, FPSUM-HE provides a traitor-tracing mechanism to unambiguously identify a copyright violator once a pirated copy Y' is found. □

Theorem 4.3. *The buyer's privacy is well protected in the BF distribution protocol.*

Proof. The essential protection of the buyer's privacy is by taking advantage of the one-time anonymous public and private key pair. However, this anonymity is revocable since B_i computes his/her pseudo-identity with a help of CA_R, and his/her one-time anonymous key pair generated for a transaction with M is also certified by CA_R. Under the assumption of CA_R's existence, B_i can keep his/her real identity unexposed unless he is found guilty by J in a dispute resolution protocol.

□

Theorem 4.4. *A malicious buyer cannot deduce a real identity of any buyer from his/her pseudo-identity.*

Proof. A pseudo-identity of B_i is obtained from a cryptographic hash function. Thus, any attempt of de-anonymization attack by a malicious buyer is withstood by the collision resistance of the hash function, i.e. it is computationally infeasible to find a pair (u_1, u_2) such that $h(u_1) = h(u_2)$. Moreover, for a hash function with w-bit hash values, $2^{w/2}$ calculations are required to find a collision with probability $1/2$, which is infeasible for $w \geq 128$. In FPSUM-HE, SHA-1 is considered with $w = 160$ bits for high security such that it is computationally infeasible for an attacker to compute 2^{80} calculations to find a real identity from a pseudo-identity. Furthermore, a malicious buyer cannot use the pseudo-identity of another buyer because he/she does not know the secret number r shared by the buyer with CA_R. □

4.6.1.2 Security Attacks on the *BF* Distribution Protocol

This section analyzes the security of the *BF* distribution protocol and explains how it fulfils the design requirements presented in Section 4.4.3.

Traceability

Once a pirated copy Y' is found, the traitor-tracing algorithm of Nuida et al. (2007) c_0-secure codes is used by M to trace the copyright violator with the help of MO. The traitor tracing algorithm (Algorithm 3) employs a scoring technique that outputs a guilty user with the highest score S_i. Once the algorithm outputs a guilty user, his/her identity is revealed by J with the help of CA_R.

Unlinkability

In spite of B_i's anonymity, the transactions carried out by the same pseudonym or anonymous key pair are linkable to one another, and there are still risks for B_i's private information to be inferred through data mining techniques. The solution to this problem is to allow B_i to compute multiple pseudonyms and anonymous key pairs and randomly chooses one of each for each transaction.

Collusion resistance

Nuida et al.'s codes are c_0-secure with ε-error with $c \leq c_0$ (where c is the number of pirates). In FPSUM-HE, $c_0 = 3$ with $\varepsilon = 10^{-3}$ and $N = 10^5$ (N = number of users) are considered, thus a code of size $m = 267$ bits is obtained. This code is then embedded into the content to uniquely identify the user. As long as c remains lower than c_0 and the piracy tracing Algorithm 3 is followed, the copyright violator can be identified successfully. Thus, the proposed scheme offers resistance against three colludes. A value of $c_0 > 3$ can also be considered. However, this large value of c_0 results in an increased length m of the codeword, which will provide high collusion resistance but at a cost of lower imperceptibility. The value of c_0 is decided keeping in mind the desired security level of the system.

4.6.1.3 Formal Analysis of the *SF* Distribution Protocol

Formal proofs are provided in this section to analyze the security of the *SF* distribution protocol.

Theorem 4.5. *Assuming that it is computationally infeasible to solve the discrete logarithms problem, if a malicious buyer $\hat{E}$ interacts the protocol with a responder R to impersonate the initiator I and convince R that he/she is I, then, the probability that $\hat{E}$ succeeds is $1/P$.*

Proof. In order to convince the responding peer R about his/her identity, a malicious peer $\hat{E}$ needs to know the secret number r. However, the probability of $\hat{E}$ guessing correctly r is $1/P$ (P is the cardinality of the finite cyclic group $\mathbb{G}$). According to the authentication procedure steps in Section 4.5.1.4, $\hat{E}$ must know the secret number r so that he/she can impersonate the initiator I. Because it is computationally infeasible to solve the discrete logarithms problem, $\hat{E}$ cannot compute r. Thus, $\hat{E}$ can guess a secret number r' by computing $y = (g^{\upsilon_2})^{r'} \bmod P$. However, the probability of soundness that $\hat{E}$ guesses r is $1/P$, i.e. the probability that $\hat{E}$ succeeds is $1/P$. □

Theorem 4.6. *An attempt by a malicious peer $\hat{E}$ to access and modify messages between the peers I and R without either peer knowing that the link between them has been exposed is unsuccessful in the SF distribution protocol.*

Proof. The *SF* distribution protocol defends against such an attack (also known as man-in-the-middle attack (MIMA)) by making use of a zero-knowledge proof-of-identity-based authentication. In the authentication step, the proof, tail node's information, and the key exchanged data are bound together with a peer's pseudo-identity. By doing so, any attempt by an attacker to modify the identity messages would not pass the verification of genuine protocol participants. MIMA can be successful in the protocol if a malicious peer $\hat{E}$ is able to convince peer I or peer R that $T_{\hat{E}}$, which is indeed the tail node of $\hat{E}$, is T_I or T_R, a tail node of I or R. MIMA is based on two possible scenarios: (1) R does not receive I's query q, or (2) R receives I's q.

In case 1, since R does not receive q, R does not respond. In this case, to cheat R, $\hat{E}$ has to (1) forward I's query q directly to R, or (2) forge q' and send it to R. For (1), $\hat{E}$ acts like other relaying nodes in the transmission. Since $\hat{E}$ does not modify anything, R connects with T_I through T_R directly. Thus, $\hat{E}$ cannot cheat anyone. For (2), the possible modification on q by $\hat{E}$ leads to two sub-cases: (*i*) $\hat{E}$ replaces $\text{Cert}_{CA_R}(I)$ with his/her $\text{Cert}_{CA_R}(\hat{E})$ in q such that R considers $\hat{E}$ as an initiator. This is useless for $\hat{E}$'s attack because it would fail in the later verification without a valid PI_I. In the second sub-case (*ii*), $\hat{E}$ modifies q to $q' = (\text{Cert}_{CA_R}(I), T_{\hat{E}}, fl)$ (fl is the index of the requested file). After receiving q', R replies to I with $(\text{Cert}_{CA_R}(R), T_R, fl)$. $\hat{E}$ intercepts this reply, modifies this message to $(\text{Cert}_{CA_R}(R), T_{\hat{E}}, fl)$, and delivers it to I. Here E has to modify T_R to $T_{\hat{E}}$, otherwise I would ask T_I to contact T_R. In the following step of the authentication procedure (cf. Section 4.5.1.4), I randomly chooses $\upsilon 1$ and computes $u = h(PI_R, T_{\hat{E}}, \upsilon_1)$. Then,

I sends them back to $T_{\hat{E}}$. Upon intercepting this message, $\hat{E}$ has only two choices of how to continue his/her intruding actions:

1. $\hat{E}$ relays the message to R without modification. Then R computes $u' = h(PI_R, T_R, \upsilon_1)$. According to the pseudo-random feature of the hash function, $u \neq u'$. R terminates the authentication procedure, and the attack fails.

2. $\hat{E}$ computes $u'' = h(PI_R, T_R, \upsilon_1)$ and sends it to R. In such a case, $u'' = u'$. R continues the authentication. R then sends a challenge v to $T_{\hat{E}}$. $\hat{E}$ cannot know v in advance and the best choice for $\hat{E}$ is to deliver the challenge to I.

In (2), I and R continue the authentication procedure until the point where I generates a proof and sends it back. Since the secret number r is unknown to $\hat{E}$, $\hat{E}$ cannot forge a proof to pass R's verification. If $\hat{E}$ changes r so as to pass the verification, it must guess the value of r, and change the value of y accordingly. Since the probability of such a successful guess is $1/(P-1)$, it is infeasible. Thus, MIMA attempts made by $\hat{E}$ in case 1 fail.

In case 2, R receives I's query q. In this case, R has multiple queries containing an identical pseudo-identity with different tail nodes. Aware of being under attack, R can simply discard the query, or randomly select one of them to initiate the authentication procedure. The remaining analysis is similar to case 1. □

4.6.1.4 Security Attacks on the *SF* Distribution Protocol

The security of the *SF* distribution protocol against an attack model presented in Sections 4.4.4.3 and 4.4.4.4 are analyzed in this section.

Anonymity

The degree of anonymity is determined from the probability that the attacker can identify the initiating peer I (or the responding peer R). Assuming that the total number of peers in a system is N, the anonymity of FPSUM-HE is analyzed from two perspectives: initiating peer I (or the responding peer R) and a middle node.

- **The initiator (or the provider):** Every peer in the session network has the same probability of serving as I (or R). Therefore, I and R can correctly guess each other's identity with the same probability $1/(N-1)$.

- **A middle node:** The probability that a middle node randomly guesses which node I (or R) is $1/(N-1)$. However, if the number of middle nodes is known to be k, then the probability of correctly guessing the identity of I (or R) is changed to $1/(N-k)$.

Leakage of a secret number r

Because the secret number r is a key used in FPSUM-HE, it would be a serious problem if r is leaked. Hence, in the scheme, CA_R updates r when a new peer joins the system or an old peer leaves the network, so that new peers do not know previous r and old peers do not know new r. Hence, CA_R generates a new number r, encrypts it with each peer's public key, and sends it to each peer. Then, each peer decrypts it using his/her private key and obtains r.

4.6.1.5 Collusion Attacks

This section presents the robustness of the fingerprinting scheme against the linear (averaging) and non-linear (minimum, maximum and median) collusion attacks presented in Section 4.4.4.2. The attacks are performed on a sample video file "Dragon" (details of "Dragon" video file are provided in Table 4.3) with varying number of colluders U. Under the averaging attack, each pixel in the pirated video is average of the corresponding pixels of the fingerprinted videos associated with the colluders U. For minimum, maximum and median attacks, each pixel in pirated video is the minimum, maximum or median, of the corresponding pixels of the fingerprinted video.

Table 4.1 shows the number of colluders U which have been successfully traced through Nuida et al. (2007) codes tracing Algorithm 3. In almost all the cases, the colluders have been successfully traced by analyzing a pirated video copy Y'. In order to test the resistance of the fingerprint against more than 3 colluders, the fingerprint codewords are generated using $c_0 = 4$ and $c_0 = 5$, which results into codewords with an increased length m. The reason that the number of colluders U are considered up to 5 is due to the fact that an increase in U degrades the quality of the content. The larger value of c_0 results in a larger code length m, which degrades the quality of the content and requires more embedding capacity. Thus, to provide a better trade-off between collusion resistance property and imperceptibility, a lower value of c_0 is selected.

TABLE 4.1: Resistance against collusion attacks

No. of Colluders	**No. of Colluders Detected for Attacks**			
U	**Average**	**Minimum**	**Maximum**	**Median**
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	4	4	5

4.6.2 Performance Analysis

This section presents a performance analysis of FPSUM-HE in terms of robustness, imperceptibility, computational (especially cryptographic) effort required by the entities and the communication cost. To show the performance of FPSUM-HE, the experiments are carried out in Matlab 7.0 and Java on three audio and three video files, with varying sizes, on a workstation equipped with an Intel i-7 processor at 3.4 GHz and 8 GB of RAM. To partition the files into *BF* and *SF*, the experiments were conducted in Matlab 7.0 in which the DWT is used to decompose the original files into approximation and detail coefficients. The fingerprint generation protocol is also implemented in Matlab 7.0. The embedding of the fingerprint and the distribution phase of *BF* and *SF* are executed in the Java programming language.

The simulation parameters for fingerprint generation, *BF* and *SF* generation, and *BF* and *SF* distribution protocols are presented in Table 4.2.

TABLE 4.2: Simulation parameters

Name	**Value**	**Description**
N	10^5	No. of users
c_0	3	Coalition Size
ε	10^{-3}	Error probability
L	3/4	Levels of DWT decomposition
Δ	0.5	Quantization step size
c'	5	Scaling factor
$\mathcal{E}(\cdot)$	1024-bits	Paillier encryption
$\mathcal{D}(\cdot)$	1024-bits	Paillier decryption
d_j	$[-\Delta, \Delta]$	Dither vector
a'	2	Constant in key frame's threshold calculation
$h(\cdot)$	160-bits	SHA-1 function
P	1024-bits	Prime number $\in$ finite cycle group $\mathbb{G}$
Q	160-bits	Prime number that divides $P-1$
r	1024-bits	Secret number used in the pseudo-identity generation
υ_1/υ_2	1024-bits	Secret numbers used in authentication
γ_1/γ_2	160-bits	Secret numbers used for session key generation
T_a/T_b	2/3	Tail nodes in onion routing

4.6.2.1 Analysis of Audio Fingerprinting

In this section, the results of three audio files, namely, "LoopyMusic, "Hugewav" and "Aasan Nai Yahan" are presented in terms of imperceptibility, robustness, computational, communicational and cryptographic costs. The details of the three audio files are presented in Table 4.3. The simulation parameters are those presented in Table 4.2. However, in the experiments for the partitioning protocol of the "LoopMusic" and "Hugewav" files, level-3 DWT decomposition with a 4-coefficient Daubechies (*db*4) filter is used, and for "Aasan Nai Yahan", level-4 DWT decomposition with a *db*4 filter is used. The levels of the DWT decomposition are selected to provide a good trade-off between robustness, capacity and imperceptibility. The experiments and simulations are performed for each channel of audio signals separately. *SF* is formed with double-bit precision values since Matlab 7.0 stores signals as double-precision values and, otherwise the file reconstruction at the user end would not be perfect due to quantization errors.

TABLE 4.3: Details of audio files

Details	**Loopy Music**	**Hugewav**	**Aasan Nai Yahan**
Time Length (min:sec)	00:10	00:17	03:34
File Size (MB)	0.89	2.97	36.01
Format	WAV	WAV	WAV
Bits per Sample	16	16	16
Sample Rate (Hz)	44100	44100	44100
Channel Mode	Mono	Stereo	Stereo
Base File Size (MB)	0.52	0.88	9.80
Supplementary File Size (MB) with double-bit precision	1.79	5.94	72.16

Transparency

The Objective Difference Grade (ODG) is the output variable obtained from the perceptual evaluation of audio quality (PEAQ) measurement algorithm specified in the ITU-R BS.1387 standard (Thiede et al., 2000). It corresponds to the subjective grade used in human based audio tests. The ODG ranges from 0 to −4 (corresponding to imperceptible to very annoying) as shown in Table 4.4. To measure the ODG between the original and fingerprinted audio signals the *Opera* (1999) software is used.

TABLE 4.4: Objective Difference Grades (ODG)

ODG	Impairment Description	Quality
0.0	Imperceptible	Excellent
−1.0	Perceptible, but not annoying	Good
−2.0	Slightly annoying	Fair
−3.0	Annoying	Poor
−4.0	Very annoying	Bad

Table 4.5 presents the imperceptibility results as ODG of the three fingerprinted audio files. To evaluate the imperceptibility of each audio file, 20 different Nuida et al.'s c_0-secure fingerprints are generated and embedded into the selected approximation coefficients using the embedding algorithm of Prins et al. (2007). The computed range of ODG values for each file varies. For example, in case of "LoopyMusic", 20 different fingerprints produce 20 different fingerprinted audio files with ODG values in the range $[-0.89, -0.40]$. Similarly, for "Hugewav" and "Aasan Nai Yahan" audio files, the computed ODG values are in the range $[-1.25, -0.71]$ and $[-1.52, -0.90]$, respectively. This variation in the ODG values depends on the embedded fingerprint. Some embedded fingerprints result in the fingerprinted content with ODG values tend towards a grade of slightly annoying and fair. On the other hand, a few embedded fingerprints result in the fingerprinted audio files with ODG values tending towards a grade of imperceptible and excellent. Thus, on average, the fingerprinted audio files show convenient behaviour in terms of imperceptibility with the ODG values of three audio files in the range $[-1.20, 0.00]$.

TABLE 4.5: ODG of audio files

Audio Files	ODG
Loopy Music	−0.48
Huge Wave	−0.98
Aasan Nai Yahan	−1.20

Robustness against signal processing attacks

The signal processing attacks mentioned in Section 4.4.4.1 are performed on an audio file "LoopyMusic" to assess the robustness of the fingerprint. The bit error rate (BER) and normalized correlation (NC) are used to evaluate the robustness between the original fingerprint and the extracted fingerprint.

The BER is defined as follows:

$$BER = \frac{\sum_{j=1}^{m} f_{i,j} \oplus f'_{i,j}}{m},$$

where, $\oplus$ denotes the exclusive OR operation between the original fingerprint f_i and the extracted fingerprint f'_i respectively, i is an index of the buyer in FPSUM-HE, j is equal to the length of the fingerprint and m is the size of the fingerprint code. BER values close to zero indicate robustness against signal processing attacks.

NC is defined as follows:

$$NC = \frac{\sum_{j=1}^{m} f_{i,j} f'_{i,j}}{\sum_{j=1}^{m} \sqrt{(f_{i,j})^2}\sqrt{(f'_{i,j})^2}}.$$

If NC is close to 1, then the similarity between f_i and f'_i is very high. If NC is close to 0, then the similarity between f_i and f'_i is very low.

The NC and BER values for the re-quantization, re-sampling, MP3 compression and AWGN attacks on an audio file are summarized in Table 4.6.

Table 4.6: Robustness of an audio file against signal processing attacks

Attacks	**Parameters**	**BER**	**NC**	**Traceability**
Re-quantization	16-8-16 bits	0.07	0.951	Yes
Re-sampling	44.1-22.05-44.1 kHz	0.11	0.902	Yes
MP3 Compression	256 kbps	0.09	0.912	Yes
AWGN	18 dB	0.13	0.882	Yes

The results in Table 4.6 shows that the selected embedding algorithm (Prins et al., 2007) provides better performance against common signal processing attacks. The algorithms have good NC and BER values against various attacks for “LoopyMusic”. The minimum BER and the maximum BER values for “LoopyMusic” are 7% and 13% respectively against different attacks. Moreover, the last column of Table 4.6 shows that the fingerprint of a buyer is traceable against these common signal processing attacks. Thus, these results indicate that the fingerprint embedding algorithm satisfies the fingerprint’s robustness requirement.

Computation and Communication Time

In this section, the performance of FPSUM-HE is discussed in terms of computation and communication time. The time taken to generate the collusion-resistant fingerprint and creating *BF* and *SF* is considered as a computation time. For *BF* and *SF* generation, the implementation of file partition protocol in Section 4.5.1.2 contributes to the total computation time. Table 4.7 shows the CPU time of three audio files.

TABLE 4.7: Computation time of an audio file

File Name	CPU Time (secs)			
	Fingerprint generation	***BF* generation**	***SF* generation**	**Total Time**
LoopyMusic	6.01	14.08	0.03	20.13
Hugewav	6.01	31.15	0.18	37.34
Aasan Nai Yahan	6.01	181.39	1.19	188.60

Table 4.8 presents the distribution of the *BF* generation's CPU time. In the file partitioning algorithm (cf. Section 4.5.1.2), the DWT is applied only once to the content to obtain the approximation and detail coefficients. *M* stores the approximation and detail coefficients of each file, and thus avoids the costs of applying the DWT and creating *SF* every time an audio file is requested by a buyer. The embedding part of the *BF* generation process includes the time taken to perform the quantization and encryption on the approximation coefficients. Hence, from Table 4.8, it is evident that the embedding part of the *BF* generation process is the major contributor in the total computation time of FPSUM-HE.

TABLE 4.8: Details of a computation time of an audio file

File Name	CPU Time (secs)			
	DWT	**Embedding Process**		**Total Time**
		Quantization	**Encryption**	
LoopyMusic	0.06	0.02	14.00	14.08
Hugewav	0.15	0.02	31.07	31.15
Aasan Nai Yahan	1.70	0.04	179.65	181.39

The communication time (or response time) is the time calculated from the query issuance of a peer to the download of *BF* and *SF* to reconstruction of the file. *BF* is downloaded in a centralized manner between a peer, *M* and *MO*, whereas for distribution of *SF*, FPSUM-HE incorporates the APFS protocol (cf. Section 3.5.2.1) proposed by Scarlata et al. (2001), in which peers construct an anonymous path with tail nodes using onion routing. In APFS, peers need one onion path, one TCP link to deliver the response between tail nodes, and two onion paths to send the response anonymously. In FPSUM-HE, two-phase authentication between two peers,

the numbers of used onion paths and TCP links are twelve and six to follow APFS, respectively.

The response time for *BF* distribution includes the time taken to transfer already created *BF* from *M* to a buyer. Similarly, the response time for the distribution of *SF* is evaluated by considering the two-party AKE protocol between a receiving peer and a providing peer and the time taken for the complete transfer of *SF*. The response time also includes file re-construction time at the user end. Table 4.9 summarizes the response time for the audio file "LoopyMusic".

TABLE 4.9: Communication time of an audio file

File Name	Communication Time (secs)				
	***BF* Delivery Time**	***SF* Delivery Time**	**File Reconstruction**	**Total Distribution Time**	**Direct Delivery Time**
LoopyMusic	8.01	10.00	3.89	21.90	7.00

The last column of Table 4.8 shows the delivery time of a direct file transfer between *M* and a buyer without considering security, privacy and accountability properties. The direct delivery time is calculated as a time taken to download "LoopyMusic" at a bit rate of 1.5 Mbps. It can be seen, from the table, that the total distribution time of FPSUM-HE is comparatively higher than the direct transfer time. This larger value of distribution time is due to the anonymous paths construction, authentication and encryption. Moreover, the total response time presented in 4th column of Table 5.9 represents the addition of an individual time of each process (*BF* and *SF* distribution and reconstruction). Since the audio file is divided into two parts: *BF* and *SF*, the *BF* and *SF* distribution protocols can be initiated simultaneously at a request of a peer to *SP* without interfering with each other. The parallel execution of *BF* and *SF* distribution protocols could result in reduction of the total distribution time of *BF* and *SF* from 8.01 + 10.00 = 18.01 seconds to 10 seconds. The reduced time (10 seconds) is slightly more than the direct delivery time (7 seconds) which in fact does not incorporate security and privacy properties. Hence, in achieving privacy and security in P2P systems, there is always a trade-off between anonymity, security and efficiency.

However, the concurrent execution of the protocols depends on the bit rate available at the peer's end. It might be possible that the parallel execution of the protocols require higher bit rate than available at the peer's end. For example, for "LoopyMusic" distribution, the parallel execution of *BF* and *SF* protocols requires a total bit rate of 1.94 Mbps at a peer's end. It could be a problem for the peers with a downloading bit rate limited to 1.94 Mbps or less. However, with constant advancements in the Internet and its related technology and the increased worldwide demand for rapid, low-latency and high-volume communication of information to homes and

businesses, nowadays the bit rates offered to home users by the Internet service providers typically ranges from 512 kbps to 10 Mbps in the direction to the downstream. The need for faster speed has changed the options available to consumers in terms of how fast the connections can be made. Thus, with the availability of increased bandwidth capacities, the parallel execution of the protocols can be easily performed.

Cryptographic costs

Cryptographic algorithms are applied in FPSUM-HE to ensure the desired level of security, privacy and accountability. The cryptographic algorithms are implemented in Java. AES-128, public-key encryption/decryption and AKE based on Diffie-Hellman are used in different phases of FPSUM-HE. Table 4.10 shows the CPU execution time of each cryptographic block for achieving the desired security for the audio file "LoopyMusic". As shown in Table 4.10, the costs of public-key cryptography used in encryption/decryption of *BF* and certificates generation, one time two-party AKE between two peers, and the AES-128 encryption/decryption of *SF* are 5.73, 9.62 and 1.89 seconds, respectively. It is evident from the table that the anonymous paths construction and authentication through these paths is the most expensive cryptographic operation. However, in achieving anonymity in P2P systems, there is always a cryptographic overhead. This overhead is due to encryptions and decryptions, insertion of fake traffic, and increasing the routing path to provide anonymity between two communicating users. However, the cost of authentication is small if compared to the cost of the systems implemented with onion routing based on asymmetric encryption. In FPSUM-HE, this cost is comparatively low due to use of symmetric encryption in onion routing.

TABLE 4.10: Cryptographic overhead of an audio file in FPSUM-HE

Cryptographic Algorithms	**Time (secs)**
Public-key cryptography	5.73
AES Encryption/Decryption	1.89
Anonymous Key Exchange	9.62
Total	17.24

4.6.2.2 Analysis of Video Fingerprinting

In this section, the results of three video files, namely, "Traffic", "Dragon" and "Breaking Bad" are presented in terms of imperceptibility, robustness, computational, communicational and cryptographic costs. The details of the three video files are presented in Table 4.11. The

simulation parameters are same as those in Table 4.2. However, in experiments for video file partitioning algorithm of "Traffic", level-3 DWT decomposition with *db*4 filter is used, and for "Dragon" and "Breaking Bad" video files, level-4 DWT decomposition with *db*4 filter is used. The levels of the DWT decomposition are selected to provide a good trade-off between robustness, capacity and imperceptibility.

TABLE 4.11: Details of video files

Details	**Traffic**	**Dragon**	**Breaking Bad**
Time Length (min:secs)	00:10	23:00	50:00
File Size (MB)	0.19	51.10	305.00
Format	AVI	AVI	MP4
Resolution (pixels)	120×160	320×240	720×406
Total Frames	120	32,975	67,817
Key Frames	15	2,228	2,649
Base File Size (MB)	0.08	9.21	11.80
Supplementary File Size (MB)	0.18	69.40	216.00

Transparency

The quality of video files are determined by the Peak Signal-to-Noise Ratio (PSNR) of the fingerprinted video. The PSNR provides a reliable indication of the variation of subjective video quality in decibels. To calculate the PSNR, first the Mean Square Error (MSE) between the original and the fingerprinted frame is computed as follows:

$$\text{MSE} = \frac{1}{H \cdot W} \sum_{i=1}^{H} \sum_{j=1}^{W} [X(i, j) - Y(i, j)]^2,$$

where, H, W are the size of the frame of a video, and $X(i,j)$, $Y(i,j)$ are the pixel values at location (i, j) of the original and fingerprinted frames. Then, the PSNR is defined as follows:

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right).$$

where, MAX is the maximum pixel value of the image. In FPSUM-HE, the pixels of all three video files have 8 bits per sample, thus the value of MAX in the above equation is equal to 255.

Since the inter and intra frames of all three video files are in RGB format, the equation of MSE, which is specified for only monochrome images, is divided by a factor of three for color images. Typical PSNR values for the fingerprinted video are between 30 and 50 dB, where higher values of PSNR indicate more imperceptibility of the fingerprinting scheme. The PSNR values of a video file are obtained by using *MSU Video Quality Measurement Tool* (2011) (VQMT). VQMT is an application for objective video quality assessment that can calculate the PSNR for all YUV and RGB components. Table 4.12 presents the imperceptibility results as PSNR of three fingerprinted video files.

Similar to audio fingerprinting, the imperceptibility of a video file is evaluated by performing the embedding experiment with 20 different collusion-resistant fingerprints. In the selected approximation coefficients of a video file, 20 different fingerprints are embedded using the embedding algorithm of Prins et al. (2007), thus producing 20 fingerprinted copies of a video file. The PSNR values of the fingerprinted video files varies, e.g. the PSNR values of "Traffic" vary in the range [40.00, 46.50] db. Similarly, for "Dragon" and "Breaking Bad" video files, the computed PSNR values are in the range [39.00, 43.50] and [37.50, 42.55] db, respectively. This variation in the PSNR values depends on the embedded fingerprint. From Table 4.12, it can be seen, that on average, the PSNR values are above 40.00 dB, which confirms that the fingerprinted video files are perceptually similar to the original video files.

TABLE 4.12: PSNR of video files

Video Files	PSNR in dB
Traffic	44.00
Dragon	42.00
Breaking Bad	41.00

Robustness against signal processing attacks

The signal processing attacks mentioned in Section 4.4.4.1 are performed on a video file "Traffic" to assess the robustness of the fingerprint. The BER and NC are used to evaluate the robustness between the original fingerprint and the extracted fingerprint. BER values close to zero indicate robustness against signal processing attacks. In the case of NC, if NC is close to 1, then the similarity between an original fingerprint f_i and an extracted fingerprint f_i' is very high. If NC is close to 0, then the similarity between f_i and f_i' is very low.

TABLE 4.13: Robustness of a video file against signal processing attacks

Attacks	Parameters	BER	NC	Traceability
Median Filter	$[3 \times 3]$	0.09	0.912	Yes
Re-sizing	320 – 640 – 320 pixels	0.06	0.972	Yes
H.264 Compression	768 kbps	0.09	0.912	Yes
AWGN	20 dB	0.14	0.856	Yes

The results in Table 4.13 show that the selected embedding algorithm (Prins et al., 2007) provides convenient performance against common signal processing attacks. The algorithm has good NC and BER values against various attacks for "Traffic". The minimum BER value is 6% and the maximum BER value is 14% against different attacks. The NC values are in the range 0.856 – 0.972, thus indicating close similarity between the original and retrieved fingerprints. Moreover, the fingerprint of a buyer is traceable against these common signal processing attacks. Thus, the results presented in Table 4.13 indicate that the fingerprint embedding algorithm satisfies the fingerprint's robustness requirement.

Computation and communication time

The time taken to generate the collusion-resistant fingerprint, and create *BF* and *SF*, is considered as a computation time. For *BF* and *SF* generation, the implementation of the file partition protocol in Section 4.5.1.2 excluding the key frames extraction process, contributes to the total computation time. Table 4.14 shows the CPU time for the three video files.

Table 4.15 presents the distribution of the *BF* generation's CPU time. In the video file

TABLE 4.14: Computation time of a video file

File Name	CPU Time (secs)			
	Fingerprint generation	*BF* generation	*SF* generation	Total Time
Traffic	6.01	10.77	7.22	24.00
Dragon	6.01	68.22	24.16	98.40
Breaking Bad	6.01	70.04	36.15	112.20

partitioning algorithm (cf. Section 4.5.1.2), the Canny-edge detection technique is applied only once to the video content to obtain the key frames. Similarly, the RGB conversion to $\mathcal{YUV}$ format and the DWT on the $\mathcal{Y}$ components of the key frames are applied once to obtain the approximation and detail coefficients. M stores the key frames, the inter frames (P and B), the approximation and detail coefficients of each video file. By doing so, M is able to avoid the costs of performing the Canny-edge detection technique, converting the RGB format frames to

$\mathcal{YUV}$ format, applying the DWT on the key frames and creating a *SF* every time a video file is requested by a buyer. Similar to the audio file partitioning process, the embedding part of the *BF* generation process includes the time taken to perform a quantization and encryption on the approximation coefficients of the key frames. For the "Traffic" video file, only two key frames are selected for embedding the collusion-resistant fingerprinting, whereas for the "Dragon" and "Breaking Bad" video files, the key frames are selected after a duration of 40 seconds from the obtained key frames. Thus, when a buyer requests for a particular video from the merchant *M*, the only computational process that a merchant needs to perform is the embedding process.

TABLE 4.15: Details of a computation time of a video file

File Name	CPU Time (secs)		
	Canny-Edge Detection	DWT	Embedding
Traffic	6.11	1.78	8.99
Dragon	720.00	24.50	43.72
Breaking Bad	1105.50	25.70	44.34

The communication time (or the response time) is the same time as described for an audio file in Section 4.6.2.1. The response time for *BF* distribution includes the time taken to transfer *BF* from *M* to a buyer. Similarly, the response time for the distribution of *SF* is evaluated by considering the two-party AKE protocol between a receiving peer and a providing peer, and the time taken for the complete transfer of *SF*. The response time also includes file reconstruction time at the user end. Table 4.16 summarizes the response time for the video file "Traffic".

TABLE 4.16: Communication time of a video file

File Name	Communication Time (secs)				
	BF Delivery Time	*SF* Delivery Time	File Reconstruction	Total Distribution Time	Direct Delivery Time
Traffic	1.01	9.88	7.02	17.91	3.00
Breaking Bad	184.00	657.29	595.05	1436.34	1560.00

The last column of Table 4.16 shows the execution time of a direct file transfer between *M* and a buyer without considering the security, privacy and accountability properties. The direct delivery time is calculated as a time taken to download "Traffic" at a bit rate of 1.5 Mbps. It is evident from the table that the *BF* distribution is short as compared to the direct transfer time. However, the total distribution time of a video file in FPSUM-HE is comparatively higher than the direct transfer time. This large value of distribution time is due to the anonymous paths construction, authentication and encryption. However, with an increase in the original file size, the

direct file transfer time also increases. For example, in case of the "Breaking Bad" video file, the size of *BF* is 11.80 MB, and the original size is 305 MB. It can be seen, in the last row of the Table 4.16, that the time to download *BF*, *SF* and an original file are 184, 657.29 and 1560 seconds, respectively. Thus, if the distribution time of both *BF* and *SF* are combined, it can been seen that the added time 841.29 seconds (184 + 657.29) is smaller than the direct download time 1560 seconds. Thus, it can be said that FPSUM-HE is suitable for a large-sized multimedia files.

Similar to the audio file distribution, the *BF* and *SF* protocols can be initiated simultaneously at a request of a peer to *SP* without interfering with each other. For example, in case of "Breaking Bad" video file, the parallel execution of *BF* and *SF* distribution protocols could result in reduction of the total distribution time of *BF* and *SF* from 841.29 seconds to 657.29 seconds. The reduced time (657.29 seconds) is two times smaller than the direct delivery time (1560 seconds). However, the concurrent execution of the protocols depends on the bit rate available at the peer's end. For example, in "Breaking Bad" video file, the parallel execution of *BF* and *SF* protocols require a total bit rate of 3.14 Mbps at a peer's end. It could be a problem for the peers with a downloading bit rate limited to 3.14 Mbps or less. In the past, the majority of the home users used Internet packages with 500 Kbps (downstream) to 250 Kbps (upstream), However, with the advancement in technology and market expansion, the Internet service providers nowadays offer faster services with typical bit rates of up to 10 Mbps downstream. Thus, with the availability of higher bit rates, it is possible to carry out the parallel execution of the protocols easily.

Cryptographic costs

AES-128 and 1024-bit public-key cryptography are employed in FPSUM-HE. Table 4.17 shows the CPU execution time of each cryptographic block for achieving the desired security for the video file "Traffic". The anonymous authentication process based on ZKPI, the asymmetric encryption/decryption of *BF* and the symmetric encryption/decryption of *SF* are evaluated to obtain cryptographic overhead for "Traffic". As shown in Table 4.17, the public-key cryptography used in encryption/decryption of *BF* and certificates generation, one time two-party AKE between two peers, and the AES-128 encryption/decryption of *SF* are 8.8, 9.62 and 0.11 seconds, respectively. It is evident from the table that the anonymous paths construction and the authentication through these paths is the most expensive cryptographic operation. The reason

for this high cost cryptographic operation is a need of accountability and anonymity in FPSUM-HE. As discussed in audio analysis (cf. section 4.5.1.5), anonymity is achieved at an additional cost.

TABLE 4.17: Cryptographic overhead of a video file in FPSUM-HE

Cryptographic Algorithms	**Time (secs)**
Public-key cryptography	8.80
Anonymous Key Exchange	9.62
AES Encryption/Decryption	0.11
Total	18.53

4.7 Conclusions

In this chapter, a framework FPSUM-HE is presented that enables the digital media producers to distribute their products efficiently to end users without worrying about illegal usage and distribution of their products, and the end users to avail the facilities of P2P system without a fear of a privacy breach.

In the proposed framework, the multimedia content is partitioned into a small-sized base file and a large-sized supplementary file. The base file is dispensed by the merchant on payment from the buyer and the supplementary file is distributed through the P2P network. Thus, the scheme lessens the computational cost of the merchant by only sending the small-sized base file and using the P2P network to support the majority of the file transfer process. For generation and distribution of the base file, an asymmetric fingerprinting protocol is performed between the merchant and the buyer in the presence of a trusted monitor. A robust, blind and imperceptible watermarking scheme is used to embed a collusion resistant digital fingerprint into the multimedia content. In the event that the merchant detects an unauthorized distribution of the content, he/she extracts the fingerprint from the pirated copy and gives the pirated code to the monitor. The monitor runs the tracing algorithm on the fingerprint to identify the pirate. The user's privacy is well-protected until there is a need to trace the identity of a user who distributes unauthorized copies of the copyright content. Even in case of arbitration, the co-operation from the buyer is not required. The data is well-protected from unauthorized reads and modifications during each data transfer phase in FPSUM-HE. The security and performance analysis demonstrate that FPSUM-HE provides the guaranteed security and privacy properties discussed in Chapter 3. In the next chapter, the computational and communicational efficiency aspects of the proposed framework are further improved with a different fingerprinting strategy.

Chapter 5

Framework for preserving Privacy and Security of User and Merchant with Proxy-based Distribution

This chapter presents a Framework for preserving Privacy and Security of User and Merchant with Proxy-based Distribution (FPSUM-PD) that is proposed to improve the performance of FPSUM-HE. In contrast to FPSUM-HE, which uses homomorphic encryption to embed a fingerprint into the encrypted small-sized multimedia base file and inflicts computational and communication burden on the merchant, FPSUM-PD lessens this cost by sending a small-sized base file composed of pre-computed fingerprinted information bits through a set of proxy peers to the buyers.

5.1 Introduction

In Chapter 4, FPSUM-HE, a P2P content distribution framework for preserving privacy and security of the user and the merchant based on homomorphic encryption, is presented. In the framework, some Discrete wavelet transform (DWT) low-frequency (approximation) coefficients are selected according to a secret key *sk* for embedding an encrypted fingerprint to prevent data expansion due to homomorphic encryption. The remaining approximation coefficients are encrypted block-by-block using public-key cryptography. Although the selective public-key encryption of the multimedia content results in lesser data expansion, yet it imposes

computational burden on a merchant and an increased complexity in file reconstruction at the buyer's end.

In this chapter, a solution is proposed to achieve an efficient asymmetric fingerprinting scheme in which public-key encryption is not applied to the multimedia content rather it used only for encrypting short-binary strings and data signing. In addition, the communication bandwidth and computation power of the merchant are further reduced by transmitting a small but significant part of the multimedia content in a semi-centralized way and using a network of peer buyers to distribute the remaining large portion of the content. In the proposed system, Framework for preserving Privacy and Security of User and Merchant using Proxy-based Distribution (FPSUM-PD), the multimedia file is partitioned by the merchant into a base and a supplementary file. The base file contains the most important information and, without it, the supplementary file is unusable. A merchant forms a base file by using a pre-computation-based secure embedding mechanism in which the DWT low-frequency coefficients are embedded in parallel with all 1s and all 0s bits. An asymmetric fingerprinting protocol based on collusion-resistant codes and a secure embedding scheme is performed between a merchant, a buyer and a set of P2P proxies in the presence of a trusted third party (monitor), in such a way that the merchant does not know the fingerprint and the fingerprinted content. In addition, the proxies are unable to frame honest buyers by combining their assigned permuted fingerprint bits, while the buyer receives the fingerprinted content with his/her unique identity. The collusion-resistant codeword is generated by a monitor and is decomposed into fixed length blocks. The monitor permutes the bits of these blocks using different permutation keys generated by the buyer, and then assigns the permuted segments of the fingerprint to a set of proxy peers. FPSUM-PD also enables buyers to obtain digital contents anonymously by using dynamic pseudonyms based on a one-way hash function instead of their real IDs, but this anonymity can be revoked as soon as he/she is found guilty for copyright violation. To ensure anonymous communication between buyers, onion routing is used for an anonymous data transfer. A symmetric-key encryption is performed on the supplementary file to prevent the onion routers (or middle nodes) from correlating the incoming and outgoing content. The implementation with a software solution of the proposed system is discussed with formal security and detailed performance analysis.

The work described in this chapter has been published as a conference paper (Qureshi et al., 2014) and has been submitted to an international journal (Qureshi, Megías, & Rifà-Pous, n.d.).

This chapter is organized as follows. Section 5.2 presents an overview of a framework which describes an environment and the design fundamentals of the framework. In Section 5.3,

the architecture of FPSUM-PD is presented, and describe the protocols of this framework designed to address the security and privacy concerns of the merchant and the user, respectively. In Section 5.4, formal security analysis of the framework's protocols through a number of attack scenarios are presented. In this section, the performance and efficiency analysis of the framework are also presented. Finally, a conclusion is provided in Section 5.5.

5.2 Overview of the Framework

FPSUM-PD consists of two components, namely, environment and design fundamentals, which provide general guidelines on its architecture. The environment part of FPSUM-PD is similar to FPSUM-HE, thus only a brief review of the elements of the environment component is provided in Section 5.2.1. The design fundamentals are described in Section 5.2.2. Fig. 5.1 illustrates the environment and design fundamentals of FPSUM-PD.

5.2.1 Environment

The environment aims to identify the elements that are available for constructing FPSUM-PD. It consists of the following components: P2P network, trust infrastructure and building blocks.

- **P2P Network:** In FPSUM-PD, a hybrid P2P network is opted as a platform for content distribution. Hybrid P2P consumes less network resources and is more scalable than centralized and pure P2P systems. Moreover, the idea of centralized and P2P distribution can easily be achieved by using a hybrid P2P system, since multiple coordinators, called super peers, can easily manage both base file and supplementary file distribution.

- **Trust Infrastructure:** FPSUM-PD involves the following trust infrastructures: Public key support and Trusted Third Parties (TTPs) with specific services. In FPSUM-PD, the existence of a public key infrastructure (PKI) is assumed for providing a public/private key pair for each entity. Also, one offline external *CA* and one online internal *CA* are considered in FPSUM-PD for providing revocable anonymity to the buyers of the system. The offline *CA* is only responsible for validating the real identity of a buyer by providing a signed public-key certificate to the buyer. On the other hand, the internal *CA* validates the anonymous key pairs used by the authenticated buyer during the anonymous content distribution protocol. Two trusted third parties namely, the monitor and the judge, are

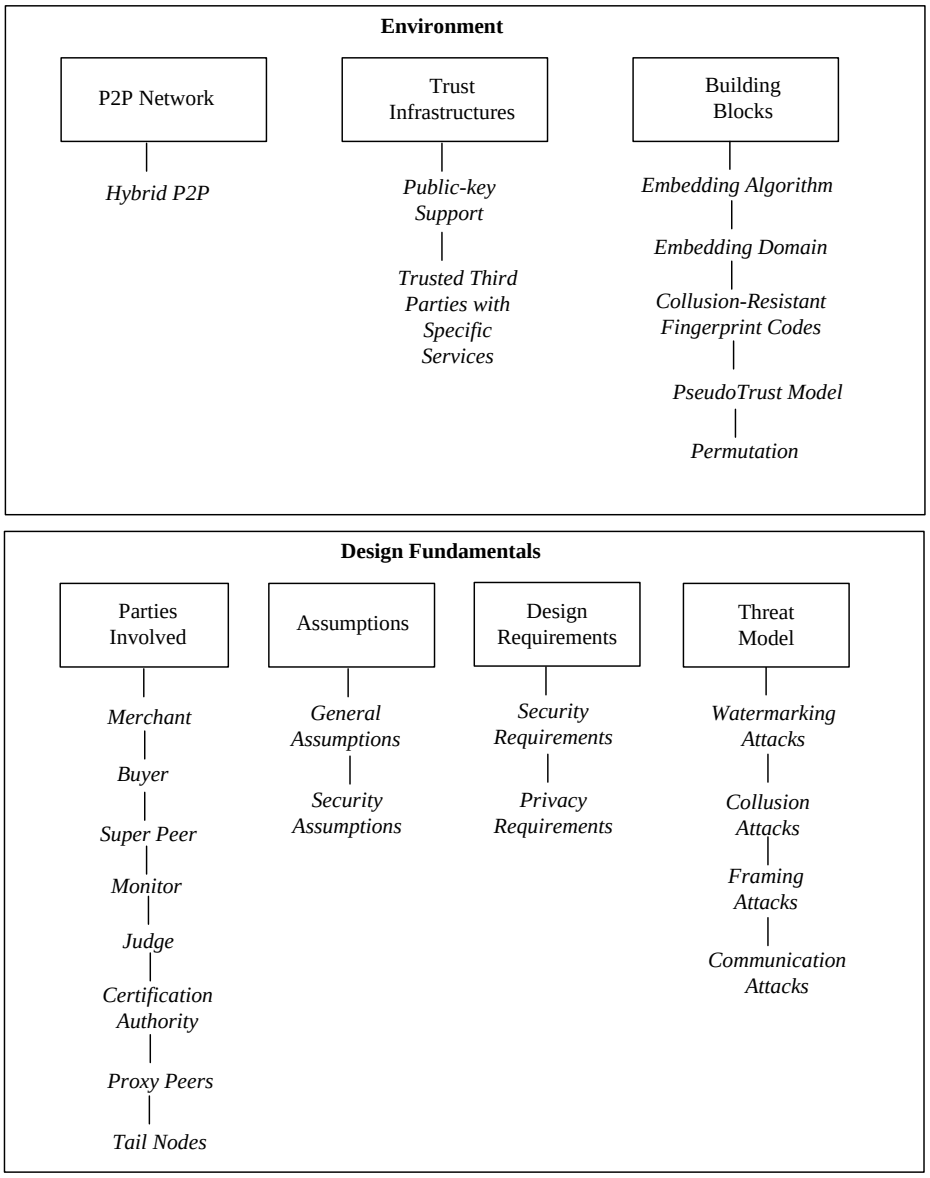


FIGURE 5.1: An overview of FPSUM-PD

used in FPSUM-PD to provide buyer frameproofness in the content distribution protocol (cf. Section 5.3.1.3) and privacy preservation of a buyer in arbitration and identification protocol (cf. Section 5.3.1.5).

- **Building Blocks:** In this section, a brief overview of the building blocks (embedding domain and algorithm, collusion-resistant fingerprinting codes, PseudoTrust model and permutation of FPSUM-PD) is presented.

  1. **Embedding Domain:** The DWT is used in FPSUM-PD to embed the collusion-resistant fingerprint into a multimedia content. In the signal processing research

area, the wavelet transform has gained widespread acceptance in recent years. Because of their inherent multi-resolution nature, wavelet-coding schemes are especially suitable for applications where scalability and tolerable degradation are important. The DWT of a signal results into approximation coefficients indicating the low frequency components of the signal and detail coefficients representing the high frequency components. Since the low frequency coefficients can effectively resist various signal processing attacks, the fingerprint bits are typically embedded into the approximation coefficients of the signal after the DWT. Moreover, the original signal can be reconstructed from the approximation and detail coefficients, which is called the inverse discrete wavelet transform (IDWT).

2. **Embedding Algorithm:** An embedding algorithm is used to insert a fingerprint into different copies of the same content. The embedding schemes of FPSUM-PD are categorized into two groups: audio and video fingerprinting.

   a). **Audio Embedding Algorithm:** In order to embed a collusion-resistant fingerprint into an audio signal, a blind and adaptive audio watermarking algorithm (Xinkai et al., 2013) based on vector norm is used. In this scheme, a watermark is embedded into a vector norm of the segmented approximation components after DWT of original audio signal through Quantization Indexed Modulation (QIM) with adaptive quantization steps. The adaptive quantization steps are determined by the signal-to-noise ratio (SNR). Moreover, a detailed method has been designed in (Xinkai et al., 2013) to search the suitable quantization step parameters. In addition, the watermark can be extracted without the help of the original audio signal, thus implying a blind extraction. The details of the embedding algorithm for an audio signal are provided in Protocol 1.

   b). **Video Embedding Algorithm:** An oblivious image watermarking based on the DWT and Quantization Index Modulus Modulation (QIMM) proposed by Leelavathy et al. (2011) is employed to embed a fingerprint into the key frames of the video file. The embedding quantization step size Δ_{QIM} of QIM is almost equal to two times of Δ_{QIMM} of QIMM, i.e. QIMM can achieve the same mean square error with half of the quantization step size in QIM. Therefore, a better robustness and peak signal-to-noise ratio (PSNR) is obtained in QIMM than compared to QIM with a constant quantization step size. In the embedding

algorithm of Leelavathy et al. (2011), the low-frequency wavelet coefficients of an image are quantized using QIMM and the coefficients are then modified according to the binary watermark. Also, the watermark can be extracted from the watermarked video signal without the help of the original video signal. The details of the video embedding algorithm can be found in Protocol 2.

3. **Collusion-Resistant Fingerprinting Codes:** Nuida et al.'s c_0-secure codes (Nuida et al., 2007) are used in FPSUM-PD for the generation of the collusion-resistant code. Nuida et al. proposed a discrete distribution of state-of-the-art collusion-resistant Tardos codes with a δ-marking assumption (the number of undetectable bits that are either erased or flipped is bounded by δ-fraction of the total code length m) to reduce the code length m and the required memory amount without degrading the traceability. The code length m is evaluated under the binary symmetric channel with a certain error rate. The tracing algorithm of Nuida et al. outputs one user with the highest accusation score. The details of Nuida et al.'s fingerprint generation and traitor-tracing algorithms can be found in Sections 4.5.1.1 and 4.5.1.5.

4. **PseudoTrust Model:** A PseudoTrust model (cf. Section 3.4.3) proposed by Lu et al. (2007) based on a zero-knowledge proof-of-identity is used in FPSUM-PD to provide revocable anonymity and unlinkability security properties (cf. Section 2.2.4.2). The PseudoTrust model enables pseudonym-based trust management so that the real identities of peers are protected during the authentication. In addition, the communication between two peers is anonymized using onion routing within FPSUM-PD. In a PseudoTrust model, the pseudo-identities are generated by the peers without any trusted third party, which leads to an accountability problem in the system. Thus, to add accountability to FPSUM-PD, an internal certificate authority (CA_R) is incorporated in the PseudoTrust model. Each peer is authenticated by CA_R before he/she joins the network. Hence, each peer has a private key, a public key and a public key certificate signed by CA_R. The pseudo-identities and certificates are used by the peers for anonymous communication within the P2P system. The details of generation of pseudo-identities and anonymous authentication process are given in Sections 4.3.3.5 and 4.5.1.4.

5. **Permutation:** The security requirement presented in Section 2.2.4.2, namely buyer

frameproofness, requires the fingerprinting scheme to be able to provide framing resistance to a buyer from any malicious attack. In FPSUM-HE, buyer frameproofness was provided through a homomorphic cryptosystem. The homomorphic encryption of multimedia content results in increased complexity and computational costs at a buyer's and a merchant's end, respectively. In FPSUM-PD, buyer's security and non-repudiation properties are provided by using the concept of the permutation. The permuted fingerprint generated by the trusted monitor (*MO*) is permuted using different permutation keys and is then assigned to a set of proxy peers Pr_j in such a way that the merchant cannot predict about the fingerprint and the fingerprinted content, and Pr_j are unable to frame honest buyers by combining their information bits.

Permutation is an ordered arrangement of a set of elements. Only sets with a finite number of elements can be considered for permutation. The number of possible permutations of a set of n elements is $n!$. For example, a permutation on a fingerprint f_i of 30 bits results into 30! possible arrangements of f_i. Fig. 5.2 illustrates the permutation concept of a fingerprint in FPSUM-PD. Fig. 5.2 shows a fingerprint f_i of 30 bits, and a random permutation key σ_j of 30 bits. σ_j is applied to f_i such that the bit position 1 of a permutation key corresponds to bit position 2 of a permuted fingerprint ($1 \rightarrow 2$), the second bit position corresponds to the bit position 9 of a permuted fingerprint ($2 \rightarrow 9$), and so on. On applying the inverse permutation key σ_j^{-1} to a permuted fingerprint, the original fingerprint f_i is obtained.

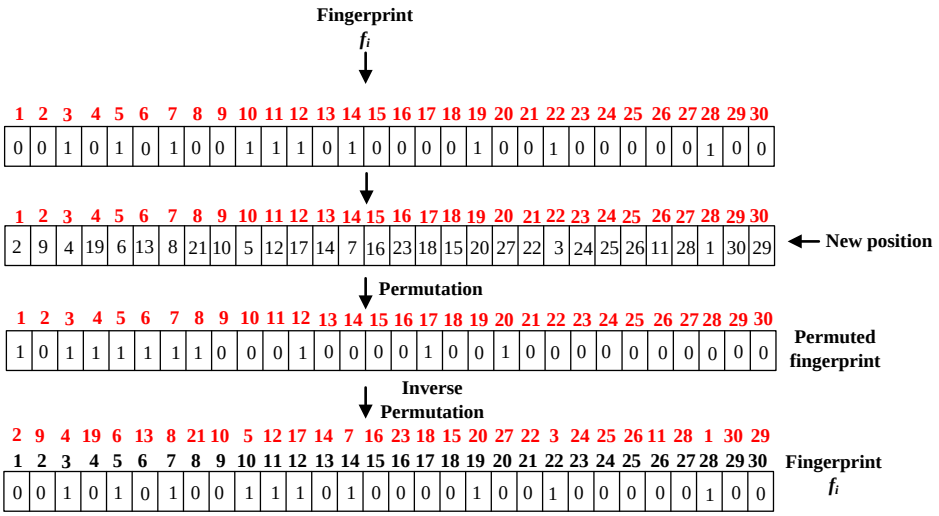


FIGURE 5.2: Permutation of a fingerprint in FPSUM-PD

5.2.2 Design Fundamentals

The design fundamentals aim to provide a proper definition of the objectives of FPSUM-PD. It consists of the parties involved, the assumptions, the design requirements and the threat model.

5.2.2.1 Parties Involved

FPSUM-PD involves eight entities and the function of each entity is defined as follows:

- A merchant (M) is an entity that distributes the copyright content to the buyers in the P2P system. It is involved in the generation and distribution of BF and SF, the traitor-tracing and the dispute resolution protocols.

- A buyer (or a peer B_i) is an entity that either plays a role of a data requester or provider. B_i is involved in the acquisition of BF from the merchant, the distribution of SF in FPSUM-PD and a dispute resolution if he/she is found guilty of copyright violation.

- A super peer (SP) acts as a coordinator for a small portion of the group of peers (buyers). SP facilitates B_i's acquisition of BF from M and SF from the peers present in FPSUM-PD.

- A certification authority (CA_R) is a trusted party that is responsible of issuing certificates to B_i for acquisition of BF from M and SF from other peers. The certificate is used to certify that the pseudonym is correctly registered to CA_R and CA_R knows about the real identity of B_i.

- A monitor (MO) functions as a trusted party which is responsible for the generation of collusion-resistant fingerprint codes. The existence of MO ensures that the generated fingerprints are not revealed to M and B_i. MO is also responsible for assigning segments of fingerprint codeword s_j to a set of proxy peers (Pr_j, for $j = 1, \ldots, n$) in such a way that proxy peers are unable to frame an honest B_i by colluding. It also keeps the record of the transactions made with the proxy peers and the buyers. In addition, MO also starts the traitor-tracing protocol in case of a piracy claim by M. In case of a dispute resolution between M, B_i, and a judge, MO provides the pseudonym of the guilty B_i to the judge.

- A proxy peer (Pr_j) is responsible for querying content of BF available at M's end with the pre-assigned bits of a fingerprint codeword f_i and transferring the retrieved content to the requesting B_i.

- A judge (J) is assumed to be a trusted party which resolves the disputes between M and B_i with the cooperation of MO and CA_R.
- A tail node T_A is a message transferring agent that manages anonymous communication on behalf of a peer A. Each peer within the P2P network has one such agent. The tail node forwards the query of a requesting peer to the providing peer through an anonymous path and returns the reply back to the requesting peer.

5.2.2.2 Assumptions

The architecture of FPSUM-PD is quite similar to FPSUM-HE's architecture. Thus, the general and security assumptions of FPSUM-PD closely relate to FPSUM-HE's assumptions. This section only describes the assumptions that are different from FPSUM-HE's architecture. The remaining assumptions can be found in Section 4.4.2.

General assumptions

In the following, the general assumptions related to the construction of FPSUM-PD are defined:

- There are six major players involved: merchant M, buyer/peer B_i, monitor MO, certification authority CA_R, proxy peers Pr_j, and judge J.
- In order to deliver BF from M to B_i, MO selects a fixed number (n) of proxy peers.
- The number of proxy peers n and the length of a fingerprint m are known constants of the system.
- Pr_j must follow each other in a sequential manner to transfer BF to B_i from M.
- In order to protect data privacy during BF exchange, MO must wait for some time τ until at least two buyers request for a content from M. This step is enforced on MO to ensure that M obtains no knowledge about which L-level DWT approximation coefficient is accessed and transferred to B_i.

Security assumptions

The security assumptions of FPSUM-PD are defined in this section.

- M and B_i do not trust each other but they both trust MO. Because of the anonymity of the embedding procedure, MO generates the collusion-secure fingerprints as this is the only party that is trusted by both M and B_i to generate a valid fingerprint. Also, in case of traitor-tracing process, it is expected that MO does not form a coalition with any other party to frame B_i.

- The permutation keys σ_j (for $j = 1, \ldots, n$) are generated by B_i to perform permutation of a fingerprint codeword to be assigned to the proxy peers (Pr_j). The purpose of generating σ_j is to ensure that a collusion of malicious Pr_j is unable to generate a valid fingerprint codeword or a fingerprinted content.

- Pr_j are not trusted and the content transferred through them is encrypted in such a way that only M and B_i have access to the clear-text.

- Public-key cryptography is restricted to the encryption of small-length binary strings such as symmetric session and permutation keys.

5.2.2.3 Design requirements

In this sub-section, the design requirements of FPSUM-PD are presented in terms of content and privacy protection.

Security requirements

The following are the security requirements of FPSUM-PD:

- **Traceability:** M should be able to trace and identify an illegal re-distributor in case of finding a pirated copy with the help of MO, J and CA_R.

- **Collusion resistance:** The scheme should be collusion resistant against a given number of colluders c_0 as specified by Nuida et al. (2007) codes.

- **Buyer frameproofness:** The possible collusion of Pr_j should be unable to frame an honest B_i. Also M should not be able to frame an honest B_i of illegal re-distribution.

- **Non-repudiation:** B_i accused of re-distributing an unauthorized copy should not be able to claim that the copy was created by M or a collusion of the proxies Pr_j.

- **Robustness and Transparency:** The embedded fingerprint should be imperceptible and robust against common signal processing attacks.
- **Extraction:** The extraction of a fingerprint should be blind.

Privacy requirements

In the following, the desired privacy properties of FPSUM-PD are presented:

- **Anonymity:** The identity of B_i should remain anonymous during transactions until he/she is proven to be guilty of copyright violation.
- **Unlinkability:** B_i's should not be linked to his/her activities such as purchasing, transferring of a file and so on.
- **Anonymous authentication:** The real identity of B_i should be protected during authentication process, thus enabling each B_i to verify the authenticity of each other anonymously.
- **Data privacy:** None of the tail nodes should know about the requesting buyer's and source provider buyer's identity or an item being exchanged. Thus, a supplementary file transfer between the requesting buyer and the providing buyer must be secure.
- **Buyer Privacy:** J, with the help of MO, should be able to resolve the disputes without involving B_i in the process.

5.2.2.4 Attack Model

This section highlights an attack model for FPSUM-PD related to the robustness of an embedding scheme, and B_i's privacy and security attacks from malicious entities.

Watermarking attacks

An attack succeeds in defeating a watermarking scheme if it impairs the fingerprint beyond acceptable limits while maintaining the perceptual quality of the attacked data. A fingerprint embedding scheme used for copyright protection must have a capability to survive attacks such as signal enhancement, geometrical operations and noise filtering. In case of audio, re-quantization, re-sampling, MPEG-1 layer 3 (MP3) compression, and Additive White Gaussian Noise (AWGN) attacks are considered for evaluating the robustness of the fingerprint embedding

scheme. Similarly, robustness of a fingerprint embedding algorithm for video data is evaluated under median filtering, re-sizing, H.264 compression and AWGN attacks.

Collusion attacks

The collusion attack from a group of malicious buyers (colluders), combining several copies with the same content but different fingerprints to try to remove the embedded fingerprints or frame honest buyers, is the major challenge to digital fingerprinting. If a digital fingerprint is not properly designed, a fingerprinting system might fail to detect the traces of any fingerprints under collusion attacks with only a few colluders. To ensure the reliable tracing of true traitors and avoid framing honest buyers, linear (averaging) and non-linear (maximum, minimum and median) collusion attacks are performed.

Privacy and security attacks on a buyer

The following types of attacks are aimed to de-anonymize B_i and accuse an innocent B_i of illegal re-distribution of the purchased content:

1. Different transactions carried out by B_i with the same pseudo-identity are linkable to one another and an attacker could infer some private information of B_i through data mining techniques.

2. A malicious entity may try to find two different but real identities such that the two identities have the same pseudo-identity. It might then use one of the two identities to impersonate B_i to obtain a fingerprinted copy of the content that would be linked to the impersonated B_i.

3. M and one or more Pr_j may collude to create a new fingerprinted content Y.

4. Pr_j may collude to create a new fingerprint f_i.

5. A possible collusion of B_i and all (or some of) Pr_j may try to obtain the complete (or partial) set of approximation coefficients and produce non-fingerprinted copies of the original content X.

Communication attacks

The following attacks allow attackers to exploit the communication between the two entities in FPSUM-PD:

1. ***Eavesdropping:*** A malicious proxy peer might eavesdrop on a communication between M, MO, and B_i to obtain a secret permutation key σ_j.

2. ***Leakage of the secret number r:*** The secret number r is a key generated by CA_R for sharing it with an authenticated B_i in the pseudo-identity generation step. However, if r is leaked, then any malicious node can use r to impersonate other buyers.

5.3 Model

This section describes the architecture of FPSUM-PD. Fig. 5.3 shows the structure of FPSUM-PD that contains six main entities: merchant, buyer, super peer, monitor, proxy peers and judge. These entities are involved in six key protocols: fingerprint generation, file partitioning into BF and SF, distribution of BF and SF, traitor tracing and dispute resolution of the system.

5.3.1 Protocols

The protocols of the proposed framework are reviewed in the following sections:

5.3.1.1 Generation of Collusion-resistant Fingerprint

The fingerprint f_i is generated by MO using the Nuida et al. (2007) codes algorithm. The fingerprint generation algorithm takes ε, N and c_0 as inputs, and outputs a collection $F = (f_1, \ldots, f_N)$ of binary codewords (f_i) of size m and a secret bias vector p. The length of the fingerprint is calculated as $m = (c_0^2 K \log(N/\varepsilon))$, where the value of K is 4.245. The details of the algorithm can be found in Algorithm 1 in Section 4.5.1.1.

5.3.1.2 File Partitioning

In this section, the partitioning of a multimedia file X into a small-sized BF and a large-sized SF is discussed. The proposed method employs the DWT to split a multimedia content into low-frequency (approximation coefficients) and high-frequency (detail coefficients) components. An approximation coefficient is then itself split into a second-level approximation and detail coefficients, and the process is repeated as many times as desired (levels of decomposition). The

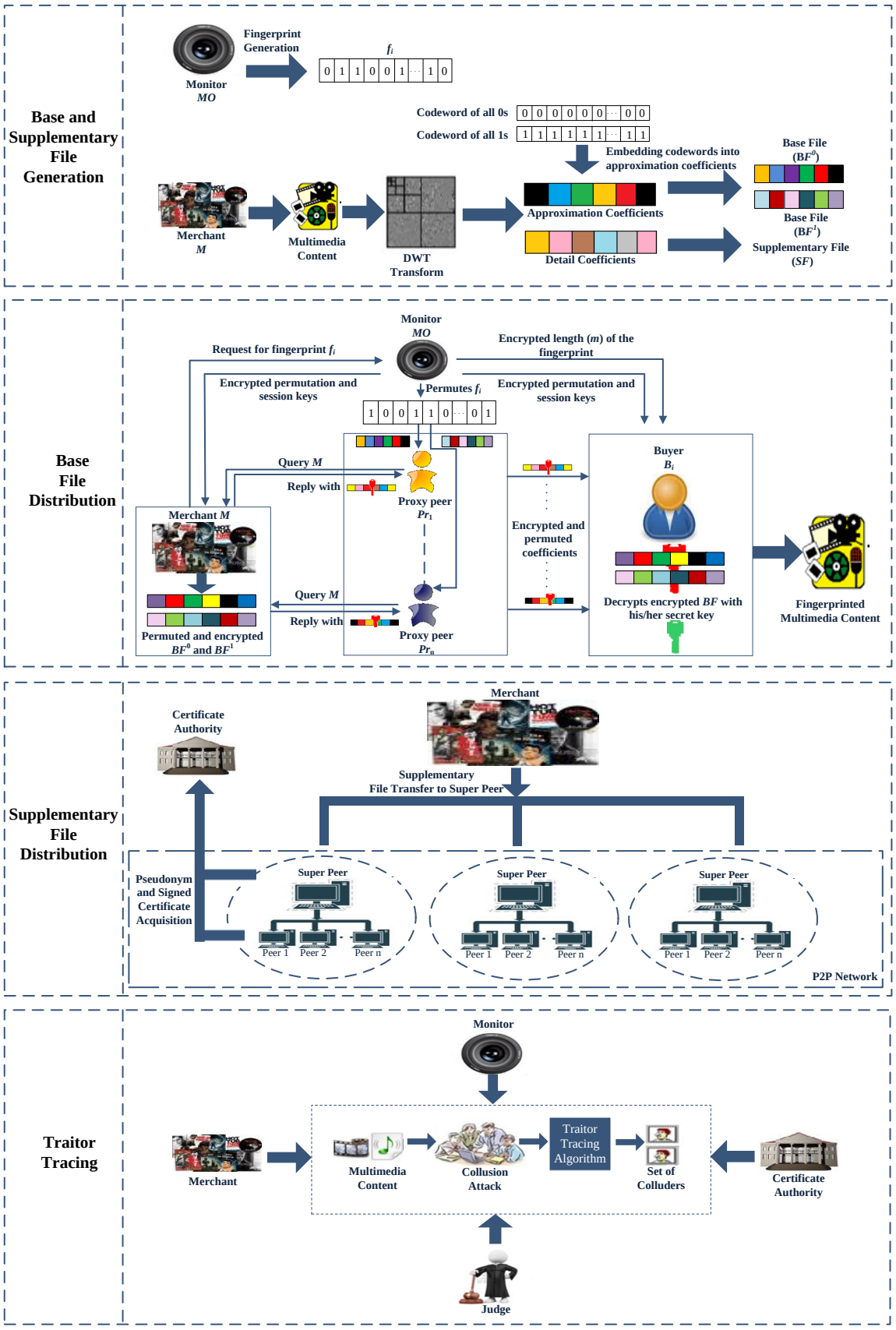


FIGURE 5.3: An overview of FPSUM-PD

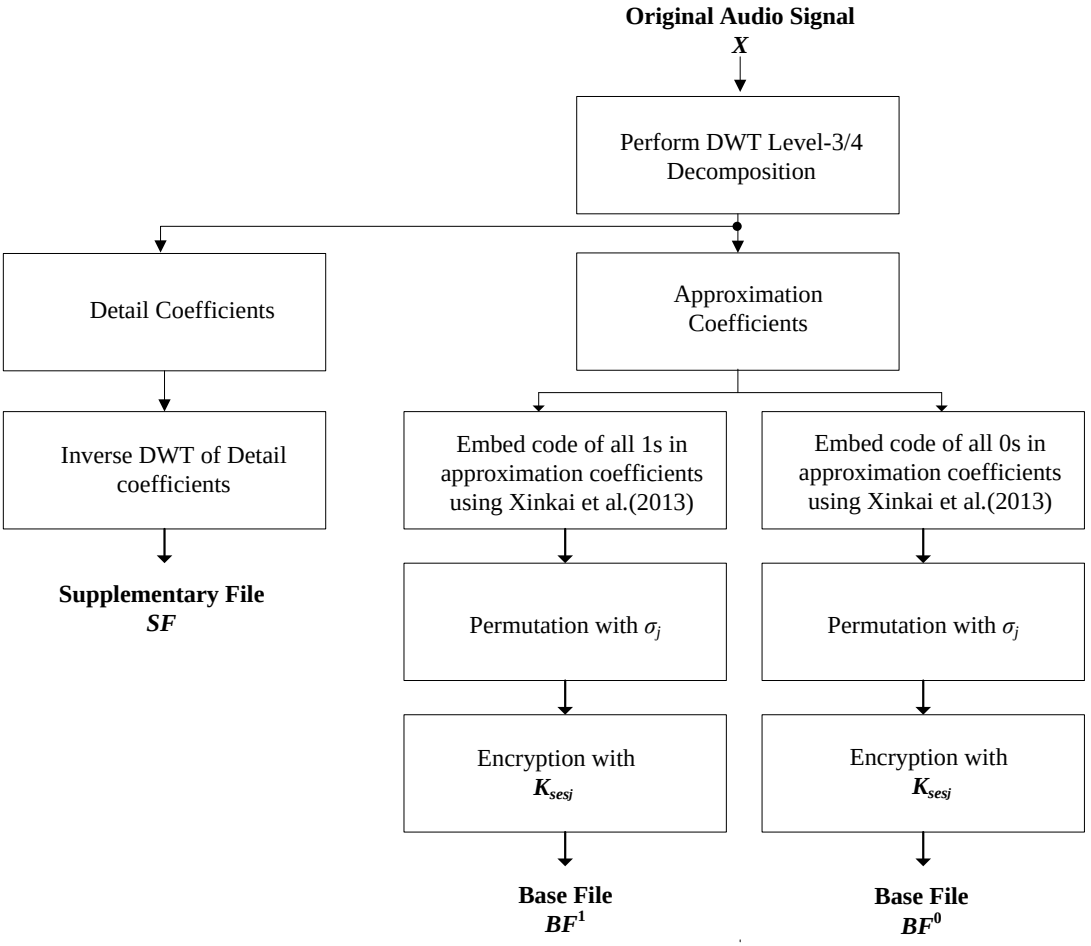


FIGURE 5.4: Audio file partitioning

approximation coefficients are used to form *BF* and the detail coefficients are used in *SF* creation. *BF* contains a collusion-resistant fingerprint f_i and is dispensed by the merchant through a set of proxies on payment from B_i, and *SF* is distributed through the P2P network. The asymmetric fingerprinting protocol is performed between M, B_i, and a set of proxy peers Pr_j, in the presence of *MO*, in such a way that M does not know f_i and the fingerprinted *BF*, while B_i receives *BF* with his/her unique identity.

In FPSUM-PD both audio and video multimedia files are considered. Therefore, the explanation of the partitioning method for each type of the content is required. The audio and video file partitioning algorithms are explained below.

- **Partitioning of an Audio File:** Fig. 5.4 illustrates an audio file partitioning process, whereas Algorithm 1 describes the audio file partitioning into *BF* and *SF*.

- **Partitioning of a Video File:** Fig. 5.5 illustrates a video file partitioning process, whereas Algorithm 2 describes the video file partitioning into *BF* and *SF*.

Algorithm 1 Steps that are executed for an audio file partitioning into *BF* and *SF*

1: The 4-level DWT is applied to an audio signal X with X_1 samples to split it into approximation and detail coefficients.

2: The level-4 approximation coefficients ($a_k = \{a_k | 1 \leq k \leq X_1/16\}$) are divided into non-overlapping frames $\mathcal{F}_\ell \lfloor X_1/(16 \cdot m \rfloor)$, where ℓ is the number of the total frames. The length $\mathcal{L}$ of each frame is equal to m, where m is the length of the fingerprint f_i.

3: The frames are then used to embed codewords of all 0s and all 1s using the audio watermarking algorithm proposed by Xinkai et al. (2013).

4: The first step in Xinkai et al. (2013)'s audio embedding algorithm is to calculate ω_ℓ by applying vector norm on each frame $\mathcal{F}_\ell$.

$$\omega_\ell = \|\mathcal{F}_\ell\| .$$

5: The adaptive quantization step Δ is computed according to the formula:

$$\Delta = \alpha_1 + \alpha_2 \sqrt{\frac{10^{-attack_{SNR}/10} \sum_{\varsigma=1}^{\mathcal{L}} \mathcal{F}_\ell(\varsigma)^2}{\mathcal{L}}}.$$

where, α_1, α_2 and $-attack_{SNR}$ are the constants used in audio embedding algorithm (Xinkai et al., 2013).

6: Compute $e = \lfloor \omega_\ell / \Delta \rfloor$ and $d = \omega_\ell \bmod \Delta$.

7: In case the codeword contains all 1s, ω_ℓ is modified as following:

$$\omega_{\ell 1} = \begin{cases} \omega_\ell + \Delta/2 - (\omega_\ell \bmod \Delta), & \text{if } e \bmod 2 = 1, \\ \omega_\ell + \Delta + \Delta/2 - (\omega_\ell \bmod \Delta), & \text{if } e \bmod 2 = 0 \text{ and } d > \Delta/2, \\ \omega_\ell + \Delta + \Delta/2 - (\omega_\ell \bmod \Delta), & \text{if } e = 0 \text{ and } d < \Delta/2, \\ \omega_\ell + \Delta + \Delta/2 - (\omega_\ell \bmod \Delta), & \text{if } e \bmod 2 = 0 \text{ and } d > \Delta/2 \text{ and } \& e \neq 0. \end{cases}$$

And in case the codeword contains all 0s, ω_ℓ is modified as follows:

$$\omega_{\ell 2} = \begin{cases} \omega_\ell + \Delta/2 - (\omega_\ell \bmod \Delta), & \text{if } e \bmod 2 = 0, \\ \omega_\ell + \Delta + \Delta/2 - (\omega_\ell \bmod \Delta), & \text{if } e \bmod 2 = 1 \text{ and } d > \Delta/2, \\ \omega_\ell + \Delta + \Delta/2 - (\omega_\ell \bmod \Delta), & \text{if } e \bmod 2 = 1 \text{ and } d < \Delta/2. \end{cases}$$

8: The modified frames $\mathcal{F}_\ell^1$ in case of all 1s codeword are obtained as follows:

$$\mathcal{F}_\ell^1 = \omega_{\ell 1} \cdot u_\ell^T .$$

where, $u_\ell^T = \mathcal{F}_\ell / \omega_\ell$. In case all 0s codeword is used, then the modified frames $\mathcal{F}_\ell^0$ are obtained as follows:

$$\mathcal{F}_\ell^0 = \omega_{\ell 2} \cdot u_\ell^T .$$

9: The approximation components are then reconstructed by combining all the modified frames $\mathcal{F}_\ell^1$ or $\mathcal{F}_\ell^0$. $\mathcal{F}_\ell^1$ and $\mathcal{F}_\ell^0$ are saved in a text format as BF^1 as BF^0, respectively.

10: BF^1 and BF^0 are permuted with the permutation key σ_j.

11: The permuted BF^1 and BF^0 are then encrypted with K_{ses_j}. Both permuted and encrypted BF^1 and BF^0 are saved in a block form in a text format.

12: An inverse 4-level DWT is performed on the detail coefficients to obtain *SF* in "wav" format. Other formats, such as binary and text, can also be used for the formation of *SF*.

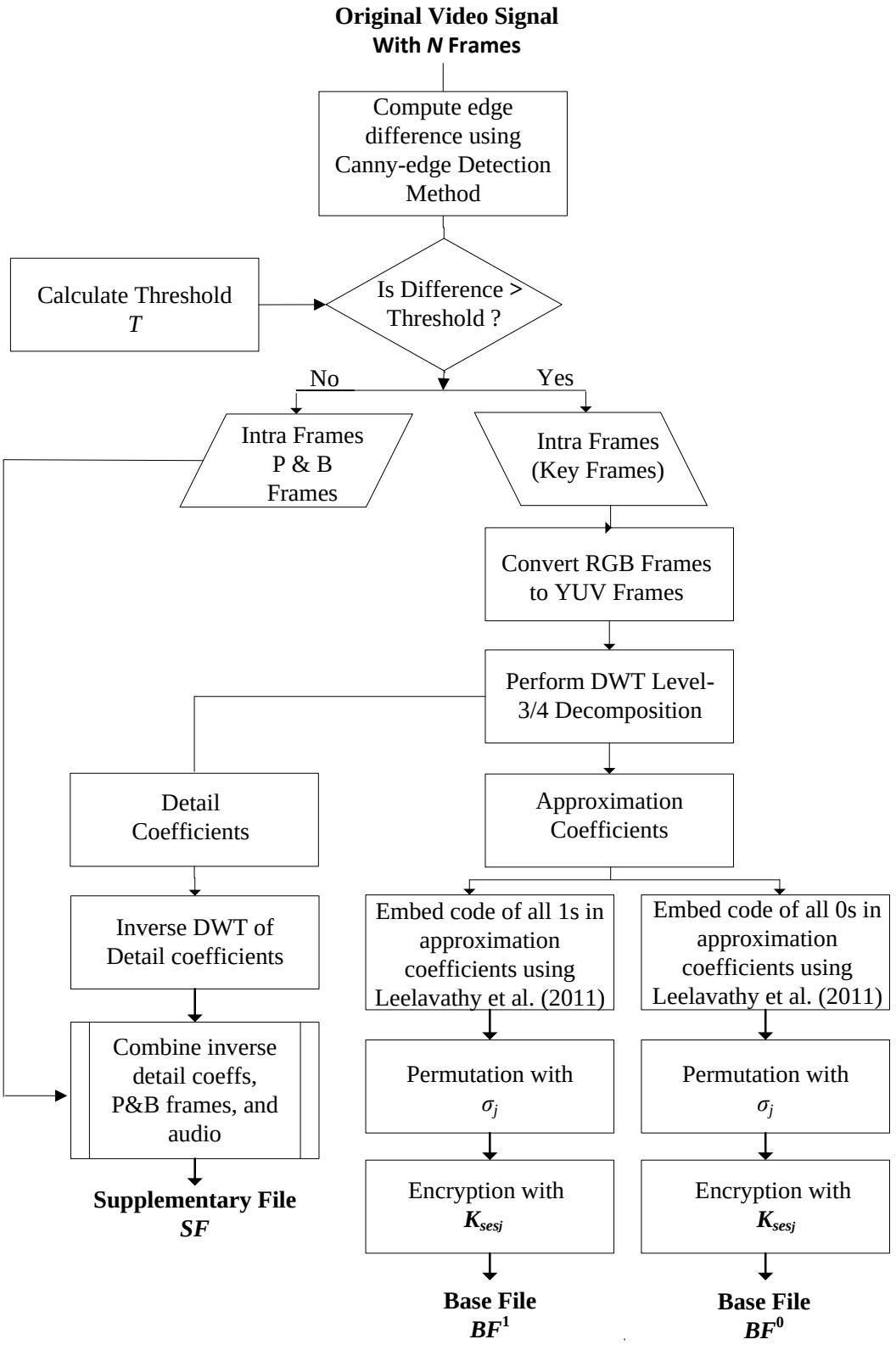


FIGURE 5.5: Video file partitioning

Algorithm 2 Steps that are executed for a video file partitioning into *BF* and *SF*

1: In order to divide a video file into *BF* and *SF*, first the intra-frames (key frames) from a video file are extracted, since the other frames (inter-frames) of the video do not contain important information. It is not meaningful to use both intra (I) and inter-frames (P & B). Thus, only the key frames which contain important information are used. For the detection of a key frame, the Canny-edge detection technique (Khurana & Chandak, 2013) is used. The details of the key frame extraction method can be found in Algorithm 2 (cf. Section 4.5.1.2).

2: The key frames obtained in Step 1 are converted from RGB format to $\mathcal{YUV}$ format.

3: For each key frame, a $\mathcal{Y}$ component is selected. A 4-level DWT is applied to $\mathcal{Y}$ to obtain the approximation a_k and detail coefficients. The codewords containing all 0s and all 1s are then embedded into approximation coefficients using the video watermarking algorithm proposed by Leelavathy et al. (2011).

4: The first step of Leelavathy et al. (2011) requires a calculation of the nearest integer value v_1 by taking modulo of a_k with Δ.

$$v_1 = a_k \bmod \Delta.$$

5: In case the codeword contains all 1s, a_k coefficients are modified as follows:

$$a_k^1 = a_k - v_1 + z_k^1.$$

Where, z_k^1 is given as follows:

$$z_k^1 = \begin{cases} -\Delta/8, & \text{if } v_1 < \Delta/8, \\ 3\Delta/8, & \text{if } \Delta/8 \leq v_1 < 5\Delta/8, \\ 7\Delta/8, & \text{if } 5\Delta/8 \leq v_1 < \Delta. \end{cases}$$

In case the codeword contains all 0s, then the a_k coefficients are modified as follows:

$$a_k^0 = a_k - v_1 + z_k^0.$$

Where, z_k^0 is given below.

$$z_k^0 = \begin{cases} \Delta/8, & \text{if } v_1 < 3\Delta/8, \\ 5\Delta/8, & \text{if } 3\Delta/8 \leq v_1 < 7\Delta/8, \\ 9\Delta/8, & \text{if } 7\Delta/8 \leq v_1 < \Delta. \end{cases}$$

6: a_k^1 and a_k^0 are saved as BF^1 and BF^0 in a text file, respectively.

7: BF^1 and BF^0 are permuted with the permutation key σ_j.

8: The permuted BF^1 and BF^0 are then encrypted with K_{ses_j}. Both the permuted and encrypted BF^1 and BF^0 are saved in a block form in text format.

9: An inverse 4-level DWT is applied on the detail coefficients, and then the obtained values, the P and B-frames, and the audio of the original video file X constitute *SF* in a compressed (ZIP) form.

5.3.1.3 Base File Distribution Protocol

On receiving a file request from a buyer B_i, *SP* provides him/her the details of M who has the requested content. For a secure distribution of BF to B_i, M, MO, B_i and a selected set of Pr_j perform an asymmetric fingerprinting protocol. Fig. 5.6 illustrates the distribution protocol of BF.

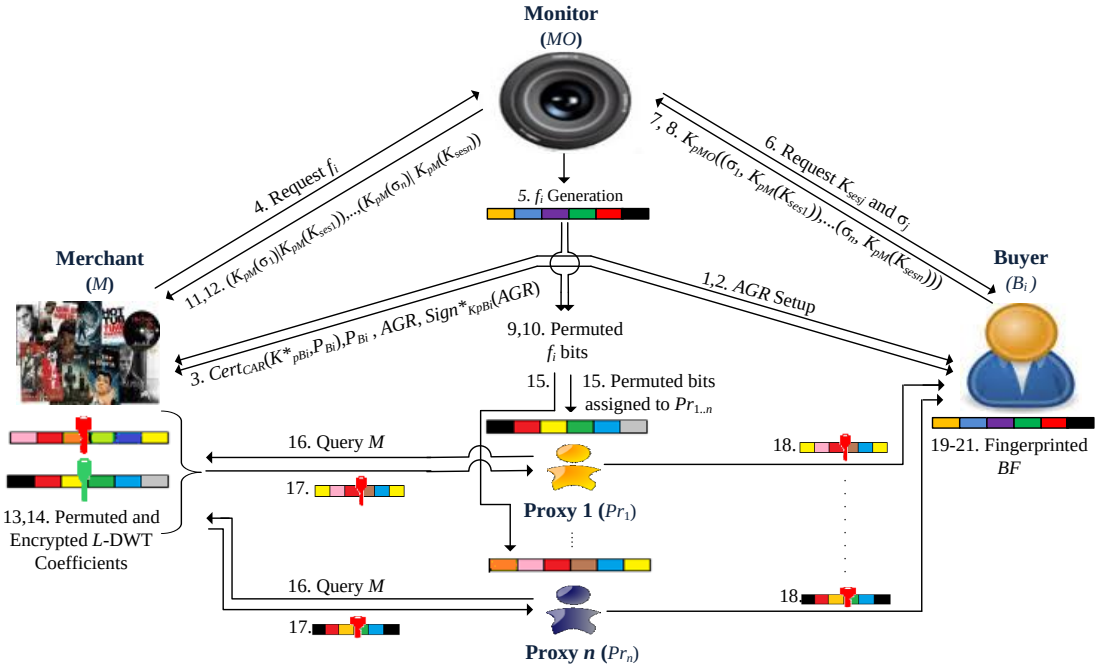
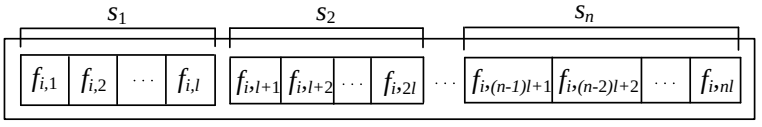
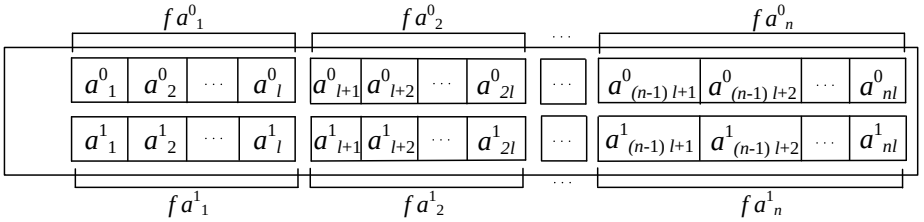


FIGURE 5.6: BF distribution protocol of FPSUM-PD



(a) Set of segments s_j of a fingerprint f_i



(b) Sets of contiguous fragments (fa^0_j,fa^1_j)

FIGURE 5.7: Permutation of the fingerprint and the approximation coefficients

Protocol 3 Steps that are executed between MO, M, B_i, and Pr_j to distribute the fingerprinted BF to B_i

1: Before starting a purchase negotiation of the multimedia content with the merchant, B_i generates a pseudo-identity to keep his/her anonymity. For pseudo-identity generation, CA_R generates a random number r and shares it only with B_i.
2: B_i negotiates with M to set-up an agreement (AGR) that explicitly states the rights and obligations of both parties and specifies the multimedia content (X). AGR uniquely binds this particular transaction to X. During the negotiation process, B_i uses his/her pseudonym P_{B_i} to keep his/her anonymity.
3: After the negotiation, B_i generates a key pair $(K^*_{pB_i}, K^*_{sB_i})$, signs the public key with his/her private key, and sends $\text{Sign}_{\text{B}_\text{i}}(K^*_{pB_i}, P_{B_i})$ to CA_R. CA_R verifies $\text{Sign}_{\text{B}_\text{i}}(K^*_{pB_i}, P_{B_i})$ using the public key of B_i. If valid, he/she generates an anonymous certificate $\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$ and sends it to B_i. B_i then sends $\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$, AGR, P_{B_i} and $\text{Sign}_{K^*_{pB_i}}(AGR)$ to M.
4: M verifies the received certificate, using the CA_R public key and the signature of the agreement using the certified key K_{pB_i}. If the received data is valid, then M generates a transaction ID (TID) for keeping a record of the transaction between him and B_i, and sends a request for a fingerprint to MO by sending $\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$, $\text{Cert}_{CA_R}(M)$, TID, AGR, P_{B_i} and $\text{Sign}_{K^*_{pB_i}}(AGR)$. If the received certificates and signatures are not valid, then the transaction is terminated by M.
5: MO validates the certificates and signatures of M and B_i from CA_R. After successful verification, MO generates a Nuida's c-secure codeword f_i of length m and randomly selects n proxy peers (Pr_j, for $j = 1, \ldots, n$) for a secure transfer of fingerprinted BF from M to B_i.
6: MO sends a request for permutation keys σ_j and session keys K_{ses_j} to B_i.
7: After receiving a request from M, B_i generates n random permutation keys σ_j (for $j = 1, \ldots, n$) of length $l = \lfloor m/n \rfloor$ and n session keys K_{ses_j}. The session keys are generated to be shared with M, such that the proxy peers that are responsible for transferring the fingerprinted a_k to B_i are unable to see the clear-text of a_k.
8: B_i encrypts K_{ses_j} with K_{p_M} and sends $K_{p_{MO}}(\sigma_j | K_{p_M}(K_{ses_j}))$ to MO.
9: MO decrypts $K_{p_{MO}}(\sigma_j, K_{p_M}(K_{ses_j}))$ with $K_{s_{MO}}$ and obtains σ_j and $K_{p_M}(K_{ses_j})$.
10: MO divides f_i into n segments (s_j) of length l (as shown in Fig. 5.7(a)) and permutes each segment using the permutation keys σ_j (Fig. 5.7(b)) in the same order as received by B_i.
11: MO waits for a specific time τ such that it receives multiple requests of a content from different buyers. If by that specified time, MO receives other requests, then the steps $1 - 10$ are repeated for the new buyer.
12: For each B_i, MO sends $E_{K_{p_M}}(\sigma_j) | E_{K_{p_M}}(K_{ses_j})$ to the corresponding M.
13: M decrypts $E_{K_{p_M}}(\sigma_j) | E_{K_{p_M}}(K_{ses_j})$ with K_{s_M} and obtains σ_j and K_{ses_j}.
14: M permutes sequentially both pre-computed variants of the modified approximation coefficients with σ_j. An exchange of σ_j between M and MO is performed to ensure that proxy peers do not obtain the positions of the permuted fingerprint bits. M then encrypts the permuted approximation coefficients' variants with K_{ses_j}.
15: The contiguous permuted fingerprint segments (ps_j) are then sequentially assigned to n proxy peers (Pr_j) by MO.
16: Pr_j contact M in a sequential manner to obtain the fragments of encrypted and permuted approximation coefficients (fa^0_j,fa^1_j).
17: M sends a set of encrypted and permuted fragments of pre-computed approximation coefficients $\left\{fa^0_j, fa^1_j\right\}$ to Pr_j.
18: Pr_j selects the correct pre-computed (permuted and encrypted) approximation coefficients a'_k from the received coefficients $\left\{fa^0_j, fa^1_j\right\}$ using the assigned permuted fingerprint segment ps_j, as shown in Fig. 5.8.

19: When B_i receives the encrypted and permuted approximation coefficients from a proxy peer, it permutes back the encrypted coefficients with ${\sigma_j}^{-1}$. With K_{ses_j}, B_i decrypts the received encrypted approximation coefficients and obtains the fingerprinted coefficients of *BF*.
20: B_i obtains his/her complete copy of *BF* by composing all the coefficients received sequentially from all Pr_j.
21: An inverse *L*-level DWT is applied on *BF* to obtain a fingerprinted *BF*, which is then recombined with *SF* obtained from the P2P network.

5.3.1.4 Supplementary File Distribution Protocol

On joining the system, a peer constructs an onion path with existing peers which points to it and adds this path to his/her associated *SP*. By doing so, a requesting peer (*RP*) can use this onion path to contact the content providing (*CP*) peer while knowing nothing about the *CP*'s identity. The peer requests for a particular file to *SP* of his/her group. If found, he/she displays the list of the peers having that particular file; else he/she sends a request for the file to other connected *SP*s. The other *SP*s, on finding the particular *CP*, send the response to the requesting *SP*. *SP* then establishes a path between *RP* and that *CP* peer. After receiving a positive reply from *CP* peer, the requesting peer initiates a two-party authenticated key exchange (AKE) protocol as shown in Fig. 4.4 (cf. Section 4.5.1.4), to authenticate each other identities and exchange the content of *SF* anonymously. For anonymous data exchange, a one-time session key is generated during the AKE protocol to encrypt the contents of *SF*. The details of *SF* distribution can be found in Section 4.5.1.4.

5.3.1.5 Traitor-tracing Protocol

A traitor-tracing protocol is executed by *MO* on receiving a request from *M* that a codeword *pc* has been extracted from a pirated copy Y', and B_i corresponding to *pc* needs to be identified. Before the execution of the traitor-tracing protocol, *M* needs to extract *pc* from Y', and this is achieved using the fingerprint extraction process. In the following, fingerprint extraction techniques for both audio and video data are presented.

Fingerprint extraction

The fingerprint extraction does not require the original audio signal. The fingerprint extraction procedure for an audio file is summarized as follows:

1. Let Y' be the pirated content on which the 3/4-level inverse DWT is performed to obtain the approximation coefficients in which the pirated code $pc \in \{0, 1\}^*$ is embedded.

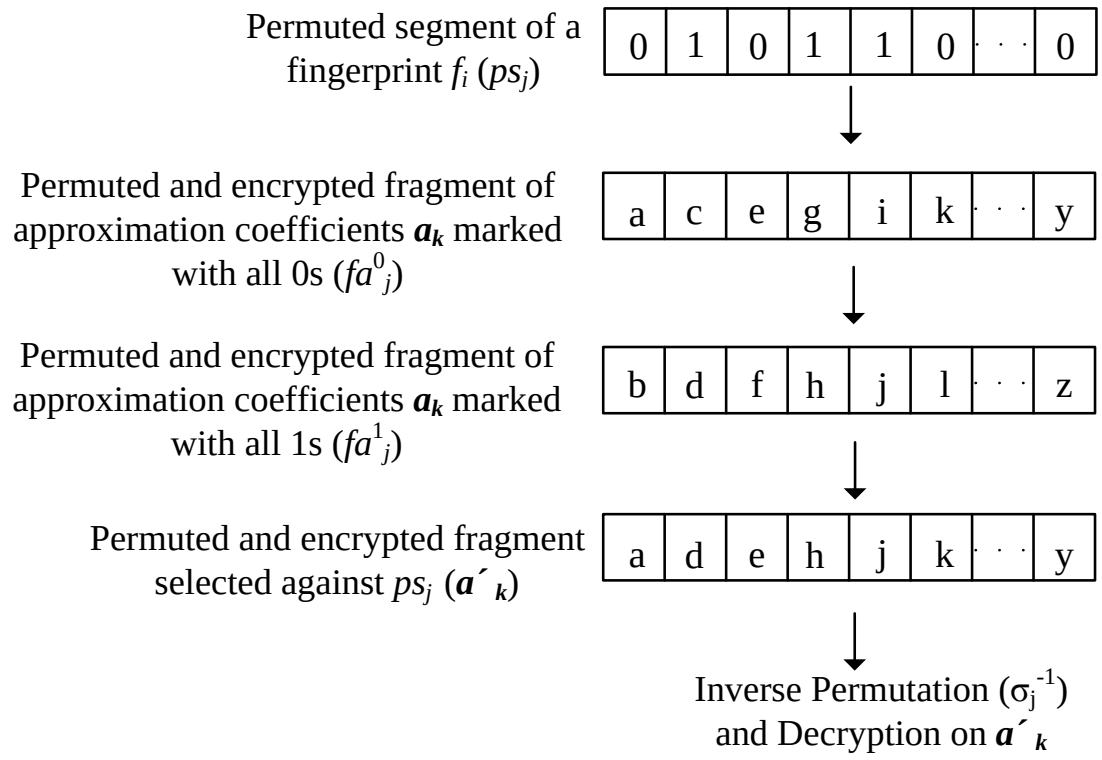


FIGURE 5.8: Fragment construction

2. The approximation coefficients are divided into non-overlapping frames $\mathcal{F}'_\ell$.

3. The vector norm is applied to each $\mathcal{F}'_\ell$ to obtain ω'_ℓ.

4. Compute $e = \omega'_\ell / \Delta$.

5. If e mod 2 = 0, then the embedded fingerprint bit is 0, otherwise it is 1.

The steps used in extracting a fingerprint from a video file are the same as the fingerprint embedding steps but in the reverse direction. The original video sequence is not required for the extraction procedure and, hence, the algorithm is blind.

1. Let Y' be the pirated video content, from which the key frames are extracted using the Canny-edge detection algorithm (Khurana & Chandak, 2013).

2. Each key frame in RGB format is converted to the $\mathcal{YUV}$ representation.

3. For each $\mathcal{Y}$ component, apply the 3/4-level inverse DWT to decompose $\mathcal{Y}$ into approximation (a_f) and detail coefficients.

4. Compute a nearest integer value v_2 by taking a_f modulo Δ:

$$v_2 = a_f \bmod \Delta.$$

5. The fingerprint pc is detected as follows:

$$pc = \begin{cases} 0, & \text{if } 0 \leq v_2 < \Delta/4 \text{ or } \Delta/2 < v_2 < 3\Delta/4, \\ 1, & \text{if } \Delta/4 \leq v_2 < \Delta/2 \text{ or } 3\Delta/4 \leq v_2 < \Delta. \end{cases}$$

Traitor tracing

Once pc is extracted by M from Y', M sends the extracted pc to MO which performs the tracing algorithm of Nuida et al.'s codes to identify the colluder(s). The output of this tracing algorithm is B_i with the highest score. The details of the tracing algorithm can be found in Algorithm 3 presented in Section 4.5.1.5. The real identity of B_i is not known to MO, only the pseudo-identity of the guilty B_i is revealed. MO retrieves a TID that contains the fingerprint f_i from his/her database for the arbitration and identification protocol.

5.3.1.6 Arbitration and Identification Protocol

The goal of the arbitration and identification protocol, performed between M, MO, CA_R and J, is to reveal the real identity of the traitor or reject the claims made by M. In order to reveal the real identity of the traitor, MO sends $(Y', pc, K_{p_{MO}}(f_i))$ and M sends $\text{Cert}_{CA_R}(K^*_{pB_i}, P_{B_i})$, $\text{Cert}_{K_{pB_i}}(K^*_{pB_i})$, AGR, $\text{Sign}_{K^*_{pB_i}}(AGR), K^*_{pB_i}$ to J. J verifies the validity of all the certificates and the signatures. If valid, it asks MO to decrypt $E_{K_{p_{MO}}}(f_i)$. If pc and f_i match with a high correlation, it requests CA_R to give the real identity of B_i. Otherwise, B_i is proved innocent.

5.4 Results and Discussion

This section examines how the design goals of FPSUM-PD described in Section 5.2.2.3 are achieved. The security analysis provide formal proofs and informal analysis concerning the correctness and soundness of the protocols of FPSUM-PD in terms of security and privacy. The performance analysis examines the performance of the protocols of FPSUM-PD in terms

of robustness, imperceptibility, computational and communicational costs and cryptographic overhead.

5.4.1 Security Analysis

This section analyzes the security and privacy properties of FPSUM-PD according to the design requirements and the attack model presented in Sections 5.2.2.3 and 5.2.2.4.

5.4.1.1 Formal Analysis of the *BF* Distribution Protocol

Formal proofs are provided in this section to analyze the security of Protocol 3.

Theorem 5.1. *In Protocol 3, a malicious proxy peer is unable to obtain a secret permutation key σ_j transmitted from B_i to MO or from MO to M.*

Proof. *MO* initiates a fingerprinting protocol with M and B_i only after verification of certificates and signatures from CA_R. The secret permutation key transferred between B_i and MO or between MO and M is encrypted with the public key of MO or M, respectively. Thus, in order to obtain σ_j, the malicious peer needs the private key of MO or M to decrypt $K_{p_{MO}}(\sigma_j)$ or $K_{p_M}(\sigma_j)$. □

Theorem 5.2. *An honest buyer is protected, in Protocol 3, from a conspiracy attack of malicious proxy peers who try to recombine their segments of a fingerprint and/or the fingerprinted content obtained from the merchant.*

Proof. In case Pr_j try to obtain a correct fingerprint by recombining their assigned permuted segments ps_j (with length of each segment equal to l), Pr_j would need to compute $l!$ combinations each on the colluded fingerprint f_i''. Thus, with more m-bits in f_i, Pr_j would need to carry out an increased number of permutations in order to obtain a correct fingerprint, which would be computationally infeasible.

In the second case, if all Pr_j combine their permuted and encrypted fragments $E_{K_{ses_j}}(a'_j)$ obtained from M apart from the permutation issue, they cannot decrypt these fragments. The fragments can only be decrypted by K_{ses_j}, which are known only to M and B_i. Hence, Pr_j are unable to obtain clear-text fingerprinted fragments to produce a fingerprinted copy similar to the buyer's copy. □

For example, imagine a randomly permuted fingerprint f_i of length 90-bits and three proxy peers Pr_1, Pr_2 and Pr_3. If each proxy peer carries 30 bits, in case Pr_1, Pr_2 and Pr_3 collude and obtain $f_i^{''}$, they need to compute 30! combinations each, resulting in $30! \cdot 30! \cdot 30! = (30!)^3$ total combinations to try to obtain a valid f_i.

5.4.1.2 Security Attacks on the *BF* Distribution Protocol

This section discusses several attack scenarios presented in Section 5.2.2.4 which can occur during the *BF* distribution protocol execution. Since the distribution of *SF* is identical to *SF* distribution protocol of FPSUM-HE (cf. Section 4.5.1.4), the security analysis of communication attacks on *SF* are same as discussed Section 4.6.1.4.

Buyer frameproofness

The possible collusion of proxy peers Pr_j cannot frame an honest buyer and held him/her responsible for illegal re-distribution (formally proved in Theorem 3). Also, M alone is unable to produce a fingerprint f_i, since MO is responsible for generation of f_i. However, it may be possible that a malicious M colludes with MO to frame an honest buyer for illegal re-distribution. Similarly, another possible collusion can occur between the proxy peers and M.

In the first scenario, the collusion can be disregarded since MO is an entity that is trusted by both M and B_i (as described in Section 3.2). In the second case, when Pr_j query M to obtain the permuted pre-computed $\left\{fa_j^0, fa_j^1\right\}$, it might be possible that both M and Pr_j collude to obtain a valid fingerprint codeword or a fingerprinted copy. Since M has a clear-text of σ_j, it could permute the fingerprint bits obtained from all the proxy peers by using σ_j and obtain a valid fingerprint of a buyer. However, this conspiracy attack against an honest buyer requires that all the proxy peers (n) collude with M, thus making a collusion size equal to $n + 1$.

In addition, the merchant would not be interested in forming such a big collusion at a price of being possibly caught since it is possible that one of the proxy peers be honest and refuse to become a part of this coalition. Then this proxy peer can report about the collusion between M and remaining proxy peers to MO. It may be noted that if less than n proxy peers collude with M, then the probability of framing an honest buyer is very low. For example, if $n = 10$ with each proxy peer carrying $l = 10$ bits, and 20% of the proxies are malicious, then the probability of obtaining a valid fingerprint would be $0.2^{10} \approx 10^{-7}$, which is very low.

In FPSUM-PD, this conspiracy attack can be countered by compelling MO to wait for a

particular time period τ, so that by the expiry of τ, it receives more fingerprint requests from M for different buyers. By doing so, M would be accessed by various Pr_j at a time and keeping record of various bits of multiple proxy peers could be infeasible. Also, a reward mechanism can be introduced within FPSUM-PD so that proxy peers can obtain rewards, such as discounts or bonus points, for their good reputation and reliability.

Also, it could be possible that M tried to find an identity of the buyer by relating proxies to each buyer. For example, if the permuted and encrypted approximation coefficients were transferred from M to two buyers B_1 and B_2 through n and $n-2$ proxy peers, respectively. It would be easier for M to figure out that a particular set of proxy peers Pr_j with $j = 1, \ldots, n-2$ are carrying a fingerprint for a buyer B_2 or $Pr_j^{'}$ (with $j = 1, \ldots, n$) are carrying another fingerprint for B_1. Thus, to avoid a possible attack of M on B_i, the number of proxy peers is fixed to n.

Non-repudiation

From the perspective of M, FPSUM-PD is secure and fair because B_i has no idea about the original digital content and the embedded fingerprint in the purchased copy. Also, B_i cannot claim that a pirated copy is created by M since the fingerprint is generated by MO which is trusted by both B_i and M. Thus, B_i cannot accuse MO of collaborating with M to frame him/her (as described in Section 3.2). However, there can be two cases where copyright protection scheme could be broken.

1. Since the proxies receive the permuted-encrypted coefficients a'_k, a possible collusion of B_i and all (or some of) Pr_j makes it possible to obtain the complete (or partial) set of coefficients and produce non-fingerprinted copies of the content, as B_i has everything he/she needs, namely, symmetric key and permutation keys. In this case, a possible B_i and Pr_j collusion is prevented by assigning the task of selecting Pr_j to MO. Consequently, B_i should create a collusion with Pr_j that are anonymous to him/her. But it is too risky, since honest Pr_j would accuse B_i of this misbehavior. However, if it is considered that the risk of this collusion cannot be overlooked (because even a single fragment leaked could be dangerous), there is a solution. The communication between Pr_j and B_i could be implemented using a path created by MO. In this way, B_i would not even know the Pr_j who originated the fragment and he/she would be required to build a collusion with all the nodes of the all the paths for all the fragments, which is unrealisable.

2. The malicious Pr_j may choose a combination of approximation coefficients that does not correspond to the fingerprint bits. For example, Pr_j may choose the 1-coefficient

when the corresponding bit is 0. In this scenario, the malicious Pr_j would not obtain any benefit by acting in a dishonest way, since the content obtained by B_i would not carry a valid fingerprint. However, this malicious act could be evaded, again, by using the paths created by *MO* between Pr_j and B_i. Some of the nodes of this path could randomly decide to send the fragment to *MO* to check whether the embedded information coincides with the corresponding fingerprint segment. In case of a mismatch, Pr_j would be detected as malicious. Thus, it would be risky for Pr_j to act in this way, since they would not know the nodes of the path created by *MO*.

Furthermore, from an analysis of buyer frameproofness property, it is obvious that there is a very low probability that a correct fingerprint or a fingerprinted content is obtained from a possible collusion between the proxy peers and M. Thus, it is impossible for B_i to deny an act of copyright violation. Also, FPSUM-PD provides a tracing mechanism to unambiguously identify a copyright violator once a pirated copy Y' is found.

Unlinkability

B_i's online activities cannot be linked with his/her personal information since each B_i is permitted to compute multiple pseudonyms and anonymous key pairs for his/her transactions with M and other buyers. For each transaction, B_i can randomly choose one of each of pseudonyms and anonymous key pairs to attain unlinkability.

Collusion resistance

Nuida et al.'s codes are c_0-secure with ε-error with $c \leq c_0$ (c is the number of pirates). In FPSUM-PD, $c_0 = 3$ with $\varepsilon = 10^{-3}$ and $N = 10^5$ (N = number of users) are considered, thus a code of size $m = 267$ bits is obtained. This code is then embedded into the content to uniquely identify the user. As long as c remains lower than c_0 and the piracy tracing Algorithm 3 (cf. Section 4.5.1.5) is followed, the copyright violator can be identified successfully. Thus, the proposed scheme offers resistance against three colluders.

Man-in-the-middle attack

In FPSUM-PD, the deployment of PKI ensures mutual authentication between entities (M, B_i, *MO*), and thus the communication between the entities is authenticated and the possibility of eavesdropping can be defied. Furthermore, secret keys transferred from B_i to *MO* or from *MO* to M are encrypted with the receivers public keys to prevent tampering of the secret data.

5.4.1.3 Security against Collusion Attacks

This section discusses the robustness of the fingerprinting scheme against the linear (averaging) and non-linear (minimum, maximum and median) collusion attacks presented in Section 5.2.2.4. The attacks are performed on the sample video file "Dragon" (cf. Table 5.11) with varying number of colluders U. Under averaging attack, each pixel in pirated video is the average of the corresponding pixels of the fingerprinted videos associated with the colluders U. For minimum, maximum and median attacks, each pixel in the pirated video is the minimum, maximum or median, of the corresponding pixels of the fingerprinted video.

Table 5.1 shows the number of colluders U which have been successfully traced through Nuida et al. (2007) codes tracing Algorithm 3 (cf. Section 4.5.1.5). In all cases, the colluders have been successfully traced by analyzing a pirated video copy Y'. In order to test the resistance of the fingerprint against more than 3 colluders, the fingerprint codewords are generated using $c_0 = 4$ and $c_0 = 5$, which results into codewords with an increased length m. The reason that the number of colluders U is restricted to 5 is due to a fact that an increase in U degrades the quality of the content. Thus, to provide a better trade-off between collusion resistance property and imperceptibility, a lower value of c_0 is selected.

TABLE 5.1: Security against collusion attacks

No. of Colluders	**No. of Colluders Detected for Attacks**			
U	**Average**	**Minimum**	**Maximum**	**Median**
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5

5.4.2 Performance Analysis

Four experiments have been performed to show the performance of FPSUM-PD. These experiments are: the computation of transparency to show the Objective Difference Grade (ODG) and the PSNR of the fingerprinted audio and video files, the evaluation of the robustness of the fingerprint against signal processing attacks, the execution times of file partition into *BF* and *SF* files, and the calculation of the cryptographic overhead, . The experiments have been developed in Matlab 7.0 and C++ with three audio and three video files with varying sizes, on a workstation equipped with an Intel i-7 processor at 3.4 GHz and 8 GB of RAM. The fingerprint generation, file partitioning, *BF* distribution and traitor tracing protocols are implemented in

Matlab 7.0, whereas *SF* distribution protocol is executed in the C++ programming language. The simulation parameters for fingerprint generation, *BF* and *SF* generation, and *BF* and *SF* distribution protocols are presented in Table 5.2.

TABLE 5.2: Simulation and experimental parameters

Name	Value	Description
N	10^5	No. of users
c_0	3	No. of colluders
ε	10^{-3}	Error probability
L	3/4	Levels of DWT decomposition
n	10	No. of Proxy peers Pr_j
σ_j	10	No. of permutation keys
K_{ses_j}	10	Set of one-time session keys
τ	5 secs	Fixed time period set for *MO*
l	$\lfloor 267/10 \rfloor = 26$	Length of the permutation keys σ_j
α_1	0.4	Constant used in calculation of Δ for audio fingerprinting
α_2	2.5	Constant used in calculation of Δ for audio fingerprinting
$attack_{SNR}$	25	Constant used in calculation of Δ for audio fingerprinting
Δ	0.50	Quantization step size for video fingerprinting
a'	2	Constant in key frame's threshold calculation
$h(\cdot)$	160-bits	SHA-1 function
$E(\cdot)$	128-bits	Symmetric encryption/decryption
P	1024-bits	Prime number $\in$ finite cycle group $\mathbb{G}$
Q	160-bits	Prime number that divides $P-1$
r	1024-bits	Secret number used in pseudo-identity generation
Υ_1/Υ_2	1024-bits	Secret number used in authentication
γ_1/γ_2	160-bits	Secret numbers used for session key generation
T_a/T_b	2/3	Tail nodes in onion routing

5.4.2.1 Analysis of Audio Fingerprinting

The performance analysis of three audio files, namely, "LoopyMusic", "Hugewav" and "Aasan Nai Yahan" are discussed in this section in terms of imperceptibility, robustness, computational, communicational and cryptographic costs. The details of the three audio files are presented in Table 5.3. The simulation and experimental parameters are the same as those in Table 5.2. However, in the experiments for audio file partitioning algorithm of "LoopMusic", level-3 DWT decomposition with a 4-coefficient Daubechies (*db*4) filter is used, and for "Hugewav" and "Aasan Nai Yahan" audio files, level-4 DWT decomposition with a *db*4 filter is used. The number of levels of DWT decomposition is selected to provide a convenient trade-off between

robustness, capacity and imperceptibility. In case of stereo files, the experiments and simulations are performed for each channel separately. The size of each *BF* and *SF* of three audio files are also presented in Table 5.3.

TABLE 5.3: Details of audio files

Details	**Loopy Music**	**Hugewav**	**Aasan Nai Yahan**
Time Length (min:sec)	00:10	00:17	03:34
File Size (MB)	0.89	2.97	36.01
Format	WAV	WAV	WAV
Bits per Sample	16	16	16
Sample Rate (Hz)	44100	44100	44100
Channel Mode	Mono	Stereo	Stereo
Base File Size (MB)	0.11	0.29	3.58
Supplementary File Size (MB) with double-bit precision	1.79	5.94	72.16

5.4.2.1.1 Transparency

The ODG is the objective measure to evaluate the imperceptibility of the embedding algorithm, and it is a good objective evaluation of auditory quality for the audio watermarking technology, which is used to calculate the imperceptible difference between the reference (original) signal X and the test (fingerprinted) signal Y. It can be obtained by the *Opera* (1999) software. The ODG ranges from 0 to -4 (corresponding to imperceptible to very annoying) as shown in Table 4.4 (cf. Section 4.6.2). Table 5.4 presents the imperceptibility results as ODG of the three fingerprinted audio files. It is evident that all ODG values are between 0 (not perceptible) and -1.0 (not annoying), showing excellent behavior in terms of the imperceptibility. Thus, the imperceptibility results confirm that the fingerprinted audio signals are perceptually similar to the original audio signals.

TABLE 5.4: ODG of audio files

Audio Files	**ODG**
Loopy Music	-0.2
Huge Wave	-0.58
Aasan Nai Yahan	-0.62

The results of the embedding audio algorithm are compared with Huang et al. (2012) as

shown in Table 5.5. Huang et al. proposed an audio watermarking algorithm based on the human auditory system which applies the theory of dither modulation (DM). The use of the human auditory system masking effect is to achieve DM algorithm for the adaptive selection of the quantization step Δ. The DWT low-frequency coefficients in the audio are used for embedding. It is evident from Table 5.5 that the embedding algorithm of FPSUM-PD (Xinkai et al., 2013) shows better performance than (Huang et al., 2012) in terms of imperceptibility.

TABLE 5.5: Comparison with imperceptibility results

Algorithm	**FPSUM-PD (Xinkai et al., 2013)**	**Huang et al. (2012)**
ODG	−0.20	−0.60

Robustness against signal processing attacks

Table 5.6 presents the robustness results of an audio file "LoopyMusic" against signal processing attacks such as re-quantization, re-sampling, MP3 compression and AWGN (cf. Section 5.2.2.4). The bit error rate (BER) and normalized correlation (NC) are used to evaluate the robustness between the original fingerprint f_i and the extracted fingerprint f_i'. BER values closer to zero and NC values closer to 1 indicate robustness against signal processing attacks.

TABLE 5.6: Robustness of an audio file against signal processing attacks

Attacks	**Parameters**	**BER**	**NC**	**Traceability**
Re-quantization	16-8-16 bits	0.00	1.000	Yes
Re-sampling	44.1-22.05-44.1 kHz	0.00	1.000	Yes
MP3 Compression	256 kbps	0.03	0.979	Yes
AWGN	18 dB	0.08	0.925	Yes

The results in Table 5.6 show that the selected embedding algorithm (Xinkai et al., 2013) provides excellent performance against common signal processing attacks for "LoopyMusic". The minimum BER and the maximum BER values are 0% and 8% respectively and, similarly, the minimum NC and the maximum NC values are 92.5% and 100% against different signal processing attacks. Moreover, the last column of Table 5.6 shows that the fingerprint of a buyer is traceable against these common signal processing attacks. Thus, these results indicate that the fingerprint embedding algorithm satisfies the fingerprint's robustness requirement.

The robustness results of the audio algorithm (Xinkai et al., 2013) used in FPSUM-PD are compared with Yong-Mei, Wen-Giang, and Hai-Yang (2013) as shown in Table 5.7. Yong-Mei et al. proposed an audio blind watermarking algorithm scheme based on DWT and singular value decomposition (SVD). In their scheme, an audio signal is split into blocks and each block is decomposed with DWT. The first quarter audio approximate sub-band coefficients are then decomposed further with SVD transform to obtain a diagonal matrix. The watermarking information is embedded into the diagonal matrix.

TABLE 5.7: Comparison with BER and NC values

Attacks	**Parameters**	**Algorithm used in FPSUM-PD (Xinkai et al., 2013)**		**Algorithm proposed by Yong-Mei et al. (2013)**	
		BER	**NC**	**BER**	**NC**
Re-quantization	16-8-16 bits	0.00	1.000	0.26	0.731
Re-sampling	44.1-22.05-44.1 kHz	0.00	1.000	0.00	1.000
MP3 Compression	256 kbps	0.03	0.979	0.00	1.000
AWGN	18 dB	0.08	0.925	0.15	0.860

Table 5.7 demonstrates the performance of the algorithm (Xinkai et al., 2013) used in FPSUM-PD in comparison with algorithm (Yong-Mei et al., 2013) under various attacks. The results in Table 5.7 reveal that the (Xinkai et al., 2013)'s algorithm demonstrates a superior response against conventional attacks and, furthermore, in all cases, all NC values are close to 1 and all BER values are close to 0.

Computational and communicational costs

This section discusses the performance of FPSUM-PD in terms of computation and communication time. The time taken in generation of Nuida et al.'s fingerprint codes and partitioning of original multimedia file X into BF and SF is considered as a computation time. For BF and SF generation, the implementation of the file partition protocol in Section 5.3.1.2 contributes to the total computation time. Table 5.8 shows the execution time of three audio files in seconds.

The communication time (or response time) is the time calculated from the query issuance of a peer to the download of BF and SF to reconstruction of the file. The response time in Table 5.9 is calculated as a time taken in BF distribution from M to B_i through Pr_j, the complete transfer of SF from the providing peer to the requesting peer through an anonymous path, and the reconstruction of a file at B_i's end. The response time is Table 5.9 summarizes the response time for the audio file "LoopyMusic".

TABLE 5.8: Computation time of an audio file

File Name	CPU Time (secs)			
	Fingerprint generation	***BF* generation**	***SF* generation**	**Total Time**
LoopyMusic	6.01	1.24	0.034	7.29
Hugewav	6.01	2.12	0.18	8.31
Aasan Nai Yahan	6.01	17.99	1.196	25.20

TABLE 5.9: Communication time of an audio file

File Name	Communication Time (secs)				
	***BF* Delivery Time**	***SF* Delivery Time**	**File Reconstruction**	**Total Distribution Time**	**Direct Delivery Time**
LoopyMusic	5.58	10.00	3.09	18.67	7.00

The last column of Table 5.9 shows the execution time of a direct file transfer between M and B_i without considering security, privacy and accountability properties. The direct delivery time is calculated as a time taken to download "LoopyMusic" at a bit rate of 1.5 Mbps. It can be seen, from the table, that the distribution time of *BF* is small as compared to the direct transfer time. Thus, FPSUM-PD enables the merchant to save file delivery and CPU time by using the P2P system for majority file distribution. Moreover, the total response time presented in the 4th column of Table 5.9 represents the addition of the individual time of each process (*BF* and *SF* distribution and reconstruction). Since the audio file is divided into two parts: *BF* and *SF*, and *BF* is downloaded in a centralized manner between a peer, M, Pr_j and MO, whereas, *SF* is delivered to a peer through middle peers from *SP* in a P2P manner, the *BF* and *SF* protocols can be initiated simultaneously at a request of a peer to *SP* without interfering with each other. The parallel execution of *BF* and *SF* distribution protocols could result in the reduction of the total distribution time of *BF* and *SF* from 5.58 + 10.00 = 15.58 seconds to 10 seconds. The reduced time (10 seconds) is slightly longer than the direct delivery time (7 seconds) which in fact does not incorporate security and privacy properties.

However, the concurrent execution of the protocols depends on the bit rate available at the buyer's end. For example, in case of "LoopyMusic" audio file, the parallel execution of *BF* and *SF* protocols require a total bit rate of 1.584 Mbps at a buyer's end. It could be a problem for the peers with a downloading bit rate limited to 1.5 Mbps or less. However, with constant advancements in Internet and its related technology, nowadays the bit rate offered to home users by the Internet service providers typically ranges from 512 kbps to 10 Mbps in the direction

to the downstream. Thus, with the availability of increased bandwidth capacities, the parallel execution of the protocols can be easily performed.

Cryptographic costs

Cryptographic algorithms are applied in FPSUM-PD to ensure the desired level of security, privacy and accountability. The cryptographic algorithms are implemented in C++ using the *NTL: A Library for doing Number Theory* (1990) library. Table 5.10 shows the CPU execution time of each cryptographic block for achieving the desired security for the audio file "Loopy-Music". It is evident, from the table, that the anonymous paths construction and authentication through these paths is an expensive cryptographic operation in FPSUM-PD. However, in achieving anonymity in P2P systems, there is always a cryptographic overhead. This overhead is due to encryptions and decryptions, insertion of fake traffic and increasing the routing path to provide anonymity between two communicating users. Still, the overhead of the authentication in FPSUM-PD is better due to the use of symmetric encryption, instead of applying asymmetric encryption. Public-key cryptography is used for the generation of an anonymous certificate and a key pair, and encrypting the small-sized session and permutation keys during BF distribution from M to B_i through Pr_j in the presence of MO.

TABLE 5.10: Cryptographic overhead of an audio file in FPSUM-PD

Cryptographic Algorithms	Time (secs)
Public-key cryptography	0.72
AES Encryption/Decryption	2.83
Anonymous Key Exchange	9.62
Total	13.17

5.4.2.2 Analysis of Video Fingerprinting

Three video files, namely, "Traffic", "Dragon" and "Breaking Bad" are used in FPSUM-PD for evaluating imperceptibility, robustness, computational, communicational and cryptographic costs. The details of the three video files are presented in Table 5.11. The simulation parameters are the same as presented in Table 5.2. Level-3 DWT decomposition with a $db4$ filter is used in experiments for video file partitioning algorithm of "Traffic", whereas for "Dragon" and "Breaking Bad" video files, level-4 DWT decomposition with a $db4$ filter is used. Table 5.11 presents the sizes of BF and SF, and it can be seen that the size of BF is relatively small.

TABLE 5.11: Details of video files

Detail	**Traffic**	**Dragon**	**Breaking Bad**
Time Length (min:secs)	00:10	23:00	50:00
File Size (MB)	0.19	51.10	305.00
Format	AVI	AVI	MP4
Resolution (pixels)	120×160	320×240	720×406
Total Frames	120	32,975	67,817
Key Frames	15	2,228	2,649
Base File Size (MB)	0.03	4.80	11.20
Supplementary File Size (MB)	0.18	69.40	216.00

Transparency

For video files, the imperceptibility is determined by the PSNR of the fingerprinted video. The PSNR provides a reliable indication of the variation of subjective video quality in decibels (dB). The PSNR values are obtained by using *MSU Video Quality Measurement Tool* (2011). Typical PSNR values for the fingerprinted video are between 30 and 50 dB, where higher values of PSNR indicate more imperceptibility of fingerprinting scheme. Table 5.12 presents the imperceptibility results as PSNR of three fingerprinted video files. The PSNR is above 35 dB in each case, and thus it can be inferred that the embedded fingerprint has no perceptible effect on the quality of the video file.

TABLE 5.12: PSNR of video files

Video Files	**PSNR in dB**
Traffic	42.00
Dragon	39.00
The Bad	36.00

The results of the embedding video algorithm (Leelavathy et al., 2011) used in FPSUM-PD are compared with W. H. Lin et al. (2009) as shown in Table 5.13. W. H. Lin et al. presented a blind watermarking method using a maximum wavelet coefficient quantization. The wavelet coefficients of a host image are grouped into blocks of varying size. A watermark is embedded in different sub-bands and each block is used to embed either the watermark bit 0 or the watermark bit 1. It is evident, from Table 5.13, that the embedding algorithm of FPSUM-PD (Leelavathy et al., 2011) shows better performance than (W. H. Lin et al., 2009) in terms of imperceptibility.

TABLE 5.13: Comparison with PSNR values

Algorithm	FPSUM-PD (Leelavathy et al., 2011)	W. H. Lin et al. (2009)
PSNR in dB	42.00	40.31

Robustness against signal processing attacks

Table 5.14 presents the BER and NC values of a video file "Dragon" tested for signal processing attacks such as median filtering, resizing, H.264 compression and Gaussian noise addition (cf. Section 5.2.2.4). The BER and NC are used to evaluate the robustness between f_i and f_i'.

TABLE 5.14: Robustness of a video file against signal processing attacks

Attacks	Parameters	BER	NC	Traceability
Median Filter	$[3 \times 3]$	0.05	0.966	Yes
Re-scaling	2	0.01	0.999	Yes
H.264 Compression	768 kbps	0.00	1.000	Yes
AWGN	20 dB	0.02	0.985	Yes

The results in Table 5.14 show that the selected embedding algorithm (Leelavathy et al., 2011) provides excellent performance against conventional signal processing attacks for "Dragon". The minimum BER and the maximum BER values are 0% and 5% respectively, and similarly, the minimum NC and the maximum NC values are 96.6% and 100% against different signal processing attacks. Moreover, the fingerprint of a buyer is traceable against these common signal processing attacks. Thus, these results indicate that the fingerprint embedding algorithm satisfies the fingerprint's robustness requirement.

The robustness results of the video embedding algorithm proposed by Leelavathy et al. (2011) are compared with the embedding algorithm of W. H. Lin et al. (2009) as shown in Table 5.15. The results in the table reveal that Leelavathy et al.'s algorithm demonstrates a superior response against common signal processing attacks, and furthermore, in all cases, the NC values are 1 or extremely close to 1 and all the BER values are 0 or slightly greater than 0.

Computational and communicational costs

The execution time of *BF* generation and distribution protocol for a video file involves the fingerprint (f_i) generation, f_i permutation, the assignment of the permuted bits to a selected set of Pr_j, key generation, the symmetric-key encryption of the pre-computed coefficients and the

TABLE 5.15: Comparison with BER and NC values

Attacks	Parameters	Algorithm used in FPSUM-PD (Leelavathy et al., 2011)		Algorithm proposed by W. H. Lin et al. (2009)	
		BER	NC	BER	NC
Median Filtering	$[3 \times 3]$	0.05	0.966	0.10	0.956
Re-scaling	2	0.01	0.999	0.02	0.985
H.264 Compression	786 kbps	0.00	1.000	0.01	0.999
AWGN	20 dB	0.02	0.985	0.10	0.905

transfer of encrypted coefficients to B_i from M. The Canny-edge detection technique used in the extraction of the key frames from the video file is performed only once by M. Similarly, the RGB conversion to $\mathcal{YUV}$ format and the DWT on the $\mathcal{Y}$ components of the key frames are applied once to obtain the approximation and detail coefficients. M stores the key frames, the inter frames (P and B), the approximation and detail coefficients of each video file. By doing so, M is able to avoid the costs of performing the Canny-edge detection technique, converting the RGB format frames to $\mathcal{YUV}$ format and applying the 3/4-level DWT on the key frames every time a video file is requested by a buyer. The execution time also includes the time taken to create SF, which is formed by taking an inverse 3/4-level DWT of the detail coefficients, and the conversion of P and B frames to the original video format. The SF execution time can also be saved by M since he/she has the detail coefficients and inter frames (P and B frames) stored at his/her end. Table 5.16 shows the computation time of the three video files.

TABLE 5.16: Computation time of a video file

File Name	CPU Time (secs)			
	Fingerprint generation	*BF* generation	*SF* generation	Total Time
Traffic	6.01	3.50	7.22	16.73
Dragon	6.01	32.99	24.16	63.16
Breaking Bad	6.01	66.03	36.15	108.19

The response time for BF distribution includes the time taken in BF distribution from M to B_i through Pr_j. Similarly, the response time for the distribution of SF is evaluated by considering the complete transfer of SF from the providing peer to the requesting peer through an anonymous path. The response time also includes file reconstruction time at the user end. Table 5.17 summarizes the response time for a video file "Traffic".

In BF delivery, the fingerprint generation and the embedding time is not considered. It is assumed that a database of fingerprint has been generated by MO before the start of the BF

TABLE 5.17: Response time of a video file

File Name	Communication Time (secs)				
	***BF* Delivery Time**	***SF* Delivery Time**	**File Reconstruction**	**Total Distribution Time**	**Direct Delivery Time**
Traffic	3.69	9.88	4.70	18.27	3.00
Breaking Bad	67.40	657.29	295.00	1019.69	1560.00

distribution protocol. Similarly, it is assumed that, before the execution of the *BF* distribution protocol, M has generated the pre-computed BF^0 and BF^1. The direct delivery time is calculated as the time taken to download files at a bit rate of 1.5 Mbps.

The total response time presented in Table 5.17 represents the addition of the individual time of each process (*BF* and *SF* distribution and reconstruction). From the table, it is evident that a direct delivery time (1560 seconds) of "Breaking Bad" is comparatively larger than the time taken by *BF* and *SF* distribution (67.40 + 657.29 = 724.69 seconds). Similar to the audio file distribution, the *BF* and *SF* protocols can be initiated simultaneously at a request of a peer to *SP* without interfering with each other. For example, in the case of the "Breaking Bad" video file, the parallel execution of *BF* and *SF* distribution protocols could result in the reduction of the total distribution time of *BF* and *SF* from 724.69 seconds to 657.29 seconds. The reduced time (657.29 seconds) is two times smaller than the direct delivery time (1560 seconds). However, the concurrent execution of the protocols depends on the bit rate available at the peer's end. For example, in case of "Breaking Bad" video file, the parallel execution of *BF* and *SF* protocols require a total bit rate of 3.95 Mbps at a peer's end. It could be a problem for the peers with a downloading bit rate limited to 3.95 Mbps or less. However, with the advancement in technology and the market expansion, the Internet service providers nowadays offer faster services with typical bit rates of up to 10 Mbps downstream. Thus, with the availability of higher bit rates, it is possible to carry out the parallel execution of the protocols easily.

Cryptographic costs

The cryptographic algorithms used in FPSUM-PD for data confidentiality and security are namely, AES-128 and 1024-bit public-key cryptosystem. Thus, the cryptographic costs include AES-128, 1024-bit asymmetric-key encryption/decryption and authentication between two peers. Table 5.18 shows the CPU execution time of the symmetric-key encryption/decryption of the "Traffic" video *BF* and *SF*, and an anonymous authentication process based on ZKPI between two peers. Public-key cryptography is used for generation of anonymous certificates

and a key pair, and encryption of small-sized session and permutation keys during *BF* distribution from *M* to B_i through Pr_j in the presence of *MO*.

On comparing the CPU execution time to perform asymmetric-key encryption/decryption of *BF* (8.80 secs) (cf. Section 4.6.2.2), the time taken to perform symmetric-key encryption/decryption of *BF* is considerably shorter. However, the anonymous paths construction and authentication through these paths is the most expensive cryptographic operation. As discussed in audio analysis (cf. section 5.4.2.1), anonymity is achieved at an additional cost.

TABLE 5.18: Cryptographic overhead of a video file in FPSUM-PD

Cryptographic Algorithms	**Time (secs)**
Public-key cryptography	0.72
AES Encryption/Decryption	0.98
Anonymous Key Exchange	9.62
Total	11.32

5.5 Conclusions

In this chapter, an efficient P2P content distribution system, i.e. a system that provides copyright and privacy protection to the merchant and the buyers, is presented. In contrast to the known asymmetric fingerprinting schemes, which use homomorphic encryption to embed a fingerprint into a multimedia content and inflict high computational and communication burden on a merchant, the proposed system lessens this cost for the merchant by only sending a small-sized base file composed of pre-computed fingerprinted information bits through proxies to the buyers. The main achievements of the FPSUM-PD system are (1) buyer security and privacy preservation, (2) collusion-resistance, (3) piracy tracing, and (4) efficient content distribution by avoiding multi-party security protocols, bit commitments and public-key cryptography of the multimedia content.

In FPSUM-PD, the multimedia content is partitioned into a small-sized base file and a large-sized supplementary file. The base file is dispensed by the merchant on payment from the buyer and a supplementary file is distributed through the P2P network. For generation and distribution of a base file, an asymmetric fingerprinting protocol is performed between the merchant and the buyer in the presence of a trusted monitor. The base file is distributed to the buyer through proxies in such a way that the merchant cannot predict about the fingerprinted content, and the proxies are unable to frame honest buyers by combining their information bits.

The buyer's privacy is preserved until he/she is found guilty of illegal re-distribution. The security and performance analysis show that FPSUM-PD is secure, privacy-preserving and efficient. The next chapter provides a comparative analysis of FPSUM-PD with FPSUM-HE and the P2P content distribution systems presented in Chapter 3.

Chapter 6

Comparative Analysis

In this chapter, the solutions presented in Chapter 4 (Framework for preserving Privacy and Security of User and Merchant based on Homomorphic Encryption, FPSUM-HE) and Chapter 5 (Framework for preserving Privacy and Security of User and Merchant with Proxy-based Distribution, FPSUM-PD) are compared in terms of imperceptibility, robustness against common signal processing attacks, security against collusion attacks, computational and communicational costs and cryptographic overhead. The chapter also compares FPSUM-HE and FPSUM-PD with the P2P content distribution systems presented in Chapter 3.

This chapter is organized as follows. Section 6.1 presents the comparative analysis of FPSUM-HE and FPSUM-PD. In Section 6.2, FPSUM-HE and FPSUM-PD are compared with other P2P content distribution systems in terms of guaranteed security and privacy properties. A conclusion is provided in Section 6.3.

6.1 Comparative Analysis of FPSUM-HE and FPSUM-PD

Though both FPSUM-HE and FPSUM-PD use a concept of partitioning a multimedia file into a small-sized base file and a large-sized supplementary file to lessen the computational cost of the merchant, the fingerprinting schemes proposed in both systems are different. In FPSUM-HE, content protection is provided through homomorphic encryption-based asymmetric fingerprinting, whereas FPSUM-PD proposes an asymmetric fingerprinting protocol without requiring a public-key encryption, bit commitments or multi-party security protocols. In Chapters 4 and 5,

both proposed frameworks are evaluated in terms of imperceptibility, robustness against common signal processing attacks, security against collusion attacks, computational and communicational costs and cryptographic overhead. In this section, both systems are compared with respect to the above defined evaluation criteria to demonstrate the improvement in efficiency and advantages of FPSUM-PD over FPSUM-HE.

6.1.1 Imperceptibility

Table 6.1 presents the imperceptibility results as ODG of three fingerprinted audio files for both FPSUM-HE and FPSUM-PD.

TABLE 6.1: Comparison of ODG values

Scheme	ODG			No. of Modified Coefficients		
	LoopyMusic	Hugewav	Aasan Nai Yahan	LoopyMusic	Hugewav	Aasan Nai Yahan
FPSUM-HE	−0.48	−0.98	−1.20	267	267	267
FPSUM-PD	−0.20	−0.58	−0.62	58,470	48,594	590,871

PSNR of three fingerprinted video files of both FPSUM-HE and FPSUM-PD are presented in Table 6.2.

TABLE 6.2: Comparison of PSNR values

Scheme	PSNR in dB			No. of Modified Coefficients		
	Traffic	Dragon	Breaking Bad	Traffic	Dragon	Breaking Bad
FPSUM-HE	44.00	42.00	41.00	534	14,685	17,622
FPSUM-PD	42.00	39.00	36.00	4005	59,4876	707,283

From the results of Table 6.1, it is evident that the imperceptibility results of the audio files in FPSUM-PD are relatively better than the results in FPSUM-HE. The watermarking schemes used to embed the collusion-resistant fingerprint into the audio files in both FPSUM-HE and FPSUM-PD are based on QIM watermarking. It can be seen, from Table 6.1, that only 267 approximation coefficients of all audio files are modified in FPSUM-HE in comparison to FPSUM-PD, where approximately all the approximation coefficients are modified. The low capacity of the embedding scheme in FPSUM-HE must corresponds to better imperceptibility (ODG) values. However, it can be seen in the table that worse (lower) ODG values are obtained in FPSUM-HE. These lower ODG values are obtained due to the fact that the collusion-resistant fingerprint needs to be encrypted first, and then embedded into an encrypted content to achieve

the desired security properties. The content is then decrypted to obtain a fingerprinted content. This encryption, embedding in the encrypted domain and decryption of the content, introduces additional noise that affects the imperceptibility of the audio signal.

An experiment is performed on an audio file "Hugewav" to prove the hypothesis that imperceptibility is affected due to encryption, embedding in the encrypted domain and decryption of the content. The SD-QIM embedding algorithm (Prins et al., 2007) of FPSUM-HE is employed twice to obtain a fingerprinted file. In the first case, the fingerprint f_i with length $m = 267$ bits is embedded into the 4-level DWT approximation coefficients (t_a). In the second scenario, the same fingerprint f_i is encrypted with a public key of a buyer, and then embedded into encrypted t_a using the SD-QIM scheme. Then, to obtain the fingerprinted copy, the content is first decrypted followed by an inverse 4-level DWT of decrypted approximation coefficients. The ODG values obtained for both cases are -0.40 and -0.98, respectively. Thus, the lower ODG value obtained in the second case is due to the fact that homomorphic encryption of a content requires integer quantization step sizes, thus introducing a distortion (Prins et al., 2007).

From Table 6.2, it can be seen that the PSNR values of all three video files in FPSUM-HE are slightly larger than the values of the video files in FPSUM-PD. The better PSNR values in FPSUM-HE are due to the fact that only a few key frames are selected from the total key frames of the video files to embed the collusion-resistant code. For example, in the case of the "Breaking Bad" video file, out of 2652 total key frames, 55 key frames are embedded with a collusion-resistant code, which implies that $17,622$ approximation coefficients are modified in comparison to FPSUM-PD, in which all the key frames are embedded with a collusion-resistant code, thus affecting $707,283$ approximation coefficients. Consequently, the selective embedding results in better PSNR values of video files since the selected key frames affect some inter-frames (P and B), which in turn produce a better quality fingerprinted video file. However, if all the key frames of the video file are embedded with a collusion-resistant code in FPSUM-HE, it would result in a lower quality fingerprinted video file. The reason for the lower PSNR value is similar to the one discussed in the audio fingerprinting case, since homomorphic encryption of the content requires integer quantization step sizes, which introduces distortion that in turn affects the imperceptibility of the video signal.

On the other hand, the PSNR values of three video files in FPSUM-PD are slightly lower than the PSNR values obtained in FPSUM-HE. This is due to the fact that all the key frames of the video files are embedded with a collusion-resistant code, which in turn affects all the inter-frames (P & B) during the formation of a fingerprinted video file. However, it is evident from Table 6.2, that embedding a fingerprint in all the key frames does not degrade the quality of the

fingerprinted video file, since typical PSNR values for the fingerprinted video file are between 30 and 50 dB.

6.1.2 Robustness against Attacks

Tables 6.3 and 6.4 present a comparative analysis of robustness results of both FPSUM-HE and FPSUM-PD for the audio file "LoopyMusic" and the video file "Dragon", respectively.

TABLE 6.3: Comparison of BER and NC values of an audio file

Attacks	**Parameters**	**FPSUM-HE**		**FPSUM-PD**	
		BER	**NC**	**BER**	**NC**
Re-quantization	16-8-16 bits	0.07	0.951	0.00	1.000
Re-sampling	44.1-22.05-44.1 kHz	0.11	0.902	0.00	1.000
MP3 Compression	256 kbps	0.09	0.912	0.03	0.979
AWGN	18 dB	0.13	0.882	0.08	0.925

TABLE 6.4: Comparison of BER and NC values of a video file

Attacks	**Parameters**	**FPSUM-HE**		**FPSUM-PD**	
		BER	**NC**	**BER**	**NC**
Median Filter	$[3 \times 3]$	0.09	0.912	0.05	0.966
Re-sizing	$320 - 640 - 320$ pixels	0.06	0.972	0.01	0.999
H.264 Compression	768 kbps	0.09	0.912	0.00	1.000
AWGN	20 dB	0.14	0.856	0.02	0.985

It is evident, from Table 6.3 and 6.4, that robustness results of both audio and video files in FPSUM-PD are better than BER and NC values obtained in FPSUM-HE. The lower BER and NC values obtained in FPSUM-HE are attributed to quantization and rounding noise generated due to the encryption and decryption of the approximation coefficients of the content that in turn affects the robustness of the fingerprint. In FPSUM-HE, if a fingerprint is embedded more than once into the approximation coefficients rather than embedding in selected coefficients, the resulting BER and NC values would be worst. Consequently, the noise introduced by the quantization and encryption affects the robustness of the fingerprint as well as the imperceptibility.

6.1.3 Security against Collusion-attacks

Table 6.5 shows the number of colluders U which are successfully traced through Nuida et al. (2007) codes tracing Algorithm 3 (cf. Section 4.5.1.5) in both FPSUM-HE and FPSUM-PD.

TABLE 6.5: Security against collusion attacks

No. of Colluders	**No. of Colluders Detected for Attacks in FPSUM-HE**				**No. of Colluders Detected for Attacks in FPSUM-PD**			
U	**Average**	**Minimum**	**Maximum**	**Median**	**Average**	**Minimum**	**Maximum**	**Median**
2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4
5	5	4	4	5	5	5	5	5

It can be seen, from Table 6.5, that in most of the cases in FPSUM-HE, the colluders are successfully traced, whereas in FPSUM-PD all the colluders are successfully traced in all cases.

6.1.4 Computational and Communicational Costs

Table 6.6 shows the computation time of three audio and three video files for both FPSUM-HE and FPSUM-PD. The computation time of FPSUM-HE and FPSUM-PD includes the time taken to generate a fingerprint and base and supplementary files for audio and video files.

The columns in Table 6.6 containing the total CPU time show that the computational cost

TABLE 6.6: Comparison of computation time

CPU Time in secs for FPSUM-HE						
Process	**Loopy Music**	**Hugewav**	**Aasan Nai Yahan**	**Traffic**	**Dragon**	**Breaking Bad**
Fingerprint Generation	6.01	6.01	6.01	6.01	6.01	6.01
Base File Generation	14.08	31.15	181.39	10.77	68.22	70.04
Supplementary File Generation	0.034	0.180	1.196	7.22	24.16	36.15
Total Time	20.13	37.34	188.60	24.00	98.40	112.20
CPU Time in secs for FPSUM-PD						
Fingerprint Generation	6.01	6.01	6.01	6.01	6.01	6.01
Base File Generation	1.24	2.12	17.99	3.50	32.99	66.03
Supplementary File Generation	0.034	0.180	1.196	7.22	24.16	36.15
Total Time	7.29	8.31	25.20	16.73	63.16	108.19

of FPSUM-PD is comparatively shorter than that of FPSUM-HE. Since the execution time of a

fingerprint and supplementary file generation are constant in both FPSUM-HE and FPSUM-PD, the difference lies in the columns containing the execution time of the base file generation. It is evident from the results of base file generation row in Table 6.6 that FPSUM-PD outperforms FPSUM-HE in terms of computational costs. The time taken to generate the base files for three audio and three video files in FPSUM-PD is comparatively smaller than the execution time of base file generation protocol in FPSUM-HE. In Table 6.7, the communication time of the audio file "LoopyMusic" and the video file "Traffic" are compared for both FPSUM-HE and FPSUM-PD. The communication time includes the time taken to distribute the base and supplementary files and reconstruct the original file at the buyer's end.

TABLE 6.7: Comparison of communication time

Communication Time in secs for FPSUM-HE				
File Name	**Base file Delivery**	**Supplementary File Delivery**	**File Reconstruction**	**Total Distribution Time**
Loopy Music	8.01	10.00	3.89	21.90
Breaking Bad	184.00	657.29	595.05	1436.34
Communication Time in secs for FPSUM-PD				
Loopy Music	5.58	10.00	3.09	18.67
Breaking Bad	67.40	657.29	295.00	1019.69

It can be seen, in the last column of Table 6.7, that the communicational cost of FPSUM-PD is comparatively shorter than that of FPSUM-HE. In FPSUM-HE and FPSUM-PD, the sizes of the base files of "LoopyMusic" are 0.52 MB and 0.11 MB, respectively. The increased size of the base file in FPSUM-HE is due to use of the 1024-bits Paillier encryption on the selected approximation coefficients. On the other hand, in FPSUM-PD, the idea of performing the permutation and AES-128 symmetric encryption on the pre-computed approximation coefficients results in the generation of a small-sized base file. On considering the increase in the original file size, it can be said that large-sized multimedia files can be more efficiently delivered to the buyer in FPSUM-PD compared to FPSUM-HE. For example, the original size of audio files "Hugewav" and "Aasan Nai Yahan" are 2.97 MB and 36.01 MB (cf. Table 4.3), the size of the base files of these audio files in FPSUM-HE are 0.88 MB and 9.80 MB, and in FPSUM-PD, are 0.29 MB and 3.58 MB, respectively. On comparing the base file sizes of "Hugewav" and "Aasan Nai Yahan" in FPSUM-PD and FPSUM-HE, it can be seen that the size of the base file in FPSUM-HE increases three fold compared to FPSUM-PD. In addition to base file delivery time, the communication time required in reconstruction of the file at a user end in FPSUM-HE is comparatively larger than FPSUM-PD due to the 1024-bit Paillier decryption of the base file.

Thus, it can be said, from the results of Tables 6.6 and 6.7, that the performance of FPSUM-PD is better than FPSUM-HE in terms of computational and communicational costs.

6.1.5 Cryptographic Overhead

Table 6.8 shows the CPU execution time of each cryptographic block used in FPSUM-HE and FPSUM-PD for the audio file "LoopyMusic" and the video file "Traffic".

TABLE 6.8: Comparison of cryptographic costs

Cryptographic Algorithms	**Audio File**		**Video File**	
	FPSUM-HE	**FPSUM-PD**	**FPSUM-HE**	**FPSUM-PD**
	CPU Time in secs			
Public-key cryptography	5.73	0.72	8.80	0.72
Anonymous Key Exchange	9.62	9.62	9.62	9.62
AES Encryption/Decryption	0.11	1.89	2.83	0.98
Total	17.24	13.17	18.53	11.32

It is evident, from Table 6.8, that the cryptographic costs of FPSUM-HE are relatively larger than those of FPSUM-PD for both audio and video files. The 1024-bit public-key encryption of a base file contributes to the high cryptographic costs of FPSUM-HE. The lower cryptographic overhead of FPSUM-PD is due to use of the AES-128 symmetric-key algorithm to encrypt the pre-computed base files of multimedia files instead of using the 1024-bit Paillier encryption to produce the fingerprinted base file. The cryptographic overheads due to the anonymous AKE protocol are constant in all cases since the number of tail nodes and onion paths between two communicating peers are considered fixed in both systems. Furthermore, the same supplementary file distribution protocol is implemented in both FPSUM-HE and FPSUM-PD, thus the difference lies in how the cryptographic overhead is reduced in the base file distribution protocol. Thus, it can be said that FPSUM-PD provides a more efficient solution in terms of cryptographic costs.

6.2 Comparative Analysis of FPSUM-HE & FPSUM-PD with P2P Content Distribution systems

This section carries out a comparative analysis of the proposed FPSUM-HE and FPSUM-PD schemes with the systems discussed in Chapter 3. The comparison focuses on the guaranteed

security and privacy properties described in Section 3.5.3 The results of the comparative analysis are given in Tables 6.9 and 6.10. In these tables, a cell contains "No" when a security or a privacy property is not guaranteed by the P2P content distribution system.

In this analysis, 16 P2P content distribution systems are compared with the proposed frameworks (FPSUM-HE and FPSUM-PD) in terms of content protection (copyright protection, copy prevention, traceability), privacy (data, user), revocable privacy, and robustness and security against attacks (signal processing, collusion/malicious, communication).

6.2.1 Content Protection

Content protection incorporates basic security properties such as copyright protection (association of digital rights with the content by embedding meta-data or watermark into the content), conditional access (no additional replication other than the permitted copies) and traceability (ability to trace and identify the copyright violator).

- FPSUM-HE, FPSUM-PD, Megías (2014), Megías and Domingo-Ferrer (2014), Domingo-Ferrer and Megías (2013), Tsolis et al. (2011), Stenborg et al. (2011), J. S. Li et al. (2010), X. Li et al. (2010), and Gao et al. (2010) guarantee copyright protection by using digital watermarking or fingerprinting techniques. The remaining P2P content distribution systems do not offer copyright protection.

- The systems proposed by Inamura and Iwamura (2014), Win and Emmanuel (2011), J. S. Li et al. (2010), Y. Y. Chen et al. (2009), and M. K. Sun et al. (2009) guarantee conditional access by using DRM techniques. In the other P2P content distribution systems, DRM techniques are not used, thus copy prevention property cannot be guaranteed.

- FPSUM-HE, FPSUM-PD, Megías (2014), Megías and Domingo-Ferrer (2014), Domingo-Ferrer and Megías (2013), Win and Emmanuel (2011), J. S. Li et al. (2010), X. Li et al. (2010), and Gao et al. (2010) guarantee the traceability of copyright violators by using digital fingerprinting techniques. The systems proposed by Tsolis et al. (2011) and Stenborg et al. (2011) use digital watermarking, thus the traceability property cannot be guaranteed. Similarly,the systems proposed by Inamura and Iwamura (2014), Y. Y. Chen et al. (2009), and M. K. Sun et al. (2009) offer copy prevention but fail to provide traceability. The remaining systems do not offer traceability.

TABLE 6.9: Comparison of P2P systems based on guaranteed security and privacy properties

P2P Systems	Content Protection			Privacy		Revocable Privacy
	Copyright Protection	Copy Prevention	Traceability	User	Data	
FPSUM-HE	Yes due to fingerprinting	No	Yes	Yes, due to pseudonymity and anonymous communications	Yes, due to encryption	Yes
FPSUM-PD	Yes, due to fingerprinting	No	Yes	Yes, due to pseudonymity and anonymous communications	Yes, due to encryption	Yes
Megías (2014)	Yes, due to fingerprinting	No	Yes	Yes, due to pseudonymity and anonymous communications	Yes, due to encryption	Yes
Megías & Domingo-Ferrer (2014)	Yes, due to fingerprinting	No	Yes	Yes, due to pseudonymity and anonymous communications	Yes, due to encryption	Yes
Inamura & Iwamura (2014)	No	Yes, due to DRM	No	Yes, due to anonymous communications	Yes, due to encryption	No
Domingo-Ferrer & Megías (2013)	Yes, due to fingerprinting	No	Yes	Yes, due to group signatures	Yes, due to encryption	Yes
Yu et al. (2011)	No	No	No	Yes, due to anonymous communications	Yes, due to encryption	No
Win et al. (2011)	No	Yes, due to DRM	Yes	Yes, due to pseudonymity and blind decryption	Yes due to encryption	Yes
Tsolis et al. (2011)	Yes, due to watermarking	No	No	No	Yes, due to encryption	No
Stenborg et al. (2011)	Yes, due to watermarking	No	No	No	Yes, due to encryption	No
Li et al. (2010)	Yes, due to fingerprinting	No	Yes	No	No	No
Gao at al. (2010)	Yes, due to fingerprinting	No	Yes	No	Yes, due to encryption	No
Li et al. (2010)	Yes, due to fingerprinting	Yes, due to DRM	Yes	No	Yes, due to encryption	No
Chen et al. (2009)	No	Yes, due to DRM	No	No	Yes, due to encryption	No
Sun et al. (2009)	No	Yes, due to DRM	No	Yes, due to anonymous authentication	Yes, due to encryption	No
Lu et al. (2007)	No	No	No	Yes, due to pseudonymity and anonymous communications	Yes, due to encryption	No
Sherwood et al. (2002)	No	No	No	Yes, due to anonymous communications	Yes, due to encryption	No

TABLE 6.10: Comparison of P2P systems based on guaranteed security and privacy properties

P2P Systems	Robustness and Security against Attacks			
	Signal Processing Attacks	Collusion Attacks		Communication Attacks
		Content Protection Systems	Privacy Protection Systems	
FPSUM-HE	Yes	Yes	Yes	Yes
FPSUM-PD	Yes	Yes	Yes	Yes
Megías (2014)	Yes	Yes	Yes	Yes
Megías & Domingo-Ferrer (2014)	Yes	Yes	Yes	Yes
Inamura and Iwamura (2014)	No	No	No	No
Domingo-Ferrer & Megías (2013)	Yes	Yes	Yes	No
Yu et al. (2011)	No	No	Yes	Yes
Win et al. (2011)	No	No	Yes	No
Tsolis et al. (2011)	No	No	No	No
Stenborg et al. (2011)	Yes	Yes	No	No
Li et al. (2010)	Yes	Yes	No	No
Gao et al. (2010)	Yes	Yes	No	No
Li et al. (2010)	Yes	No	No	No
Chen et al. (2009)	No	Yes	No	Yes
Sun et al. (2009)	No	Yes	No	No
Lu et al. (2007)	No	No	Yes	Yes
Sherwood et al. (2002)	No	No	Yes	Yes

6.2.2 Privacy

Privacy property incorporates user privacy (protection of user-related information) and data privacy (protection of data against unauthorized entities).

- FPSUM-HE, FPSUM-PD, Megías (2014), Megías and Domingo-Ferrer (2014), Yu et al. (2011), Lu et al. (2007), and Sherwood et al. (2002) guarantee mutual anonymity to the users of P2P system due to pseudonymity and anonymous communication techniques. A system proposed by Inamura and Iwamura (2014) guarantees user privacy by considering

an anonymous channel for a communication between the users of the system. Domingo-Ferrer and Megías (2013), Win and Emmanuel (2011), and M. K. Sun et al. (2009) use anonymous authentication techniques to provide anonymity to the users.

- FPSUM-HE, FPSUM-PD and all the P2P content distribution systems, except the system proposed by X. Li et al. (2010), guarantee data protection from unauthorized access and manipulation due to use of symmetric/asymmetric/hybrid encryption techniques.

6.2.3 Revocable Privacy

Revocable privacy implies that a user can enjoy full anonymity unless he/she violates a pre-defined set of rules of the system.

FPSUM-HE, FPSUM-PD, Megías (2014), Megías and Domingo-Ferrer (2014), Domingo-Ferrer and Megías (2013) and Win and Emmanuel (2011) guarantee revocable privacy. In these systems, the real identity of the users are only revealed by the trusted third party, i.e. the registration authority, in case a user is found guilty of copyright violation. The system proposed by M. K. Sun et al. (2009), in spite of employing an anonymous authentication technique to provide anonymity, does not offer traceability due to use of untraceable blind signatures. The remaining systems either provide full anonymity or no anonymity at all to the users.

6.2.4 Robustness and Security against Attacks

This property incorporates robustness (resistance against common signal processing attacks) and security (resistance against collusion, malicious and communication attacks).

- The watermarking schemes either employed by FPSUM-HE, FPSUM-PD, Megías (2014), Megías and Domingo-Ferrer (2014), Domingo-Ferrer and Megías (2013), Stenborg et al. (2011), J. S. Li et al. (2010), or proposed by X. Li et al. (2010) and Gao et al. (2010), are robust against common signal processing attacks such that the extracted information from the attacked content resembles the original watermark/fingerprint. Tsolis et al. (2011) claim that their proposed watermarking technique is robust enough to facilitate copyright protection and management of the digital images, but no proof-of-concept is provided to show the robustness of the scheme against common signal processing attacks.

- In FPSUM-HE, FPSUM-PD and the content protection P2P systems proposed by Megías (2014), Megías and Domingo-Ferrer (2014), Domingo-Ferrer and Megías (2013), Stenborg et al. (2011), J. S. Li et al. (2010), Gao et al. (2010), the security against collusion attacks is guaranteed due to use of collusion-resistant fingerprinting. The systems of Y. Y. Chen et al. (2009) and M. K. Sun et al. (2009) offer security against collusion attacks by using DRM-enabled application and content-key management protocol.

- In the privacy protection P2P systems proposed by Megías (2014), Megías and Domingo-Ferrer (2014) and the proposed frameworks (FPSUM-HE, FPSUM-PD), the privacy of the users are preserved against malicious attacks due to anonymous fingerprinting. In the proposed systems of Win and Emmanuel (2011) and Yu et al. (2011), the attempts to de-anonymize the users are prevented by using anonymous token sets, and random walking and flooding techniques, respectively.

- The communication channel used for transferring the data between two users of FPSUM-HE, FPSUM-PD, and the systems of Megías (2014), Megías and Domingo-Ferrer (2014), Yu et al. (2011), Y. Y. Chen et al. (2009), Lu et al. (2007), Sherwood et al. (2002), are protected against malicious attacks such as man-in-the-middle attacks, denial of service and replay attacks. In other systems, the protection against communication attacks is either not provided or not discussed.

It is apparent from Tables 6.7 and 6.8 that most of the presented content distribution systems either focus on content or privacy protection. Except for the systems by Megías (2014), Megías and Domingo-Ferrer (2014), FPSUM-HE and FPSUM-PD, all other systems fail to provide the guaranteed security and privacy properties simultaneously. Though the systems proposed by Megías (2014) and Megías and Domingo-Ferrer (2014) provide guaranteed security and privacy properties, these systems require a two-layer anti-collusion code (segment level and fingerprint level), which results in a longer codeword. Furthermore, in both systems, the construction of a valid fingerprint at a child buyer's end requires a communication channel between the P2P proxies that are carrying the content from at least two parent buyers to the child buyer. These proxy peers communicate with each other in the presence of a transaction monitor, who verifies that the constructed fingerprint at a child's end is valid and, thus, can be used for the identification purpose in the tracing protocol.

In the system of Megías and Domingo-Ferrer (2014), honest and committed proxies are required for the generation of valid fingerprints as compared to the proposed frameworks (FPSUM-HE and FPSUM-PD), where only an honest monitor is required for the fingerprint generation. However, in the improved version proposed by Megías (2014), malicious proxies are considered in the fingerprinting protocol. The proposed four-party anonymous fingerprinting protocol is designed in a way to prevent the malicious proxies to access clear-text fingerprinted content. However, the proposed system requires a two-layer anti-collusion code as compared to the proposed frameworks (FPSUM-HE and FPSUM-PD), where smaller codewords are generated by the monitor.

6.3 Conclusions

In this chapter, the proposed secure and privacy-preserving content distribution frameworks are first compared with each other in terms of efficiency. Then, both frameworks are compared with the P2P content distribution systems presented in Chapter 3 in terms of functionalities.

First, FPSUM-HE and FPSUM-PD are compared with each other to demonstrate the performance improvement in FPSUM-PD. In comparison to FPSUM-HE, the asymmetric fingerprinting protocol in FPSUM-PD is designed so as to achieve computational and communicational efficiency as well as desired security and privacy properties. The comparative analysis presented in Section 6.1 shows the fact that the content protection and privacy-preserving techniques can be integrated together without a need of too demanding cryptographic protocols.

Then, FPSUM-HE and FPSUM-PD are compared with P2P content distribution systems. The comparative analyses provided in Tables 6.7 and 6.8 show that the proposed frameworks differ from the existing P2P content distribution systems. Most of the P2P systems focused on either providing a copyright protection to content owners or privacy to end users, whereas both frameworks are proposed for P2P based content distribution focusing on copyright protection and privacy simultaneously.

Chapter 7

Conclusions and Future Work

An explosive growth in information technologies in the last few decades has given birth to seemingly limitless channels for the exchange of text, audio, video, graphics and software data. This phenomenon has consequently resulted in the need for efficient content distribution channels. P2P is one such solution. P2P is a newly emerged paradigm, having shared pool resourcing as its fundamental characteristic. It utilizes the resources of the end users such as content, CPU cycles, storage and bandwidth. It is decentralized and hence, offers autonomy to end users. It is cost-efficient, scalable and fault tolerant. Its market continues to grow, and according to an estimate, about 60% of the internet traffic is routed through this mechanism. However, P2P's core advantageous characteristics also pose most serious challenges. Its open nature of operation results in lack of security and privacy, exposing it to unregulated and uncontrollable copyrighting and distribution, loss of content ownership, and thus, posing serious threats to the users (buyers) as well as the merchants.

Despite the fact that many content distribution systems can be found in the literature, most of them are not yet able to provide security, fairness and dissuading malicious activities. A thorough examination of the existing state-of-the-art P2P content distribution mechanisms was carried out in the first part of the thesis, looking deep into content as well as privacy protection techniques. A comprehensive comparative analysis of these mechanisms shows that some systems offer content protection through traceability, copyright protection, copy prevention etc., and provide robustness and security against signal processing and collusion attacks. Within the category of these systems, the security is achieved at a cost of cryptographic protocols that require intensive computations. The other category of P2P content distribution systems focus on privacy preservation of the end users. These systems address either sender/receiver anonymity

or mutual anonymity, and offer resistance against malicious and communication attacks. Only three of the examined systems provide both security and privacy, the remaining examined P2P systems do not provide security and privacy properties simultaneously.

In this context, the main goal of this thesis is to integrate privacy and security properties in a single framework, while keeping the computational and communication loads low. The ensuing research effort has resulted in the design and development of a framework that addresses the privacy and security concerns simultaneously. The framework is named as FPSUM (Framework for preserving Privacy and Security of User and Merchant). Two variations of FPSUM are proposed: FPSUM based on Homomorphic Encryption (FPSUM-HE), and FPSUM using Proxy-based Distribution (FPSUM-PD).

Section 7.1 presents conclusions of the main contributions of this thesis. Following this, Section 7.2 presents ideas and directions for future work.

7.1 Conclusions

This section resumes the main contributions of this work: FPSUM-HE (cf. Section 7.1.1 and FPSUM-PD (cf. Section 7.1.2).

7.1.1 FPSUM-HE

Motivated by the necessity of guaranteeing copyright and privacy protection simultaneously, the first main contribution of this work aimed at proposing a secure and privacy-preserving content distribution framework for P2P systems. The first task in the integration of the content and privacy protection techniques in P2P systems was to partition the original multimedia file into two parts: a small-sized part (base file) that carries the most significant information of the content, and a large-sized part (supplementary file) that is unusable without the base file. The small-sized file is used for embedding the collusion-resistant fingerprint, and thus needs to be distributed from the merchant to the buyer of the P2P system. This part is achieved through a fingerprinting protocol between a merchant, a buyer and a trusted third party. The large-sized file does not contain any important information or a fingerprint, thus it can be easily distributed within a network of P2P buyers without any fear of copyright violation. To accomplish this task of distributing two files simultaneously in a P2P environment, a hybrid P2P system with privacy-preserving properties is used.

To counter the challenging task of resolving the conflicting interest between the protection of the content, such as the merchants' need for copyright protection and traceability for the copyright violator, and privacy protection, such as anonymity, frameproofness, and unlinkability of online activities of buyers, an asymmetric fingerprinting protocol is proposed. The fingerprinting protocol is based on homomorphic encryption and collusion-resistant fingerprinting. A trusted third party (monitor) is used to generate the collusion-resistant fingerprinting codes to prevent the customer's right problem. These fingerprint codes are embedded by the merchant into the content so as to identify an illegal re-distributor(s) from an unlawfully re-distributed content. The fingerprinting protocol is performed by the merchant and the buyer in the presence of a trusted party in such a way that the merchant does not know the fingerprint and the fingerprinted content, while the buyer receives the fingerprinted content with his/her unique identity. Moreover, the buyers can obtain their digital contents anonymously, but this anonymity can be revoked as soon as they are found guilty of copyright violation. The proposed fingerprinting protocol fulfils the guaranteed security properties (cf. Section 3.5.3) simultaneously. In addition, a security analysis of the protocol proves that it fulfils the desired security properties. The implementation of the aforementioned asymmetric fingerprinting protocols has also been evaluated to calculate the overheads. The implementation combines the basic existing cryptographic tools with the embedding of a fingerprint into an encrypted domain, allowing reduction of both the computational overhead and the need for a communication bandwidth.

A hybrid P2P system incorporating a privacy-preserving mechanism is employed to facilitate the distribution of both base and supplementary files in a secure and anonymous fashion. The base file is distributed in a centralized manner from the merchant to the buyer. To provide an anonymous transfer of the supplementary file from the super peer or a peer to another peer, a supplementary distribution protocol is proposed that incorporates latent pseudonymity, anonymous communication and symmetric encryption. The proposed distribution protocol fulfils the guaranteed user and data privacy properties (defined in Chapter 3). Furthermore, a security analysis of the protocol proves that this proposal is secure against the considered communication and de-anonymization attacks.

The implementation of a system combining a fingerprinting protocol for base file distribution, basic cryptographic tools and an anonymous supplementary file distribution protocol demonstrates that the fulfilment of security requirements, such as accountability, traceability, and integrity, is compatible with the provision of privacy guarantees. In FPSUM-HE, the novelty is to show how to merge content protection and privacy-preserving mechanisms simultaneously in a P2P environment. The deployment of security and privacy enhancing solutions is usually

at a price of lowering the performance efficiency of the system. However, from the simulations results' of FPSUM-HE, it is shown that overhead for running the FPSUM-HE is moderate, keeping in mind that secure and private systems tend to be more complex and costly than those with zero or a lesser degree of security and privacy.

7.1.2 FPSUM-PD

The second main contribution of this dissertation aimed at improving the efficiency of the base file distribution protocol of FPSUM-HE. Since, in FPSUM-HE, the use of homomorphic encryption affects the efficiency and robustness of the fingerprinting scheme, the fingerprinting protocol of FPSUM-PD is designed in such a way that it provides (1) piracy tracing, (2) collusion resistance, (3) buyer security and privacy preservation, (4) efficient content distribution, and (5) excellent robustness against signal processing attacks. In contrast to FPSUM-HE and earlier works in asymmetric fingerprinting protocols, FPSUM-PD achieves an efficient asymmetric fingerprinting scheme by avoiding multi-party security protocols, bit commitments and public-key cryptography of the content. The proposed asymmetric fingerprinting protocol based on collusion-resistant codes and a secure embedding scheme is performed between a merchant, a buyer and a set of P2P proxies in the presence of a trusted third party (monitor). The base file is distributed to the buyer through proxies in such a way that the merchant cannot predict about the fingerprinted content, and the proxies are unable to frame honest buyers by combining their information bits. A formal security analysis of the protocol validates the security guarantees described in Chapter 3. The implementation of the protocol shows that the overhead of the scheme is comparatively lower than that of FPSUM-HE. A performance analysis between FPSUM-HE and FPSUM-PD shows that the latter offers significant performance improvement and provides a relatively higher computational and communicational efficiency.

As expected, FPSUM-PD is comparatively costly as compared to direct delivery without security and privacy. However, the additional costs (due to the cryptographic and fingerprinting protocols) are kept to a minimum and are acceptable compared to those of FPSUM-HE. Thus, considering the computational and network capacity of modern systems, the results of the performance analysis suggest that the proposed framework can be practically implemented or easily incorporated into real-time P2P content distribution applications.

Both proposed techniques of FPSUM offer an integration of copyright protection to the content owners and privacy to the end users. This combination of characteristics makes the proposed framework superior than all the existing state-of-the-art P2P content distribution systems.

In conclusion, both variants of FPSUM are not permanent one-off solutions but these can be improved further to keep up with the advances in technology and the attackers' skills.

7.2 Future Work

There are four different areas where this dissertation leaves room for future work: (1) real-time experimentation, (2) real-world P2P application, (3) design improvements and (4) implementation in constrained computing environment.

- **Experimentation:** The results presented in this study are mainly obtained through computer-based simulations. Thus, one of the further works may be the experimentation on the real-world testbeds such as PlanetLab or OpenLab. These testbeds provide global platforms for deploying and evaluating network services. For example, PlanetLab has been used to evaluate a diverse set of planetary-scale network services, including content distribution networks, file sharing, network measurement and analysis, and anomaly detection. An experimentation on the testbed is necessary for taking into account the communication delays between the peers, scalability issues and the traffic overheads caused by the privacy protection mechanisms.

- **Implementation:** Two P2P-based content distribution systems are proposed in this dissertation. Both systems are evaluated as standalone applications through controlled simulations. Thus, a possible future work can be their implementation in real P2P applications, e.g. as a plug-in that enables the copyright and privacy protection features in a P2P system. Moreover, some metrics at the application layer, such as bandwidth and latency, can be measured with the real-time implementation in order to evaluate the impact of the proposed content distribution protocols on the user experience.

- **Design Improvement:** FPSUM-HE and FPSUM-PD employ the file partitioning concept to deliver a small-sized base file in a centralized manner to the buyer from the merchant, and distribute a large-sized supplementary file in a P2P fashion. Indeed, by the simulation and experimental results presented in Chapters 4 and 5, it is evident that the proposed approach has considerably reduced the computational and communicational load of the merchant. However, in FPSUM-HE, some approximation coefficients (equal to the size of the fingerprint) are selected to embed the collusion-resistant fingerprint only once in order to prevent the generation of a large-sized base file. Thus, a future work can be a reduction

of the approximation coefficients through compression techniques. Lossless compression techniques such as Limpel-Ziv-Welch (LZW) and Free Lossless Audio Codec (FLAC) may be applied to both video and audio approximation coefficients, respectively, without affecting the quality of the files. These compressed approximation coefficients can then allow the merchant to embed the collusion-resistant fingerprint more than once into the content without worrying about the increase in the size of the file.

- **Performance Evaluation in Constrained Computing Environments:** In this dissertation, the performance of the protocols proposed in FPSUM-HE and FPSUM-PD is not considered in constrained computing environment, e.g. a device that has only small memory size and computing power. In such computing systems, the execution of the protocol proposed in FPSUM-HE that uses fingerprinting in the encrypted domain based on asymmetric homomorphic encryption to provide content and privacy protection is likely to be too expensive in terms of computations and bandwidth. An alternative, such as the protocol described in FPSUM-PD that deploys symmetric building blocks, could be evaluated in low-powered devices.

P2P as a content distribution paradigm will remain the major contributor in an overall Internet traffic in future. The research effort aiming at the integration of content and privacy protection mechanisms and an efficient implementation of both mechanisms in P2P infrastructure is definitely needed.

References

Akamai. (1998). Retrieved from https://www.akamai.com/ (Last accessed on September 03, 2014)

Akbarinia, R., Pacitti, E., & Valduriez, P. (2006). Reducing network traffic in unstructured p2p systems using top-k queries. *Distributed and Parallel Databases*, *19*(2-3), 67-86.

Alon, N., Fischer, E., & Szegedy, M. (2001). Parent-identifying codes. *Journal of Combinatorial Theory*, *95*(2), 349-359.

Alstrup, S., & Rauhe, T. (2005). Introducing octoshape: A new technology for large-scale streaming over the internet. *EBU Technical Review*, *303*, 1-10.

Anthapadmanabhan, N. P., & Barg, A. (2009). Two-level fingerprinting codes. In *Ieee international symposium on information theory, 2009* (p. 2261-2265).

Apple's fairplay drm copy protection. (2001). Retrieved from http://remove-drm.blogspot.com.es/2014/01/apples-fairplay-drm-copy-protection.html (Last accessed on September 03, 2014)

Arnold, M., Schmucker, M., & Wolthusen, S. D. (2003). *Techniques and applications of digital watermarking and content protection* (2nd ed.). Artech House Publishers, Inc.

Arsenova, E. (2002). Technical aspects of digital rights management.. Retrieved from http://wob.iai.uni-bonn.de/Wob/images/01212504.pdf (Last accessed on September 03, 2014)

Barg, A., & Kabatiansky, G. (2013). Robust parent identifying codes and combinatorial arrays. *IEEE Transactions on Information Theory*, *59*(2), 994 -1003.

Barni, M., & Bartolini, F. (2004). *Watermarking systems engineering: Enabling digital assets security and other applications* (1st ed.). CRC Press.

Bassia, P., Pitas, I., & Nikolaidis, N. (2001). Robust audio watermarking in the time domain. *IEEE Transactions on Multimedia*, *3*(2), 232-241.

Bellovin, S. M., & Merrit, M. (1992). Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *Proceedings of the ieee symposium on research in security and privacy* (p. 72-84). IEEE Computer society.

Bender, W., Gruhl, D., Morimoto, N., & Lu, A. (1996). Techniques for data hiding. *IBM Systems Journal*, *35*, 313-316.

Bhat, K. V., Sengupta, I., & Das, A. (2011). A new audio watermarking scheme based on singular value decomposition and quantization. *Multimedia Tools and Applications*, *52*(2-3), 369-383.

Biehl, I., & Meyer, B. (2002). Cryptographic methods for collusion-secure fingerprinting of digital data. *Computers and Electrical Engineering*, *28*, 59-75.

Bittorrent. (2000). Retrieved from http://www.bittorrent.com/ (Last accessed September 03, 2014)

Blackburn, S. R. (2002). An upper bound on size of a code with the k-identifiable parent property. *Journal of Combinatorial Theory, pages*, *102*(1), 179-185.

Blaze, M. (1999). Using the keynote trust management system. AT&T Research Labs. Retrieved from www.cs.columbia.edu/angelos/keynote.html/ (Last accessed on September 03, 2014)

Blaze, M., Feigenbaum, J., & Lacy, J. (1996). Decentralized trust management. In *Proceedings of the ieee symposium on security and privacy* (p. 164-173). IEEE.

Boneh, D., & Shaw, J. (1999). Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, *44*(5), 1897-1905.

Bracciale, L., Lo Piccolo, F., Luzzi, D., Salsano, S., Bianchi, G., & Blefari-Melazzi, N. (2008). A push-based scheduling algorithm for large scale p2p live streaming. In *Proceddings of the 4th international telecommunication networking workshop on qos in multiservice ip networks* (p. 1-7).

Camenisch, J. (2000). Anonymous fingerprinting with group signatures. In *Asiacrypt 2000* (Vol. 1976, pp. 415–428). Springer.

Castro, M., Druschel, P., Kermarrec, A., Nandi, A., Rowstron, A., & Singh, A. (2003). Splitstream: High-bandwidth multicast in cooperative environments. In *Proceedings of the19th acm symposium on operating systems principles* (p. 298-313).

Castro, M., Druschel, P., Kermarrec, A., & Rowstron, A. (2002). Scribe: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in communications*, *20*(8), 1489-1499.

Charpentier, A., Fontaine, C., Furon, T., & Cox, I. (2011). An asymmetric fingerprinting scheme based on tardos codes. In *Proceedings of the 13th international conference on information hiding* (p. 43-58). Springer.

Chaum, D. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Communication of the ACM*, *24*(2), 84-90.

Chaum, D. (1988). The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, *1*(1), 65-75.

Chaum, D., & Van-Heyst, E. (1991). Group signatures. In *Proceedings of the 10th annual international conference on theory and application of cryptographic techniques* (p. 257-265). Springer.

Chawathe, Y., Ratnasamy, S., & Breslau, L. (2003). Gia: Making gnutella-like p2p systems scalable. In *Proceedings of acm sigcomm* (p. 407-418).

Chen, B., & Wornell, G. W. (2001). Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, *47*(4), 1423-1443.

Chen, C. L. (2008). A secure and traceable e-drm system based on mobile device. *Expert Systems with Applications*, *35*(3), 878-886.

Chen, Y. Y., Jan, J. K., Chi, Y. Y., & Tsai, M. L. (2009). A feasible drm mechanism for bt-like p2p system. In *Proceedings of international symposium on information engineering and electronic commerce* (p. 323-327).

Cheng, Y., Liu, Q., Zhu, X., Zhao, C., & Li, S. (2011). Research on digital content protection technology for video and audio based on ffmpeg. *International Journal of Advancements in Computing Technology*, *3*(8), 9-17.

Choi, J. G., Sakurai, K., & Park, J. H. (2003). Does it need trusted third party? design of buyer-seller watermarking protocol without trusted third party. In *Applied cryptography and network security* (Vol. 2846, pp. 265–279). Springer.

Cholvi, V., Felber, P., & Biersack, E. (2004). Efficient search in unstructured peer-to-peer networks. *European Transactions on Telecommunications*, *15*(6), 535-548.

Chor, B., Fiat, A., Naor, M., & Pinkas, B. (1994). Tracing traitors. In (Vol. 46, p. 893-910). IEEE.

Clarke, I., Miller, S. G., Hong, T. W., Sandberg, O., & Wiley, B. (2002). Protecting free expression online with freenet. *IEEE Internet Computing*, *6*(1), 40-49.

Clarke, R. (1998). Roger clarke's dataveillance and information privacy home-page.. Retrieved from http://www.rogerclarke.com/DV/ (Last accessed on September 03, 2014)

Cox, I. J., Kilian, J., Leighton, T., & Shamoon, T. (1997). Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, *6*(12), 1673-1687.

Cox, I. J., & Linnartz, J. P. M. G. (1998). Some general methods for tampering with watermarks. *IEEE Journal on Selected Areas in Communications*, *16*(4), 587-593.

Cox, I. J., Miller, M., Bloom, J., Fridrich, J., & Kalker, T. (2007). *Digital watermarking and steganography* (2nd ed.). Morgan-Kaufmann.

Css demystified. (1999). Retrieved from http://cs.stanford.edu/people/eroberts/cs201/projects/1999-00/dmca-2k/css.html (Last accessed on September 23, 2014)

Cuenca-Acuna, F. M., Martin, R. P., & Nguyen, T. D. (2003). Autonomous replication for high availability in unstructured p2p systems. In *Proceedings of the symposium on reliable distributed systems* (p. 99-108). IEEE.

Damiani, E., Vimercati, S. D. C. D., Paraboschi, S., Samarati, P., & Violante, F. (2002). A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proceedings of the 9th acm conference on computer and communications security* (p. 207-216). ACM.

Deng, M., & Preneel, B. (2008). On secure and anonymous buyer-seller watermarking protocol. In *Proceedings of third international conference on internet and web applications and services* (p. 524-529). IEEE.

Despotovic, Z., & Aberer, K. (2006). P2p reputation management: Probabilistic estimation vs. social networks. *Computer Networks*, *50*(4), 485-500.

Dittmann, J., Behe, A., Stabenau, M., Schmitt, P., Schwenk, J., & Ueberberg, J. (1999). Combining digital watermarks and collusion secure fingerprints for digital images. In *Proceedings of spie on electronic imaging* (Vol. 3657, p. 171-182).

Domingo-Ferrer, J., & Megías, D. (2013). Distributed multicast of fingerprinted content based on a rational peer-to-peer community. *Computer Communications*, *36*(5), 542-550.

ebay. (1995). Retrieved from http://www.ebay.com/ (Last accessed on September 03, 2014)

edonkey2000. (2000). Retrieved from http://www.emule-project.net/ (Last accessed on September 03, 2014)

El-Gamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of crypto'84 on advances in cryptology* (p. 10-18). Springer.

Fallahpour, M., & Megías, D. (2009). High capacity audio watermarking using fft amplitude interpolation. *IEICE Electron Express*, *6*(14), 1057-1063.

Fallahpour, M., & Megías, D. (2010). Dwt-based high capacity audio watermarking. *IEICE Transactions*, *93-A*(1), 331-335.

Feige, U., Fiat, A., & Shamir, A. (1998). Zero-knowledge proofs of identity. *Journal of Cryptology*, *1*, 77-94.

Fernandez, M., & Soriano, M. (2004). Soft-decision tracing in fingerprinted multimedia content. *IEEE Transactions on Multimedia*, *11*(2), 38-46.

Ffmpeg. (2000). Retrieved from https://www.ffmpeg.org/ (Last accessed on September 03, 2014)

Foo, S.-W., & Dong, Q. (2010). Audio watermarking based on compression-expansion technique. In *Proceedings of 10th ieee region conference:tencon'08* (p. 680-686).

Freehaven. (1999). Retrieved from http://www.freehaven.net/ (Last accessed on September 03, 2014)

Gao, H., Zeng, W., & Chen, Z. (2010). Fingerprinting for copyright protection in p2p context. In *Proceedings of international symposium on intelligence information processing and trusted computing* (p. 114-117).

García-Dorado, J., Finamore, A., Mellia, M., Meo, M., & Munafó, M. (2012). Characterization of isp traffic: Trends, user habits, and access technology impact. *IEEE Transactions on Network and Service Management*, *9*(2), 142-155.

Garcia-Hernandez, J. J., & Feregrino-Uribe, C. (2013). Collusion-resistant audio fingerprinting system in the modulated complex lapped transform domain. *PLoS ONE*, *8*(6), 1-15.

Gnanajeyaraman, R., Prasadh, K., & Ramar, D. (2009). Audio encryption using higher dimensional chaotic map. *International Journal of Recent Trends in Engineering*, *1*(2), 103-107.

Goldwasser, S., & Micali, S. (1984). Probabilistic encryption. *Journal of Computer and System Sciences*, *28*(2), 270-299.

Grodzinsky, F. S., & Tavani, H. T. (2005). P2p networks and the verizon v. riaa case: Implications for personal privacy and intellectual property. *Ethics and Information Technology*, *7*, 243-250.

gtk-gnutella. (2000). Retrieved from http://gtk-gnutella.sourceforge.net/en/?page=news (Last accessed on September 03, 2014)

Gupta, K. N., Agarwala, K. N., & Agarwala, P. A. (2005). *Digital signature: Network security practices*. Prentice-Hall.

He, D., Sun, Q., & Tian, Q. (2003). An object-based watermarking solution for mpeg4 video authentication. In *Proceedings of ieee international conference on acoustics, speech, and*

signal processing (Vol. 3, p. 537-540).

He, S., & Wu, M. (2006). Joint coding and embedding techniques for multimedia fingerprinting. *IEEE Transactions on Information Forensics and Security*, *1*(2), 231-247.

Hollmann, H. D. L., Lint, J. H. V., Linnartz, J. P., & Tolhuizen, L. M. G. M. (1998). *Journal of Combinatorial Theory*, *82*, 121-133.

Hu, D., & Li, Q. (2010). Bandwidth efficient asymmetric fingerprinting based on one-out-of-two oblivious transfer. *International Journal of Information and Computer Security*, *4*(2), 152-163.

Huang, Y. B., Zhang, Q. Y., Liu, Z., Di, Y. J., & Yuan, Z. T. (2012). A dither modulation audio watermarking algorithm based on has. *Research Journal of Applied Sciences, Engineering and Technology*, *4*(21), 4206-4211.

Hussein, J., & Mohammed, A. (2009). Robust video watermarking using multi-band wavelet transform. *International Journal of Computer Science Issues*, *6*(1), 44-49.

Icq. (1996). Retrieved from http://www.icq.com/en (Last accessed on September 03, 2014)

Inamura, M., & Iwamura, K. (2014). A license management system for content separate delivery over p2p network. *International Journal of Digital Information and Wireless Communications*, *4*(2), 34-44.

Internap. (1996). Retrieved from http://www.internap.com/ (Last accessed on September 23, 2014)

Isdal, T., Piatek, M., Krishnamurthy, A., & Anderson, T. (2009). Privacy-preserving p2p data sharing with oneswarm. University of Washington. (Last accessed on September 03, 2014)

Islam, N., Puech, W., & Brouzet, R. (2009). A homomorphic method for sharing secret images. In *Proceedings of the 8th international workshop on digital watermarking* (p. 121-135). Springer.

Jabber. (2008). Retrieved from http://www.jabber.org/ (Last accessed on September 03, 2014)

Jonker, H. L., & Mauw, S. (2004). Discovering the core security requirements of drm systems by means of objective trees. In S. Lian & Y. Zhang (Eds.), *Handbook of research on secure multimedia distribution.* Information Science Reference. (Last accessed on September 03, 2014)

Ju, H. S., Kim, H. J., Lee, D. H., & Lim, J. I. (2003). An anonymous buyer-seller watermarking protocol with anonymity control. In *Proceedings of the 5th international conference on*

information security and cryptology (p. 421-432). Springer.

Kaliski, B. (2000). *Pkcs #5: Password-based cryptography specification*. RFC Editor. Retrieved from http://www.apps.ietf.org/rfc/rfc2898.html

Kamvar, S. D., Schlosser, M. T., & Garcia-Molina, H. (2003). *The eigentrust algorithm for reputation management in p2p networks.* ACM.

Katzenbeisser, S., & Petitcolas, F. (2000). *Information techniques for steganography and digital watermarking*. Artech House.

Kesden, G. (2000). 15-412 operating systems: Design and implementation lecture.. (Last accessed on September 03, 2014)

Khurana, K., & Chandak, M. B. (2013). Key frame extraction methodology for video annotation. *International Journal of Computer Engineering and Technology*, *4*(2), 221-228.

Kiayias, A., Tsiounis, Y., & Yung, M. (2004). Traceable signatures. In *Proceedings of advances in cryptology* (Vol. 3027). Springer.

Kostić, D., Rodriguez, A., Albrecht, J., & Vahdat, A. (2003). Bullet: High bandwidth data dissemination using an overlay mesh. In *Proceedings of the 19th acm symposium on operating systems principles* (p. 282-297). ACM.

Kuribayashi, M. (2010). On the implementation of spread-spectrum fingerprinting in asymmetric cryptographic protocol. *EURASIP Journal on Information Security*, *2010*, 1-11.

Kuribayashi, M., & Mori, M. (2008). On the implementation of asymmetric fingerprinting protocol. In *Proceedings of european signal processing* (p. 1-5).

Kuribayashi, M., & Tanaka, H. (2005). Fingerprinting protocol for images based on additive homomorphic property. *IEEE Transactions on Image Processing*, *14*(12), 2129-2139.

Lancini, R., Mapelli, F., & Tubaro, S. (2002). A robust video watermarking technique in the spatial-domain. In *Proceedings of 4th eurasip-ieee region 8 international symposium on video/image processing and multimedia communications* (p. 251-256).

Larson, S., Snow, C., & Pande, V. (2003). *Chapter folding@home and genome@home: Using distributed computing to tackle previously intractable problems in computational biology* (B. Schölkopf, K. Tsuda, & J. P. Vert, Eds.). Horizon Press.

Lee, W. B., & Chen, T. H. (2002). A public verifiable copy protection technique for still images. *Journal of Systems and Software*, *62*(3), 195-204.

Leelavathy, N., Prasad, E. V., Kumar, S. S., & Mohan, B. C. (2011). Oblivious image watermarking in discrete multiwavelet domain using qimm. *Journal of Multimedia*, *6*(4), 359-368.

Li, J. S., Hsieh, C. J., & Hung, C. F. (2010). A novel drm framework for peer-to-peer music content delivery. *Journal of Systems and Software*, *83*(10), 1689-1700.

Li, S., Zheng, X., Mou, X., & Cai, Y. (2002). Chaotic eencryption scheme for real-time digital video. In *Proceedings of spie on electronic imaging* (Vol. 4666, p. 149-160).

Li, X., Krishnan, S., & Ma, N. W. (2010). A wavelet-pca-based fingerprinting scheme for peer-to-peer video file sharing. *IEEE Transactions on Information Forensics and Security*, *5*(3), 365-373.

Lian, S., Kanellopoulos, D., & Ruffo, G. (2009). Recent advances in multimedia information system security. *Informatica Journal of Computing and Informatics*, *33*(1), 3-24.

Lian, S., Liu, Z., Ren, Z., & Wang, H. (2006). Secure advanced video coding based on selective encryption algorithms. *IEEE Transactions on Consumer Electronics*, *52*(2), 621-629.

Lian, S., Sun, J., & Wang, Z. (2004). A secure 3d-spiht codec. In *Proceedings of european signal processing conference* (p. 813-816).

Lin, E., Eskicioglu, A. M., Lagendijk, R. L., & Delp, E. J. (2005). Advances in digital video content protection. *Proceedings of the IEEE Special Issue on Advances in Video Coding and Delivery*, *93*(1), 171-183.

Lin, W. H., Wang, Y. R., Horng, S. J., Kao, T. W., & Pan, Y. (2009). A blind watermarking method using maximum wavelet coefficient quantization. *Expert Systems with Applications*, *36*(9), 11509-11516.

Liu, K. J. R., Trappe, W., Wang, Z. J., Wu, M., & Zhao, H. (2005). *Multimedia fingerprinting forensics for traitor tracing*. Hindawi Publishing Co.

Liu, Q., Safavi-Naini, R., & Sheppard, N. P. (2003). Digital rights management for content distribution. In *Proceedings of the australasian information security workshop conference on acsw frontiers* (p. 49-58).

Lu, L., Han, J., Liu, Y., Hu, L., Huai, J., Ni, L. M., & Ma, J. (2007). Pseudo trust: Zero-knowledge authentication in anonymous p2ps. *IEEE Transactions on Parallel and Distributed Systems*, *19*(10), 1-10.

Lua, K., Crowcroft, J., Pias, M., Sharma, R., & Lim, S. (2005). A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys and Tutorials*, *7*(2), 72-93.

Mack, M. B. (2009). P2p survival guide: What users must know.. Retrieved from http://thehill.com/opinion/op-ed/8129-p2p-survival-guide-what-users-must-know (Last accessed on September 03, 2014)

Martínez-Ballesté, A., Sebé, F., Domingo-Ferrer, J., & Soriano, M. (2003). Practical asymmetric fingerprinting with a ttp. In *Proceedings of the 14th international workshop on database and expert systems applications* (p. 352-356). IEEE.

Megías, D. (2014). Improved privacy-preserving p2p multimedia distribution based on recombined fingerprints. *IEEE Transactions on Dependable and Secure Computing*. (To be published)

Megías, D., & Domingo-Ferrer, J. (2014). Privacy-aware peer-to-peer content distribution using automatically recombined fingerprints. *Multimedia Systems*, *20*(2), 105-125.

Memon, N. D., & Wong, P. W. (2001). A buyer-seller watermarking protocol. *IEEE Transactions on Image Processing*, *10*(4), 643-649.

Milojicic, D. S., Kalogeraki, V., Lukose, R., Nagaraja, K., Pruyne, J., Richard, B., ... Xu, Z. (2002). Peer-to-peer computing. HP Laborataries Palo Alto. (Last accessed on September 03, 2014)

Mp3stego tool. (1997). Retrieved from http://www.petitcolas.net/steganography/mp3stego/ (Last accessed on September 03, 2014)

Msu video quality measurement tool. (2011). Retrieved from http://compression.ru/video/quality_measure/video_measurement_tool_en.html (Last accessed on September 03, 2014)

Naor, M., & Pinkas, B. (1998). *Threshold traitor tracing.*

Napster. (2011). Retrieved from http://www.napster.co.uk/start (Last accessed on September 03, 2014)

Ntl: A library for doing number theory. (1990). Retrieved from http://www.shoup.net/ntl/ (Last accessed on September 03, 2014)

Nuida, K., Fujitsu, S., Hagiwara, M., Kitagawa, T., Watanabe, H., Ogawa, K., & Imai, H. (2007). An improvement of tardos's collusion-secure fingerprinting codes with very short lengths. In *Proceedings of the 17th international conference on applied algebra, algebraic algorithms and error-correcting codes* (p. 80-89). Springer.

Oates, B. (2005). *Researching information systems and computing*. SAGE Publishing Ltd.

Open mobile alliance. (2002). Retrieved from http://www.openmobilealliance.org/release_program/uap_v2_0.html (Last accessed on September 03, 2014)

Opentrust. (2004). Retrieved from https://www.opentrust.com/en/ (Last accessed on September 03, 2014)

Opera. (1999). Retrieved from http://www.opticom.de/products/audio-quality-testing.html/ (Last accessed on September 03, 2014)

Pagnia, H., & Gartner, F. C. (1999). On the impossibility of fair exchange without a trusted third party. Darmstadt University of Technology. (Last accessed on September 03, 2014)

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the 17th international conference on theory and application of cryptographic techniques* (p. 223-238). Springer.

Pearstreamer. (2013). Retrieved from http://peerstreamer.org/ (Last accessed on September 03, 2014)

Peercast. (2006). Retrieved from http://www.peercast.com-about.com/ (Last accessed on September 03, 2014)

Pehlivanoglu, S. (2013). An asymmetric fingerprinting code for collusion-resistant buyer-seller watermarking. In *Proceedings of the first acm workshop on information hiding and multimedia security* (p. 35-44). ACM.

Perez-Gonzalez, F., Mosquera, C., Barni, M., & Abrardo, A. (2005). Rational dither modulation: A high-rate data-hiding method invariant to gain attacks. *IEEE Transactions on Signal Processing*, *53*(10), 3960-3975.

Petitcolas, F., Anderson, R., & Kuhn, M. (1998). Attacks on copyright marking systems. In *Proceedings of the 2nd workshop on information hiding* (p. 218-238). Springer.

Pfitzmann, A., & Hansen, M. (2010). A terminology for talking about privacy by data minimization:anonymity, unlinkability, undetectability, unobservability,pseudonymity, and identity management. TU Dresden. (Last accessed on September 03, 2014)

Pfitzmann, B., & Sadeghi, A.-R. (1999). Coin-based anonymous fingerprinting. In *Proceedings of the 18th annual international conference on the theory and application of cryptographic techniques* (Vol. 1592, p. 150-164). Springer.

Pfitzmann, B., & Schunter, M. (1996). Asymmetric fingerprinting. In *Proceedings of the 15th annual international conference on theory and application of cryptographic techniques* (p. 84-95). Springer.

Pfitzmann, B., & Waidner, M. (1997). Anonymous fingerprinting. In *Proceedings of the 16th annual international conference on theory and application of cryptographic techniques* (p. 88-102). Springer.

Prins, J. P., Erkin, Z., & Lagendijk, R. L. (2007). Anonymous fingerprinting with robust qim watermarking techniques. *EURASIP Journal on Information Security*, *2007*(20), 1-7.

Qian, L., & Nahrstedt, K. (1998). Watermarking schemes and protocols for protecting rightful ownership and customers rights. *Journal of Visual Communication and Image Representation*, *9*(3), 194-210.

Qureshi, A., Megías, D., & Rifà-Pous, H. (n.d.). Psum: Peer-to-peer multimedia content distribution using collusion-resistant fingerprinting.
(Submitted)

Qureshi, A., Megías, D., & Rifà-Pous, H. (2014). Secure and anonymous multimedia content distribution in peer-to-peer networks. In *Proceedings of the 6th international conference on advances in multimedia* (p. 91-96).

Qureshi, A., Megías, D., & Rifà-Pous, H. (2015). Framework for preserving security and privacy in peer-to-peer content distribution systems. *Expert Systems with Applications*, *42*, 1391-1408. (To be published)

Qureshi, A., Rifà-Pous, H., & Megías, D. (2013a). Security, privacy and anonymity in legal distribution of copyrighted multimedia content over peer-to-peer networks: A brief overview. In *Proceedings of the 2013 fifth international conference on multimedia information networking and security* (p. 583-587). IEEE Computer Society.

Qureshi, A., Rifà-Pous, H., & Megías, D. (2013b). *A survey on security, privacy and anonymity in legal distribution of copyrighted multimedia content over peer-to-peer networks* (Vol. DWP 13-001; Working Paper). Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC).

Ratnasamy, S., Francis, P., Handley, M., Karp, R., & Shenker, S. (2001). A scalable content-addressable network. *SIGCOMM Computer Communication Review*, *31*(4), 161-172.

Reed, M., Syverson, P., & Goldschlag, D. (1998). Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, *16*(4), 482-494.

Rivest, R. L. (1990). The md4 message digest algorithm. In *Advances in cryptology-crypt0'90* (Vol. 537, p. 303-311). Springer.

Rivest, R. L. (1992). The md5 message digest algorithm. RFC Editor. (Last accessed on September 03, 2014)

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), 120-126.

Roncancio, C., Villamil, M. D. P., Labbe, C., & Serrano-Alvarado, P. (2009). Data sharing in dht-based p2p systems. *Transactions on Large Scale Data and Knowledge Centered Systems*, *5740*, 327-352.

Rowstron, A. I. T., & Druschel, P. (2001). Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proceedings of acm/ifip/usenix middleware conference* (p. 329-350).

Scarlata, V., Levine, B. N., & Shields, C. (2001). Responder anonymity and anonymous peer-to-peer file sharing. In *Ninth international conference on network protocols* (p. 272-280). IEEE.

Schaathun, H. G. (2003). Fighting two pirates. In *Applied algebra, algebraic algorithms and error-correcting codes* (Vol. 2643, p. 71-78). Springer.

Schneier, B. (1996). *Applied cryptography protocols, algorithms, and source code in c* (2nd ed.). John Wiley and Sons, Inc.

Sebé, F., & Domingo-Ferrer, J. (2002). Short 3-secure fingerprinting codes for copyright protection. , *2384*, 316-327.

Secure hash standard, fips 186. (1992). National Institute of Standards and Technology. Retrieved from http://csrc.nist.gov/publications/PubsFIPS.html (Last accessed on September 03, 2014)

Secure hash standard, fips 186-1. (1995). National Institute of Standards and Technology. Retrieved from http://csrc.nist.gov/publications/PubsFIPS.html (Last accessed on September 03, 2014)

Selimis, G., Sklavos, N., & Koufopavlou, O. (2004). Crypto processor for contactless smart cards. In *Proceedings of the 12th ieee mediterranean electrotechnical conference* (Vol. 2, p. 803-806).

Serrao, C., Serra, A., Dias, M., & Delgado, J. (2006). Protection of mp3 music files using digital rights management and symmetric ciphering. In *Proceedings of 2nd international conference on automated production of cross media content for multi-channel distribution* (p. 128-135). IEEE.

Servetti, A., Testa, C., & Carlos de Martin, J. (2003). Frequency-selective partial encryption of compressed audio. In *Proceedings. of ieee international conference on acoustics, speech, and signal processing* (Vol. 5, p. 668-671).

Seti@home. (1999). Retrieved from http://setiathome.berkeley.edu/ (Last accessed on September 03, 2014)

Sherwood, R., Bhattacharjee, B., & Srinivasan, A. (2002). P5 : A protocol for scalable anonymous communication. In *Proceedings of ieee symposium on security and privacy* (p. 58-70).

Shterev, I., & Lagendijk, R. L. (2006). Amplitude scale estimation for quantization-based watermarking. *IEEE Transactions on Signal Processing*, *54*(11), 4146-4155.

Simple directmedia layer. (1998). Retrieved from https://www.libsdl.org/ (Last accessed on September 03, 2014)

Skoríc, B., Katzenbeisser, S., & Celik, M. (2008). Symmetric tardos fingerprinting codes for arbitrary alphabet sizes. *Designs, Codes and Cryptography*, *46*, 137-166.

Skype. (2003). Retrieved from http://www.skype.com/en/ (Last accessed on September 03, 2014)

Solove, D. J. (2006). A taxonomy of privacy. *University of Pennsylvania Law Review*, *154*(3).

Sopcast. (2008). Retrieved from http://www.sopcast.com/ (Last accessed on September 03, 2014)

Staddon, J. N., Stinson, D. R., & Wei, R. (2001). Combinatorial properties of frameproof and traceability codes. *IEEE Transactions on Information Theory*, *47*(3), 1042-1049.

Star-force. (2000). Retrieved from http://www.star-force.com/ (Last accessed on September 03, 2014)

Stenborg, K. G., Herberthson, M., & Forchheimer, R. (2011). Distribution of individually watermarked content in peer-to-peer networks. In *Proceedings of 4th ifip international conference on new technologies, mobility and security* (p. 1-4).

Stinson, D. R., Tran, V. T., & Wei, R. (1998). Secure frameproof codes, key distribution patterns, group testing algorithms and related structures. *Journal of Statistical Planning and Inference*, *86*, 595-617.

Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., & Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications* (p. 149-160). ACM.

Sun, M. K., Laih, C. S., Yen, H. Y., & Kuo, J. R. (2009). A ticket based digital rights management model. In *Proceedings of 6th ieee consumer communications and networking conference* (p. 1-5).

Sun, Y. (2014). Rightholder as the center: The drm system in copyright after so many years. *Social Science Research Network*, 1-41.

Tardos, G. (2003). Optimal probabilistic fingerprint codes. In *Proceedings of the thirty-fifth annual acm symposium on theory of computing* (p. 116-125). ACM.

Theotokis, S. A., & Spinellis, D. (2004). A survey of peer-to-peer content distribution technologies. *ACM Computer Survey*, *36*(4), 335-371.

Thiede, T., Treurniet, W. C., Bitto, R., Schmidmer, C., Sporer, T., Beerends, J. G., & Colomes, C. (2000). Peaq-the itu standard for objective measurement of perceived audio quality. *IEEE Transactions on Aerospace and Electronic Systems*, *48*(1-2), 3-29.

Torrubia, A., & Mora, F. (2002). Perceptual cryptography on mpeg-1 layer iii bit-streams. In *Proceedings of international conference on consumer electronics* (p. 324-325). IEEE.

Trappe, W., Song, J., Poovendran, R., & Liu, K. J. R. (2003). Key management and distribution for secure multimedia multicast. *IEEE Transactions on Multimedia*, *5*(4), 544-557.

Trung, T. V., & Martirosyan, S. (2005). New constructions for ipp codes. *Design, Codes and Cryptography*, *35*, 227-239.

Trusted computing group. (2003). Retrieved from http://www.trustedcomputinggroup.org/ (Last accessed on September 03, 2014)

Tsai, Y. K. Y., M. J., & Chen, Y. Z. (2000). Joint wavelet and spatial transformation for digital watermarking. *IEEE Transactions on Consumer Electronics*, *46*(1), 137-144.

Tsolis, D. K., Sioutas, S., Panaretos, A., Karydis, I., & Oikonomou, K. (2011). Decentralized digital content exchange and copyright protection. In *Proceedings of ieee symposium on computers and communications* (p. 1056-1061).

Van-Schyndel, R. G., Tirkel, A. Z., & Osborne, C. F. (1994). A digital watermark. In *Proceedings of ieee international conference on image processing* (Vol. 2, p. 86-90).

Vijayan, J. (2010). P2p networks a treasure trove of leaked health care data. *Communications of the ACM*.

Vu, Q. H., & Ooi, B. C. (2010). Architecture of peer-to-peer systems. *Peer-to-Peer Computing*, 11-37.

Walkowiak, K., & Przewozniczek, M. (2011). Modeling and optimization of survivable p2p multi-casting. *Computer Communications*, *34*, 1410-1424.

Wang, Y., Nakao, A., Vasilakos, A. V., & Ma, J. (2011). P2p soft security: On evolutionary dynamics of p2p incentive mechanism. *Computer Communications*, *34*, 241-249.

Wang, Y., & Vassileva, J. (2003). Trust and reputation model in peer-to-peer networks. In *Proceedings ofthird international conference on peer-to-peer computing* (p. 150-157). IEEE.

Wang, Z. J., Wu, M., Zhao, H. V., Trappe, W., & Liu, K. J. R. (2005). Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation. *IEEE Transactions on Image Processing*, *14*(6), 804-821.

Westin, A. F. (1967). *Privacy and freedom*. The Bodley Head Ltd.

Win, T. T., L.W., & Emmanuel, S. (2011). A privacy-preserving content distribution mechanism for drm without trusted third parties. In *Proceedings of ieee international conference on multimedia and expo* (p. 1-6).

Wu, C., Zheng, Y., Ip, W. H., Chan, C. Y., Yung, K. L., & Lu, Z. M. (2010). A flexible h.264/avc

compressed video watermarking scheme using particle-swarm optimization based dither-modulation. *AEU-International Journal of Electronics and Communications*, *65*(1), 27-36.

Wu, T. (1998). The secure remote password protocol. In *Proceedings of the symposium on internet society network and distributed system security* (p. 97-111).

Xinkai, W., Wang, P., Zhang, P., Xu, S., & Yang, H. (2013). A norm-space, adaptive, and blind audio watermarking algorithm by discrete wavelet transform. *Signal Processing*, *93*(4), 913-922.

Xiong, L., & Liu, L. (2004). Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, *16*(7), 843-857.

Yong-Mei, C., Wen-Giang, G., & Hai-Yang, D. (2013). An audio blind watermarking scheme based on dwt-svd. *Journal of Software*, *8*(7), 1801-1808.

Yu, F., Lee, D., & Ramakrishnan, K. K. (2011). Nemor: A congestion-aware protocol for anonymous peer-based content distribution. In *Proceedings of ieee conference on p2p computing* (p. 260-269).

Zhang, Y., & Fang, Y. (2007). A fine-grained reputation system for reliable service selection in peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, *18*(8), 1134-1145.

List of Publications

Conferences with Proceedings

1. **Security, Privacy and Anonymity in Legal Distribution of Copyrighted Multimedia Content over Peer-to-Peer Networks: A Brief Overview.**
   Qureshi, A., Rifà, H. and Megías, D.
   Proc. of the 5th International Conference on Multimedia Information Networking and Security, MINES 2013, IEEE Computer Society, pp. 583-587.

2. **Secure and Anonymous Multimedia Content Distribution in Peer-to-Peer Networks.**
   Qureshi, A., Megías, D. and Rifà, H.
   Proc. of the 6th International Conference on Advances in Multimedia, MMEDIA 2014, International Academy, Research, and Industry Association (IARA), pp. 91-96.

Workshops

1. **Privacy-aware Multimedia Content Distribution in Peer-to-Peer Networks.**
   Qureshi, A., Megías, D. and Rifà, H.
   First UOC International Research Symposium, Universitat Oberta de Catalunya, Barcelona, Spain, 2013.

Articles Indexed in ISI JCR (SCI)

1. **Framework for Preserving Security and Privacy in Peer-to-Peer Content Distribution Systems.**

Qureshi, A., Megías, D. and Rifà, H.

Expert Systems with Applications., 42(2015), pp. 1391-1408, Elsevier, 2015. (IF=1.965, 1st quartile).

2. **PSUM: Peer-to-Peer Multimedia Content Distribution using Collusion-resistant Fingerprinting.**

   Qureshi, A., Megías, D. and Rifà, H.

   under review, September 2014.

Article in Other Peer-Review Journals

1. **A Survey on Security, Privacy and Anonymity in Legal Distribution of Copyrighted Multimedia Content over Peer-to-Peer Networks.**

   Qureshi, A., Rifà, H. and Megías, D.

   IN3 Working Paper Series, vol. DWP 13-001, pp. 1-30, 2013.