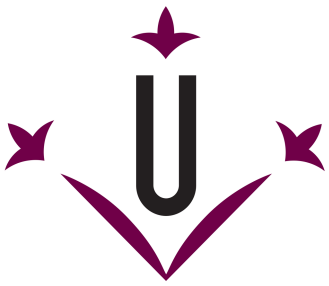
Universitat de Lleida

Algebraic Curves and Cryptographic Protocols for the e-society

Ricard Josep Garra Oronich

http://hdl.handle.net/10803/663364



Universitat de Lleida

TESI DOCTORAL

Algebraic Curves and Cryptographic Protocols for the e-society

Ricard Josep Garra Oronich

Memòria presentada per optar al grau de Doctor per la Universitat de Lleida
Programa de Doctorat en Enginyeria i Tecnologies de la Informació

Director
Josep M. Miret Biosca

Tutor
Josep M. Miret Biosca

2018

"There is nothing like looking, if you want to find something. You certainly usually find something, if you look, but it is not always quite the something you were after."

Gandalf, from *The Hobbit*, by *J.R.R. Tolkien*

"No matter how you spend your life, your wit will defend you more often than a sword. Keep it sharp!"

Abenthy, from *The Name of the Wind*, by *Patrick Rothfuss*

"Surprising what you can dig out of books if you read long enough, isn't it?"

Rand al'Thor, from *The Shadow Rising*, by *Robert Jordan*

"Too many scholars think of research as purely a cerebral pursuit. If we do nothing with the knowledge we gain, then we have wasted our study. Books can store information better than we can..., what we we do that books cannot is interpret. So if one is not going to draw conclusions, then one might as well just leave the information in the texts."

Jasnah Kholin, from *The Way of Kings*, by *Brandon Sanderson*

Resum

Amb l'augment permanent de l'adopció de sistemes intel·ligents de tot tipus en la societat actual apareixen nous reptes. Avui en dia quasi tothom en la societat moderna porta a sobre almenys un telèfon intel·ligent, si no és que porta encara més dispositius capaços d'obtenir dades personals, com podria ser un smartwatch per exemple. De manera similar, pràcticament totes les cases tindran un comptador intel·ligent en el futur pròxim per a fer un seguiment del consum d'energia. També s'espera que molts més dispositius del Internet de les Coses siguin instal·lats de manera ubiqua, recol·lectant informació dels seus voltants i/o realitzant accions, com per exemple en sistemes d'automatització de la llar, estacions meteorològiques o dispositius per la ciutat intel·ligent en general. Tots aquests dispositius i sistemes necessiten enviar dades de manera segura i confidencial, les quals poden contindre informació sensible o de caire privat. A més a més, donat el seu ràpid creixement, amb més de nou mil milions de dispositius en tot el món actualment, s'ha de tenir en compte la quantitat de dades que cal transmetre.

En aquesta tesi mostrem la utilitat de les corbes algebraiques sobre cossos finits en criptosistemes de clau pública, en particular la de les corbes de gènere 2, ja que ofereixen la mida de clau més petita per a un nivell de seguretat donat i això redueix de manera significativa el cost total de comunicacions d'un sistema, a la vegada que manté un rendiment raonable. Analitzem com la valoració 2-àdica del cardinal de la Jacobiana augmenta en successives extensions quadràtiques, considerant corbes de gènere 2 en cossos de característica senar, incloent les supersingulars. A més, millorem els algoritmes actuals per a computar la meitat d'un divisor d'una corba de gènere 2 sobre un cos binari, cosa que pot ser útil en la multiplicació escalar, que és l'operació principal en criptografia de clau pública amb corbes.

Pel que fa a la privacitat, presentem un sistema de pagament d'aparcament per mòbil que permet als conductors pagar per aparcar mantenint la seva privacitat, i per tant impedint que el proveïdor del servei o un atacant obtinguin un perfil de conducta d'aparcament. Finalment, presentem protocols de smart metering millorats, especialment pel que fa a la privacitat i evitant l'ús de terceres parts de confiança.

Resumen

Con el aumento permanente de la adopción de sistemas inteligentes de todo tipo en la sociedad actual aparecen nuevos retos. Hoy en día prácticamente todos en la sociedad moderna llevamos encima al menos un teléfono inteligente, si no es que llevamos más dispositivos capaces de obtener datos personales, como podría ser un smartwatch por ejemplo. De manera similar, en el futuro cercano la mayoría de las casas tendrán un contador inteligente para hacer un seguimiento del consumo de energía. También se espera que muchos más dispositivos del Internet de las Cosas sean instalados de manera ubicua, recolectando información de sus alrededores y/o realizando acciones, como por ejemplo en sistemas de automatización del hogar, estaciones meteorológicas o dispositivos para la ciudad inteligente en general. Todos estos dispositivos y sistemas necesitan enviar datos de manera segura y confidencial, los cuales pueden contener información sensible o de ámbito personal. Además, dado su rápido crecimiento, con más de nueve mil millones de dispositivos en todo el mundo actualmente, hay que tener en cuenta la cantidad de datos a transmitir.

En esta tesis mostramos la utilidad de las curvas algebraicas sobre cuerpos finitos en criptosistemas de clave pública, en particular la de las curvas de género 2, ya que ofrecen el tamaño de clave más pequeño para un nivel de seguridad dado y esto disminuye de manera significativa el coste total de comunicaciones del sistema, a la vez que mantiene un rendimiento razonable. Analizamos como la valoración 2-ádica del cardinal de la Jacobiana aumenta en sucesivas extensiones cuadráticas, considerando curvas de género 2 en cuerpos de característica importa, incluyendo las supersingulares. Además, mejoramos los algoritmos actuales para computar la mitad de un divisor de una curva de género 2 sobre un cuerpo binario, lo cual puede ser útil en la multiplicación escalar, que es la operación principal en criptografía de clave pública con curvas.

Respecto a la privacidad, presentamos un sistema de pago de aparcamiento por móvil que permite a los conductores pagar para aparcar manteniendo su privacidad, y por lo tanto impidiendo que el proveedor del servicio o un atacante obtengan un perfil de conducta de aparcamiento. Finalmente, ofrecemos protocolos de smart metering mejorados, especialmente en lo relativo a la privacidad y evitando el uso de terceras partes de confianza.

Abstract

With the ever increasing adoption of smart systems of every kind throughout society, new challenges arise. Nowadays, almost everyone in modern societies carries a smartphone at least, if not even more devices than can also gather personal data, like a smartwatch or a fitness wristband for example. Similarly, practically all homes will have a smart meter in the near future for billing and energy consumption monitoring, and many other Internet of Things devices are expected to be installed ubiquitously, obtaining information of their surroundings and/or performing some action, like for example, home automation systems, weather detection stations or devices for the smart city in general. All these devices and systems need to securely and privately transmit some data, which can be sensitive and personal information. Moreover, with a rapid increase of their number, with already more than nine billion devices worldwide, the amount of data to be transmitted has to be considered.

In this thesis we show the utility of algebraic curves over finite fields in public key cryptosystems, specially genus 2 curves, since they offer the minimum key size for a given security level and that significantly reduces the total communication costs of a system, while maintaining a reasonable performance. We analyze how the 2-adic valuation of the cardinality of the Jacobian increases in successive quadratic extensions, considering genus 2 curves with odd characteristic fields, including supersingular curves. In addition, we improve the current algorithms for computing the halving of a divisor of a genus 2 curve over binary fields, which can be useful in scalar multiplication, the main operation in public key cryptography using curves.

As regards to privacy, we present a pay-by-phone parking system which enables drivers to pay for public parking while preserving their privacy, and thus impeding the service provider or an attacker to obtain a profile of parking behaviors. Finally, we offer better protocols for smart metering, especially regarding privacy and the avoidance of trusted third parties.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The security and privacy of modern society rely heavily on cryptographic protocols to protect electronic systems, from personal smartphones to companies and government infrastructures. Every day, more and more personal data is generated, sent and stored worldwide, which makes the use of better cryptographic and privacy protocols necessary. Besides, many of the new devices that will be connected to the Internet are going to have small computational power, since the Internet of Things will be coed of a vast array of different, small and simple devices, which in turn will generate more data, making Big Data an even bigger topic.

Furthermore, some classical systems are being updated and/or upgraded to work with modern technology, such as electricity consumption meters, loyalty systems, paid parking spots and even election voting. Each of them requires different considerations in order to ensure, at least, that they enjoy the same benefits as the classical counterpart as well as their privacy, with some other benefits being desirable. For example, regarding privacy, in an electronic voting system each ballot should not be able to be linked to a specific voter; in parking systems, it should not be possible to create a profile of a driver automatically by centralizing the data of where and when has she parked; and smart meters should not transmit information often enough such that a detailed profile of a consumer can be created, allowing the company (or an attacker) to determine when someone is at home or not.

In this thesis, several contributions are presented related to elliptic and

genus 2 curves used in cryptography, as well as new protocols on smart metering and parking systems that improve user's privacy. In the rest of this chapter, we are going to introduce some basic concepts in relation to public key cryptography and how the use of elliptic and hyperelliptic curves can improve some protocols. Then several ideas about the need of privacy are explained, as well as the introduction to two topics touched upon in our contributions. And finally, the structure of this thesis is presented.

1.1 Algebraic curve cryptography

Most of the computers and devices connected to the Internet use public key cryptography [DH76] at some point or another, which typically requires larger keys and messages than symmetric encryption protocols. Their security is based on the intractability of some mathematical problem which is believed to be hard to solve, such as the Integer Factorization Problem (IFP) and the Discrete Logarithm Problem (DLP). The DLP can be described over the multiplicative group of a finite field, over the group of points of an elliptic curve defined over a finite field, known as the Elliptic Curve Discrete Logarithm Problem (ECDLP) or over the Jacobian of a hyperelliptic curve (denoted by HCDLP). The main cryptographic cryptosystems that are based on these problems are the RSA [RSA78], based on the IDP, and the ElGamal [ElG85], based on the DLP and its variants [Mil85, Kob87, Kob89].

The required size of the values for each problem to achieve the same level of security depends on the efficiency of the best known algorithm for solving it. For the IFP this is the Number Field Sieve [LLMP93] and for the DLP, the Index-Calculus [How98], both with subexponential cost, and therefore the values used are very large, of around 3072 bits at least (for an equivalent 128 bits of security) as of 2018. On the other hand, for the ECDLP, a field of 256 bits would suffice while attaining the same security, and just 128 bits for the HCDLP over a genus 2 curve, since the group of divisors of a hyperelliptic curve is much bigger for a given field size. There are attacks for hyperelliptic curves of genus bigger than 2, hence they are not considered suitable for cryptography. It is therefore desirable to use elliptic or genus 2 curves, especially on devices with limited resources (be it

in memory or computation), such as smart cards, smart meters, smartphones and in general for the Internet of Things, which will add many millions or billions of devices, all of which will require some form of communication and have restricted power consumption.

1.1.1 Point counting algorithms

In order to start using elliptic curves, a suitable one must be chosen. Nevertheless, not all curves are useful for cryptographic purposes. In order to determine the validity of an elliptic curve over a finite field $\mathbb{F}_q$, it is necessary to know its cardinality over $\mathbb{F}_q$ [BSS99]. The same applies for the Jacobian of a genus 2 curve. However, although doable, it is still time consuming to compute the cardinality of a cryptographic size curve.

The best known algorithms are the Schoof-Elkies-Atkin (SEA) algorithm [Sch95, Elk98, Atk88] for elliptic curves, and an extension of Schoof's algorithm for genus 2 curves [GS04, GS12]. In both cases, they construct elements of the ℓ^k-torsion subgroup, for very small values of ℓ [MMRV05, MMRV09, MPR10]. In one of our contributions [GMPT18], we study the 2-adic valuation of the cardinality of Jacobians of genus 2 curves over of a field of odd characteristic, and their quadratic extensions, which can be useful in the initial steps of the algorithm.

1.1.2 Curve scalar multiplication

The fundamental operation when encrypting or decrypting a message using ElGamal, the main cryptosystem that is based on the ECDLP and HCDLP, is scalar multiplication (see Section 2.3.2 for more detailed information). There are several algorithms for performing a group element multiplication by a scalar, some with different curve equations (being the Weierstraß the most common) and/or different coordinate types other than affine . The basic algorithm is the regular double-and-add, which uses the binary representation of the scalar and is similar to the multiply-and-square used in modular exponentiation. An alternative is to write the scalar with a different representation, and use instead a halve-and-add algorithm, substituting the doubling of points or divisors by halvings (see Section 2.1.3 for more de-

tails), that is, given an element y, find x such that $2x = y$. Both algorithms can be combined and executed in parallel in order to faster compute a scalar multiplication, so that one processor executes the double-and-add and the other one the halve-and-add, and they combine their outputs to obtain the final result, speeding up the process [TFHA+11].

For curves of genus 2 over fields with characteristic 2, this alternative becomes more interesting since the Cantor algorithm [Can87] (see Algorithm 1) for doubling divisors is less efficient than the equivalent for elliptic curves. However, the halving of a divisor in a genus 2 curve is not unique (unless it is supersingular, which already makes it undesirable for cryptography), so the type of curve must be chosen carefully such that the halving can be used reliably. Seeing that, we studied and improved the method to compute the halving of any divisor in genus 2 curves of a specific type over binary fields in one of our contributions [GMP16], giving explicit formulae for each case.

1.2 Privacy and smart systems

With the increase in the use of technologies in modern days, more and more devices are in use that may collect data from the population. It is well known that Big Data is becoming a very relevant topic since the amount of information generated around the world has increased severalfold during the last few years. In turn, governments as well as citizens have become concerned about the collection and use of private data by companies. For example, anyone who uses about any service online has probably received several emails around May 2018, regarding the General Data Protection Regulation (GDPR), a European Union law on data protection and privacy for all individuals in the EU, which aims to give control to citizens and residents over their personal data.

On the one hand, there are services which inherently use and need personal data, sometimes given voluntarily, such as Google, Facebook, Twitter etc, or even banks and Internet service providers. In theory, at least, there are laws and regulations in regard to how they store and treat the information, such as to minimize possible data leaks and reduce their effect, but they are not always correctly followed and every now and then some major leak

or hack occurs.

On the other hand, some services are susceptible to be used in a more inherently private way. Privacy can be defined as *the ability of an individual or group to keep information secret*. The possible instances where privacy is required and/or desired are plenty, and some come with the electronic variant of classic situations. We are going to focus on two of these situations, which are the ones we touch upon in some of our contributions: smart metering systems, and parking systems using e-cash.

1.2.1 Smart metering

The classic method to record domestic electric energy consumption was to use a meter, the reading of which was checked by someone working for the company every so often, e.g. every 2 months. Sometimes, the value was estimated in the meantime depending on previous data, in the case the meter could not be checked, for instance if access to the residence was needed to check the meter and nobody was present.

Nowadays, classic meters are being substituted by *smart meters*, which are able to relate electricity consumption automatically to the company, among other things. They have been set up in many countries around the world, and in Spain for example, at the present time, the vast majority of the meters have already been substituted by smart meters, and all of them up to 15 kW must be changed before the end of 2018 [Esp]. In theory, they send the consumption every hour in this case, for billing purposes, in the case that the user has a contract with hour-depending prices. This also helps consumers to be informed "in real time" about their consumption and to consult historical data online.

These meters (or other ones in different countries) could send data more frequently, and even with just hourly consumptions, it could allow a possible attacker (or the company itself) to gather a great amount of sensitive information, inferring the users' daily routine or whether they are on vacation, for example, if proper care has not been taken. In other situations, the electricity company may have legitimate reasons to want to know more fine-grained data about total consumption, like for monitoring purposes in order to find

consumption patterns and adjust energy generation accordingly.

Nevertheless, transmitting such data regularly, i.e., every 15 or 30 minutes, raises some concerns about privacy. Since in these cases the exact values of each individual customer are not needed to be known, several techniques have been proposed, achieving privacy by different means:

- *Anonymization*: the link between the electricity reading and customer identity is removed before sending the data [EK10, FB13, Pet10].
- *Perturbation*: random noise is added by smart meters to the readings before they are transmitted. In general, this noise will not be removed, so the system must be tuned in such a way that the noise cancels out, or to provide an adequate trade-off between privacy and accuracy [ÁC11, BSU10].
- *Aggregation*: smart meters are divided into neighborhoods and add together their readings before they are transmitted. They can be aggregated using the homomorphic property of some cryptosystems or by a trusted third party [CMT05, MSP+13, BPS+16, NZLS16].

Our contributions regarding smart meters use aggregation: the first one analyzes and repairs an existing system, proposing a way to fix a security flaw that we found [GLMS], and the other one improves another system by removing the trusted dealer in the key establishment step [GMMS].

1.2.2 Parking systems

With the increase in the amount of vehicles in cities around the world, and the total on-street parking space being limited, with little to no option to increase it, it becomes necessary to restrict the maximum time a vehicle is allowed to occupy a parking spot in order to encourage regular turnover of parking bays, at least in the busiest areas of a city. In doing so, drivers are encouraged to shorten their parking time and therefore give others a reasonable chance of finding parking.

Classically, the solution has been to require drivers to go to a nearby paying station and pay for a specific amount of time using cash or a credit

card: then the machine issues a parking ticket, valid for that amount of time, which has to be placed in a visible area on the dashboard of the vehicle. The price per minute may vary depending on the area of the city, and usually has a maximum time allowed. To enforce the system, parking officers patrol the controlled zones and check each vehicle for violations, either an expired ticket or no ticket at all, and issue fines as needed.

Some obvious drawbacks and inconveniences are:

- Drivers need to estimate the duration of their parking in advance, and pay for it: this both means that the driver needs to have enough cash (if credit cards are not supported), and in the case the parking takes less time than expected, the unused time (money) is lost. Moreover, most machines do not give change back, making people more prone to pay more than they would really need.

- Drivers need to go to the pay station and back to their car in order to place the ticket, and they are not alway near or easy to find, which implies more time lost for the driver.

- If the driver needs to extend her stay, she needs to get back to her car from wherever she was and obtain a new ticket.

Many cities nowadays have the option to pay for parking by phone, using an app like [EYS]. There they need to create an account, usually introducing some personal data, like an e-mail and some source for funding such as a credit card, as well as a license plate number, and maybe some other information like name, ID number etc.

With that app, a driver may pay for her parking stay introducing or selecting her license plate number, the city area she is in, and the expected parking duration. The payment is then performed, either charging directly the credit card, or from credit balance in the app. Some applications may allow to end a parking session before its time expires so that the unused money is refunded. Note that all these transactions can be performed while being away from the car, even an extension of parking time. In this scenario, parking officers use a mobile device where they can check, by typing a license plate number, whether a payment for a parked car has been made.

At first glance, this may seem like an ideal solution, as it solves the previously stated drawbacks. However, new challenges arise from the use of such technology. In that kind of system, some central server controls and collects information of all the parking operations in a city/area, since the parking officers need to be able to query it. Moreover, the data can be stored and linked to a specific user/license plate number, and thus allowing to create a profile of the user over time, inferring the parking habits of car owners. Furthermore, the app and system are usually provided by a private company, which can give rise to even more concerns with respect to the privacy of the users' data.

With all this in mind, a system which makes use of the advantages of the mobile payment while maintaining the same level of privacy as legacy systems would be desirable. Some approaches can be found in [LLZS10, YYRO11]. For this reason, we present a privacy-preserving pay-by-phone parking system in one of our contributions [GMS17], such that the only way to know if a vehicle is parked (or when and where) is for a person (the parking officer) to be on the street near the car itself, while allowing remote payment for the user, as well as the ability for a driver to complain in the case she has been fined unfairly. Moreover, the parking officer will only get to know if the driver has paid at the specific time she checks the license plate, but will not know how much time is left, so even less information is provided to the officer, just the minimum needed.

1.3 Structure

The first chapter has been devoted to introduce the core concepts touched upon in this thesis. Chapter 2 details the notation and concepts, as well as mathematical and cryptographic background, required to understand the proposals. Our contributions can be found in Chapter 3, and are the following:

- Halving in some genus 2 curves over binary fields [GMP16].
- The 2-adic valuation of the cardinality of Jacobians of genus 2 curves over quadratic towers of finite fields [GMPT18].

- A Privacy-Preserving Pay-by-Phone Parking System [GMS17].
- Repairing an aggregation-based smart metering system [GLMS].
- Improving a smart metering system using elliptic curves and removing the trusted dealer [GMMS].

The content of the papers has not been added in order to prevent copyright infringement. Finally, conclusions are explained in Chapter 4.

Chapter 2

Preliminaries

In this chapter we are going to introduce some mathematical background and cryptographic protocols as well as some concepts that will be useful to understand the contributions presented in Chapter 3.

In Section 2.1 some base definitions regarding algebraic curves in general are presented, while Sections 2.1.1 and 2.1.3 give more details about elliptic and genus 2 curves respectively. Section 2.2 the basic concepts needed for computing scalar multiplication through halvings are introduced. And finally, Section 2.3 gives an introduction to some cryptographic protocols used in some of our contributions, for encrypting, signing or authenticating messages.

2.1 Hyperelliptic curves

We are going to introduce first a general description of an algebraic hyperelliptic curve of any genus as well as some of their properties and characteristics. For more details see [Sil09, Kob12].

Let $\mathbb{F}_q$ be a finite field with $q = p^m$, with p being a prime, and let $\overline{\mathbb{F}}_q$ be the algebraic closure of $\mathbb{F}_q$. A hyperelliptic curve C of genus g over $\mathbb{F}_q$ is given by an equation of the form

$$C : y^2 + h(x)y = f(x), \quad h(x), f(x) \in \mathbb{F}_q[x, y], \tag{2.1}$$

where $\deg(h) \leq g+1$ and $\deg(f) \leq 2g+2$ with no singular points, that is, there are no solutions $(x, y) \in \overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q$ which simultaneously satisfy $y^2 + h(x)y = f(x)$ and the partial derivatives $2y+h(x) = 0$ and $h'(x)y-f'(x) = 0$.

If the field has even characteristic, that is, $p = 2$ and therefore $q = 2^m$, then $h(x) \neq 0$. Otherwise, if it has odd characteristic, with a change of variables the equation can be simplified such that $y^2 = f(x)$ [MWZ96].

We denote by $C(\mathbb{F}_q)$ the set of all points $P = (x, y) \in \mathbb{F}_q \times \mathbb{F}_q$ which satisfy Equation 2.1 of the curve C, together with the *points at infinity*, which are the points at the intersection of C with the line at infinity in the projective plane $\mathbb{P}^2(\mathbb{F}_q)$.

If there is only one point P_∞ and $f(x)$ has no repeated roots, the degree of $f(x)$ is equal to $2g+1$ and the curve is an *imaginary* hyperelliptic curve; otherwise the degree is $2g+2$ and it is called a *real* hyperelliptic curve [CFA+05].

Let $P = (x, y) \in C(\mathbb{F}_q)$. Then the *opposite* or conjugate of P is the point $\tilde{P} = (x, -y - h(x))$. The opposite of P_∞ is defined as itself. If a point satisfies that $P = \tilde{P}$ the point is called *special*, otherwise it is said to be *ordinary*. The map $P \mapsto \tilde{P}$ is an involution, that is, $\tilde{\tilde{P}} = P$, and it is called the *hyperelliptic involution*.

2.1.1 Elliptic curves

A hyperelliptic curve of genus 1 defined over $\mathbb{F}_q$ is also called an *elliptic curve*, which we denote by E, and with the general Weierstraß equation

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \quad a_i \in \mathbb{F}_q$$

If the characteristic p of the field is neither 2 nor 3, then every elliptic curve can be written in the simplified form

$$E : y^2 = x^3 + ax + b, \tag{2.2}$$

using invertible linear transformations. In this case, which will be the one used in this thesis, in order for the curve to be nonsingular, the discriminant Δ_E of E, which is equal to the polynomial discriminant of $f(x)$, i.e., $\Delta_E =$

$-16(4a^3 + 27b^2)$, must be different from 0.

Group law

An addition operation can be defined over the set $E(\mathbb{F}_q)$, using the chord-tangent method. Let $P = (x_P, y_P), Q = (x_Q, y_Q), P, Q \in E(\mathbb{F}_q)$ be two points with $x_P \neq x_Q$, represented in affine form. The addition point $R = P + Q$ is defined as the symmetric point with respect to the x-axis resulting from the intersection of the curve and the straight line defined by the points P and Q, giving the point R, as can be seen in Figure 2.1a. Analytically, the coordinates of the resulting point $R = (x_R, y_R)$ are calculated as:

$$x_R = \lambda^2 - x_P - x_Q, \quad y_R = \lambda(x_P - x_R) - y_P, \tag{2.3}$$

where $\lambda = \dfrac{y_P - y_Q}{x_P - x_Q}$.

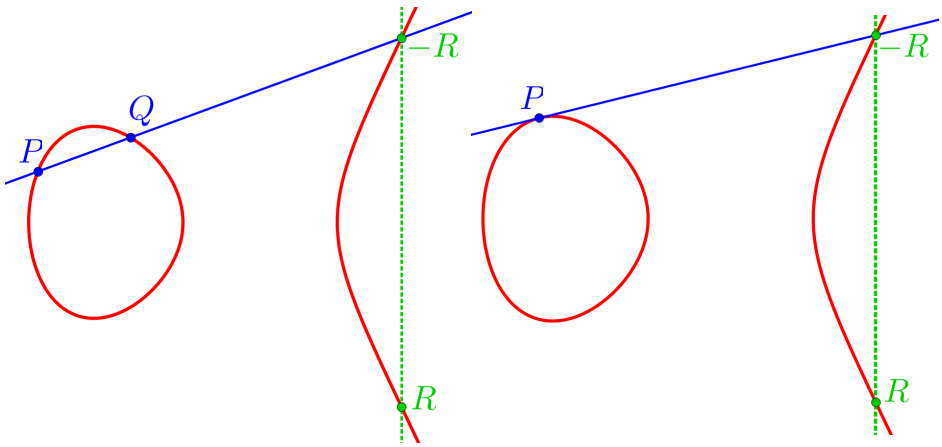
The computation of $2P$ follows the same equations as in Equation 2.3, taking $Q = P$, except that now $\lambda = \dfrac{3x_P^2 + a}{2y_P}$, where a is the coefficient of the Equation 2.2 of E. It can be seen as taking the symmetric point with respect to the x-axis of the intersection point of the tangent line to the curve at P and the curve, as in Figure 2.1b. It is relevant to note that there are other formulas used for different types of coordinates, such as projective or Jacobian, that are more efficient in some cases, and some are optimized for curves with some fixed coefficient, e.g. $a = -3$.

With this operation, the set of points $E(\mathbb{F}_q)$ forms an abelian group, in which the point at infinity $\mathcal{O} = P_\infty$ serves as the identity element [Sil09]. A multiplication of a point $P \in E$ by a scalar (integer) n is defined as the repeated addition of n times a point in that curve, denoted as

$$nP = P + \overset{n}{\cdots} + P.$$

The *order* of a point P is the smallest integer n such that $nP = \mathcal{O}$. By Lagrange's theorem, we know that the order of any point must divide the number of points on the curve, $\#E(\mathbb{F}_q)$, which can be expressed as

$$\#E(\mathbb{F}_q) = q + 1 - t$$



(a) $P + Q = R$. (b) $2P = R$.

Figure 2.1: Addition and doubling of points of $E : y^2 = x^3 - x + 1$ in $\mathbb{R}$.

being t the trace of the Frobenius endomorphism of E. The *Frobenius endomorphism* of a curve E is defined as:

$$\begin{aligned} \varphi : E(\overline{\mathbb{F}}_q) &\to E(\overline{\mathbb{F}}_q) \\ (x, y) &\mapsto (x^q, y^q) \\ \mathcal{O} &\mapsto \mathcal{O} \end{aligned}$$

This endomorphism satisfies the equation

$$X^2 - tX + q = 0.$$

Taking this into account, the Hasse inequality [Has33] states that

$$|t| \leq 2\sqrt{q}.$$

Concerning the group structure, it is isomorphic to either $\mathbb{Z}/n\mathbb{Z}$ or $\mathbb{Z}/n_1\mathbb{Z} \times \mathbb{Z}/n_2\mathbb{Z}$, where $n_1 \cdot n_2 = n = \#E(\mathbb{F}_q)$ with $n_2 | n_1$ and $n_2 | (q-1)$.

A point P is an ℓ-torsion point if and only if $\ell P = \mathcal{O}$. The set of all ℓ-torsion points of $E(\mathbb{F}_q)$, denoted by $E[\ell](\mathbb{F}_q)$, is a subgroup of $(E(\mathbb{F}_q), +)$. The ℓ-Sylow subgroup, with ℓ prime, of E over $\mathbb{F}_q$ is defined as

$$E[\ell^\infty](\mathbb{F}_q) = \{P \in E(\mathbb{F}_q) \mid \text{order}(P) = \ell^k,\ k \geq 0\}.$$

Both subgroups $E[\ell](\mathbb{F}_q)$ and $E[\ell^\infty](\mathbb{F}_q)$ can be either trivial, cyclic or of rank 2.

2.1.2 Divisors

For curves of genus $g \geq 2$ it is not possible to define a group law on the set of points of the curve C in a geometric way. Instead, sets of points are used, called divisors.

A *divisor* D is a finite formal sum of points $P \in C(\overline{\mathbb{F}}_q)$ of the form

$$D = \sum_{P \in C(\overline{\mathbb{F}}_q)} n_P(P), \quad n_P \in \mathbb{Z}$$

where only a finite amount of n_P are different from 0. In order for the divisor to be defined over $\mathbb{F}_q$ it must be invariant by the action of the Galois group $G = Gal(\overline{\mathbb{F}}_q/\mathbb{F}_q)$.

The *degree* of D, $\deg(D)$, is the sum of its coefficients:

$$\deg(D) = \sum_{P \in C(\overline{\mathbb{F}}_q)} n_P.$$

The set of all divisors of a hyperelliptic curve C is denoted by $\mathbb{D}_C$, and it forms an abelian group with the natural addition operation. The set of divisors of degree 0, which is a subgroup of $\mathbb{D}_C$, is denoted by $\mathbb{D}^0(C)$.

Consider a polynomial function ω over C, that is, a class of polynomials in the variables x and y equivalent modulo $y^2 + h(x)y - f(x)$. A *rational function* R over C is a quotient $\frac{\omega_1}{\omega_2}$ of two polynomial functions over C. Given a polynomial function R over C, the divisor formed by the zeros and poles of R over C, counted with their multiplicities, is called *divisor of* R, denoted by $\mathrm{div}(R)$.

A divisor $D \in \mathbb{D}^0(C)$ is a *principal divisor* if $D = \mathrm{div}(R)$ for some rational function R over C. The group of all principal divisors is denoted by

$$\mathbb{P}(C) = \{\mathrm{div}(R) \mid R \in \overline{\mathbb{F}}_q(C)\}.$$

The quotient group

$$\mathrm{Jac}(C)(\mathbb{F}_q) = \frac{\mathbb{D}^0(C)}{\mathbb{P}(C)}$$

is an algebraic variety called *Jacobian* variety of the curve C. In particular, $\mathrm{Jac}(C)(\mathbb{F}_q)$ is a finite abelian group, and the *neutral* element is represented as 0. The *order* of a divisor D is the smallest integer n such that $nD = 0$.

The cardinality of the Jacobian of a curve C of genus g defined over $\mathbb{F}_q$, $\#\mathrm{Jac}(C)(\mathbb{F}_q)$, is bounded by the Weil bounds [Wei49]:

$$(\sqrt{q} - 1)^{2g} \leq \#\mathrm{Jac}(C)(\mathbb{F}_q) \leq (\sqrt{q} + 1)^{2g}.$$

Since the size is of the order of q^g, much smaller fields can be used in comparison to elliptic curves while maintaining the same security level.

Mumford representation

Let C be a genus g hyperelliptic curve with $\deg(f) = 2g+1$, $\deg(h) \leq g$. We are going to use the Mumford representation so that each element of $D \in \mathrm{Jac}(C)(\mathbb{F}_q)$ is represented uniquely by a pair of polynomials $D = (u(x), v(x))$, $u, v \in \mathbb{F}_q[x]$.

By Riemann-Roch's theorem [Ful72], for each divisor class in $\mathrm{Jac}(C)(\mathbb{F}_q)$ there exists a unique divisor $D = \sum_{i=1}^{r} P_i - rP_\infty$, where $P_i \neq P_\infty$, $P_i \neq -P_j$ for $i \neq j$ and $r \leq g$, called a *reduced divisor*. Let $P_i = (x_i, y_i)$. Then the divisor can be represented by $D = (u(x), v(x))$ where

$$u(x) = \prod_{i=1}^{r} (x - x_i)$$

and $v(x)$ satisfying $v(x_i) = y_i$, with appropriate multiplicity.

Therefore, a reduced divisor $D = (u(x), v(x))$ fulfills the following properties:

1. u is monic.
2. $u \mid v^2 + vh - f$.
3. $\deg(v) < \deg(u) \leq g$.

The value $r = \deg(u)$, that is, the number of points of the support of D, is called the *weight* of the divisor.

Cantor's algorithms

The original Cantor's algorithms [Can87], improved by Koblitz [Kob89], and shown in Algorithm 1, is completely general and can be used for any field and genus to obtain the resulting reduced divisor of the addition of two divisors.

Algorithm 1 Cantor's algorithm for divisor addition.

Cantor's algorithm

Input: Two divisors $D_1 = (u_1, v_1)$ and $D_2 = (u_2, v_2)$ on the curve $C : y^2 + h(x)y = f(x)$.

Output: The unique reduced divisor D_3 such that $D_3 = D_1 + D_2$.

1: $d_1 = \gcd(u_1, u_2)$ ▷ $d_1 = e_1u_1 + e_2u_2$

2: $d = \gcd(d_1, v_1 + v_2 + h)$ ▷ $d = c_1d_1 + c_2(v_1 + v_2 + h)$

3: $s_1 = c_1e_1$, $s_2 = c_1e_2$, $s_3 = c_2$

4: $u = \dfrac{u_1u_2}{d^2}$, $v = \dfrac{s_1u_1u_2 + s_2u_2v_1 + s_3(v_1v_2 + f)}{d} \pmod{u}$

5: **while** $\deg(u) \leq g$ **do**

6: $\quad u' = \dfrac{f - vh - v^2}{u}$, $v' = (-h - v) \pmod{u'}$

7: $\quad u = u'$, $v = v'$

8: Make u monic

9: **return** $D = (u, v)$

Nevertheless, there are specific algorithms depending on the genus of the curve and characteristic of the field, as well as for other coordinate types, which are more efficient for certain operations (like doublings) or curves.

In Figure 2.2 a geometrical equivalent to elliptic curves can be seen for divisor addition on a genus 2 curve: two pairs of points representing each a divisor, a cubic curve that intersects them and the curve in another two points, which are the inverse of the result of the operation.

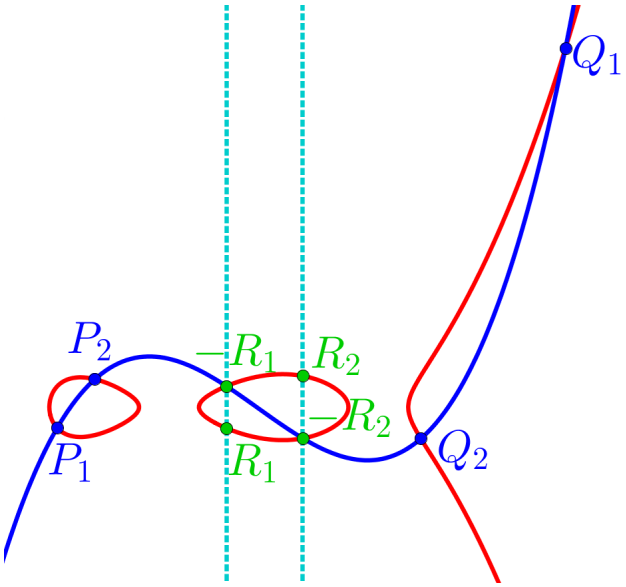


Figure 2.2: Sum of the divisors represented by the points (P_1, P_2) and (Q_1, Q_2), with result (R_1, R_2), on a genus 2 curve over $\mathbb{R}$.

2.1.3 Genus 2 curves

All genus 2 curves are hyperelliptic, but for genus $g \geq 3$ there are non-hyperelliptic curves. In general, the higher the genus of the curve, the bigger the group, however for $g \geq 3$ there are attacks for the Hyperelliptic Curve Discrete Logarithm Problem that are faster than the generic ones, hence they are not considered suitable for cryptographic purposes [Thé03].

A genus 2 curve C of over $\mathbb{F}_q$ is given by an equation of the form

$$C : y^2 + h(x)y = f(x), \quad h(x), f(x) \in \mathbb{F}_q[x, y], \tag{2.4}$$

where $f(x)$ has no repeated roots and $\deg(f) \leq 6$. In the real model, C has two different points at infinity and $h(x)$ is either 0 if p is odd or $\deg(h(x)) = 3$ if p is even. In the imaginary model C has only one point at infinity, and $\deg(f(x)) = 5$, with $h(x) = 0$ for odd p, and $\deg(h(x)) \leq 2$ otherwise.

The Frobenius endomorphism ϕ of C can be extended to its Jacobian variety. Its characteristic polynomial has the form

$$\chi(x) = x^4 - ax^3 + bx^2 - qax + q^2$$

where a and b are integers, $|a| \leq 4\sqrt{q}$ and $|b| \leq 6q$. Its roots are complex numbers $\alpha_1, \overline{\alpha_1}, \alpha_2, \overline{\alpha_2}$ such that $|\alpha_1| = |\alpha_2| = \sqrt{q}$.

From this, the cardinality of the Jacobian is

$$\#\mathrm{Jac}(C)(\mathbb{F}_q) = \chi(1) = (q^2 + 1) - a(q + 1) + b.$$

Therefore,

$$(\sqrt{q} - 1)^4 \leq \#\mathrm{Jac}(C)(\mathbb{F}_q) \leq (\sqrt{q} + 1)^4.$$

The group structure of $\mathrm{Jac}(C)(\mathbb{F}_q)$ has rank up to 4. The n-torsion subgroup of $\mathrm{Jac}(C)(\mathbb{F}_q)$ is formed by divisors $D \in \mathrm{Jac}(C)(\mathbb{F}_q)$ such that $nD = 0$, and it is denoted by $\mathrm{Jac}(C)(\mathbb{F}_q)[n]$. The l-Sylow subgroup $\mathrm{Jac}(C)(\mathbb{F}_q)[l^\infty]$ of $\mathrm{Jac}(C)(\mathbb{F}_q)$ is defined in the same way as in elliptic curves.

Imaginary model in binary fields

In binary fields, genus 2 curves can be expressed as:

$$y^2 + (h_2x^2 + h_1x + h_0)y = x^5 + f_4x^4 + f_3x^3 + f_2x^3 + f_1x + f_0, \quad h_i, f_i \in \mathbb{F}_{2^m}.$$

As already stated, since the field has even characteristic, $h(x) \neq 0$. Depending on the coefficients of h, these curves can be divided into three types, following the notation from [CY02]:

- Type I: $h_2 \neq 0$.
- Type II: $h_2 = 0, h_1 \neq 0$.
- Type III: $h_2 = h_1 = 0, h_0 \neq 0$.

Since we are interested in curves for cryptosystems based on the Discrete Logarithm problem, type III curves are not interesting, since there is a result by Galbraith [Gal01] proving that a characteristic 2 hyperelliptic curve is of type III if and only if it is supersingular.

Using isomorphisms, curves of type II can be written as

$$y^2 + xy = x^5 + f_3x^3 + f_2x^2 + f_0, \quad f_2 \in \mathbb{F}_2,$$

and if $f_2 = 1$, then the Jacobian of such a curve has order $2 \cdot n$, where n is an odd number [BT08]. Moreover, it was concluded in [BD04] that curves of this form are the best for cryptographic use out of the three, both from the arithmetic and security point of view.

Therefore, this is the type of curves that we used in one of our contributions, since appropriate Jacobians should have a large prime in their cardinality and a small cofactor, and it is desirable that this cofactor is exactly 2; this is also helpful in our contribution, which presents a method to compute halvings of a divisor.

2.2 Scalar multiplication using halvings

In elliptic curve cryptography over binary fields, Knudsen [Knu99] and Schroeppel [Sch00] proposed an alternative method which replaces the doubling of points for halvings. Since Cantor's algorithm [Can87] for doubling a divisor is less efficient for curves of genus 2 over binary fields than its equivalent for elliptic curves, the alternative of using halve-and-add algorithms for computing the multiple of a divisor, instead of the classic double-and-add, is more interesting. Some existing algorithms for computing halvings in genus 2 curves or halve-and-add can be found in [FHLM04, KKT05, Bir06, BT08, MMPR09].

2.2.1 Binary field arithmetic

When performing computations in binary fields, some operations are quite different, and have different complexities. For instance, the cost of element addition is negligible and if the field $\mathbb{F}_{2^m}$ is defined by a square-root friendly polynomial, the square and square-root operations are very cheap and efficient, much more than field multiplications. Inversions remain the most expensive operation, costing several times more than a multiplication.

Here are some results concerning the resolution of quadratic equations in a binary field, which are necessary in our contribution to compute the halving of a divisor. Let $a, b \in \mathbb{F}_{2^m}$:

- Every element in $\mathbb{F}_{2^m}$ is a quadratic residue: $a^{2^m} = a$, therefore if $x^2 = a$, then $x = a^{2^{m-1}}$.

- The Trace function is $Tr(a) = \sum_{i=0}^{m-1} a^{2^i}$, and assuming m odd, the half-trace function is $HT(a) = \sum_{i=0}^{(m-1)/2} a^{2^{2i}}$ [FHLM04].

- The equation $x^2 + x + a = 0$ has a solution in $\mathbb{F}_{2^m}$ if and only if $Tr(a) = 0$, and its solution is $x = HT(a)$. The other solution, if it exists, is $x' = x + 1$.

- An equation $ax^2 + x + b = 0$ can be solved similarly, considering the change of variables $y = ax$ and $d = ab$, solving then $y^2 + y + d = 0$, obtaining $y = HT(d)$, and finally $x = \frac{y}{a}$. The other solution is $y' = y + 1$, and therefore $x' = \frac{y}{a} + \frac{1}{a}$.

2.2.2 Non-Adjacent Form

In order to use halvings for scalar multiplication, we first need a different representation of the scalar k, called *Non-Adjacent Form* (NAF), or more generally, a NAF_ω, with an ω-wide window. This is a representation where, of every ω digits, at most one is different from 0. Apart from 0, the other digits are odd and can be negative, and their absolute value is less than $2^{\omega-1}$. All this assures a unique representation of any positive integer while minimizing the Hamming weight of the value. The classic algorithm for obtaining the corresponding representation is presented in Algorithm 2.

Algorithm 2 Compute $\text{NAF}_\omega(k)$

Input: A positive integer k, window width ω.
Output: The representation $\text{NAF}_\omega(k)$

```
1: i = 0
2: while k ≥ 1 do
3:     if k is odd then
4:         k_i = k (mod 2^ω)
5:         k = k − k_i
6:     else
7:         k_i = 0
8:     k = k/2
9:     i = i + 1
10: return (k_{i−1}, k_{i−2}, ..., k_1, k_0)
```

2.2.3 Halve-and-add algorithm

Suppose we have a curve C defined over $\mathbb{F}_q$ with $n = \#\mathrm{Jac}(C)(\mathbb{F}_q)$, $t = \lfloor \log_2 n \rfloor + 1$. We want to obtain kD using halvings, where $D \in \mathrm{Jac}(C)(\mathbb{F}_q)$ and $0 \leq k \leq n$. We first need the representation $k' = \mathrm{NAF}_\omega(2^t k \pmod n)$. Note that

$$k' = \sum_{i=0}^{t} k'_i 2^i, \quad |k'_i| < 2^{\omega-1}.$$

Then we can see that

$$k = \sum_{i=0}^{t} \frac{k'_{t-i}}{2^i} \pmod n = \frac{k'_0}{2^t} + \cdots + \frac{k'_{t-1}}{2} + k'_t \pmod n.$$

Finally the scalar multiplication can be simply computed as

$$kD = \frac{k'_0}{2^t} D + \cdots + \frac{k'_{t-1}}{2} D + k'_t D.$$

Algorithm 3 shows how to perform a scalar multiplication using the right-to-left version of the halve-and-add algorithm, using the windowed Non-Adjacent-Form NAF_ω. There is also the left-to-right version, where the result is obtained after finishing the while loop immediately, but it needs some precomputed divisors.

2.2.4 Halving in genus 2

The halving algorithm reverses the doubling algorithm by finding one of the preimages of doubling. We want to find D_1 such that

$$2D_1 = D_2,$$

where $D_2 = (x^2 + u_{21}x + u_{20}, v_{21}x + v_{20})$, $D_1 = (x^2 + u_{11}x + u_{10}, v_{11}x + v_{10})$. Following [KKT05], if the reduction step in the doubling algorithm is "undone", it leads to an equality between coordinates of "unreduced" divisors. For $2D_1$ these are $((x^2 + u_{11}x + u_{10})^2, s_3x^3 + s_2x^2 + s_1x + s_0)$ for certain indeterminates s_i. The method consists in considering an unreduced divisor $D'_2 = (u'_2(x), v'_2(x))$ corresponding to D_2, obtained using an auxiliary

Algorithm 3 Scalar multiplication using halve-and-add and NAF_ω (right-to-left)

Input: Window ω, integer k, divisor D, curve C, Jacobian cardinality n
Output: The divisor kD

1: $t = \lfloor \log_2 n \rfloor + 1$
2: $k' = \mathrm{NAF}_\omega(2^t k \pmod n)$ ▷ Note that $k' = \sum_{i=0}^{t} k'_i 2^i, \quad |k_i| < 2^{\omega-1}$
3: **for** $d \in l = \{1, 3, \ldots, 2^{\omega-1} - 1\}$ **do**
4: $\quad Q_d = P_\infty$
5: $i = t$
6: **while** $i \geq 0$ **do**
7: $\quad$ **if** $k'_i > 0$ **then**
8: $\quad\quad Q_{|k'_i|} = Q_{|k'_i|} + D$
9: $\quad$ **else if** $k'_i < 0$ **then**
10: $\quad\quad Q_{|k'_i|} = Q_{|k'_i|} - D$
11: $\quad D = \dfrac{D}{2}$
12: $\quad i = i - 1$
13: **return** $Q = \sum_{d \in l} dQ_d$

polynomial $k(x) = k_1 x + k_0$, and equating the first components of $2D_1$ and D'_2.

Note that if D_1 is a half divisor of D_2, the remaining halved divisors of D_2 can be found by computing $D_1 + W$, where W is a divisor of order 2.

We show here the method from [KKT05] for weight 2 divisors only.

Even characteristic

For $q = 2^m$, C is a genus 2 curve over $\mathbb{F}_q$ with one point at infinity P_∞, and therefore C has an imaginary model

$$C : y^2 + h(x)y = f(x),$$

where $f(x) = f_5 x^5 + f_4 x^4 + ... + f_0 \in \mathbb{F}_q[x]$, $\deg(f) \leq 5$, and $h(x) = h_2 x^2 + h_1 x + h_0 \in \mathbb{F}_q[x]$, $h(x) \neq 0$.

In this case, to obtain the divisors $D_1 = (u(x), v(x))$ such that $2D_1 = D_2$,

with $D_2 = (u_2(x), v_2(x))$, notice that the coordinates of an unreduced D_2'

$$\begin{aligned} v_2'(x) &= v_2(x) + h(x) + k(x)u_2(x) \\ u_2'(x) &= \frac{f(x) + h(x)v_2'(x) + v_2'(x)^2}{u_2(x)} \end{aligned}$$

Equating the first coordinate of D_2' with the first coordinate of $2D_1$, which is $(x^2 + u_{21}x + u_{20})^2$, we see that

$$\begin{aligned} k_1h_2 + u_{21}k_1^2 + 1 &= 0, \\ k_1h_1 + k_0h_2 + k_1^2u_{20} + k_0^2 + c_2 &= u_{11}^2k_1^2, \\ k_1h_0 + u_{21}k_0^2 + u_{21}^2 + k_0h_1 + c_1 &= 0, \\ k_0h_0 + k_0^2u_{20} + c_0 &= u_{10}^2k_1^2. \end{aligned}$$

where

$$\begin{aligned} c_2 &= f_4 + u_{21}, \\ c_1 &= f_3 + h_2v_{21} + u_{20} + c_2u_{21}, \\ c_0 &= f_2 + h_2v_{20} + h_1v_{21} + v_{21}^2 + c_2u_{20} + c_1u_{21}. \end{aligned}$$

Odd characteristic

In this case, following [MPR09], the unreduced divisor $D_2' = (u_2'(x), v_2'(x))$ of D_2 can be expressed as :

$$\begin{aligned} v_2'(x) &= v_2(x) + k(x)u_2(x), \\ u_2'(x) &= \frac{v_2'(x)^2 - f(x)}{k_1^2u_2(x)}. \end{aligned}$$

It follows that $u_1(x)^2 = x^4 + 2u_{11}x^3 + (2u_{10} + u_{11}^2)x^2 + 2u_{10}u_{11}x + u_{10}^2$ must be equal to

$$\begin{aligned} u_2'(x) = \; & x^4 + \tfrac{1}{k_1^2}[x^3(-1 + 2k_0k_1 + u_{21}k_1^2) + \\ & x^2(k_0^2 + u_{20}k_1^2 + u_{21} + 2k_0u_{21}k_1 + 2v_{21}k_1) + \\ & x(-f_3 + u_{20} + 2k_0u_{20}k_1 + k_0^2u_{21} - u_{21}^2 + 2v_{20}k_1 + 2k_0v_{21}) + \\ & (-f_2 + k_0^2u_{20} + f_3u_{21} - 2u_{20}u_{21} + u_{21}^3 + 2k_0v_{20} + v_{21}^2)]. \end{aligned}$$

Equating u_{11} and u_{10} from the degree 3 and 2 terms respectively one obtains

$$\begin{aligned} u_{11} &= \tfrac{1}{2k_1^2}(k_1^2 u_{21} + 2k_0k_1 - 1) \\ u_{10} &= \tfrac{1}{8k_1^4}(-k_1^4(u_{21}^2 - 4u_{20}) + k_1^3(4k_0u_{21} + 8v_{21}) + 6k_1^2u_{21} + 4k_1k_0 - 1), \end{aligned} \tag{2.5}$$

and substituting the values of u_{10} and u_{11} above into the degree 1 and 0 monomials and clearing denominators, we find two multivariate polynomials of degrees $2, 6$ and $2, 8$ in k_0 and k_1 respectively.

$$\begin{aligned} p_1(k_0, k_1) = \; & k_1^6(u_{21}^2 - 4u_{20})u_{21} + 2k_1^5(8v_{20} - 4u_{21}v_{21} - k_0(u_{21}^2 - 4u_{20}))+ \\ & k_1^4(12u_{20} - 15u_{21}^2 - 8f_3) + 4k_1^3(2v_{21} - 3k_0u_{21})+ \\ & k_1^2(7u_{21} - 8k_0^2) + 6k_1k_0 - 1 \end{aligned}$$

$$\begin{aligned} p_2(k_0, k_1) = \; & k_1^8(u_{21}^2 - 4u_{20})^2 + 8k_1^7(u_{21}^2 - 4u_{20})(k_0u_{21} + 2v_{21})+ \\ & 4k_1^6(-4k_0^2(u_{21}^2 - 4u_{20}) + 16k_0(2v_{20} - u_{21}v_{21}) + 16f_3u_{21}- \\ & 16f_2 + 19u_{21}^3 - 44u_{20}u_{21}) - 8k_1^5(5k_0u_{21}^2 + 4k_0u_{20} + 12u_{21}v_{21})+ \\ & 2k_1^4(4u_{20} - 32k_0v_{21} - 16k_0^2u_{21} - 19u_{21}^2) + 8k_1^3(2v_{21} - 5k_0u_{21})+ \\ & 4k_1^2(3u_{21} - 4k_0^2) + 8k_1k_0 - 1 \end{aligned}$$

Finally, with them we obtain the degree 16 polynomial

$$p_{D_2}(x) := Res(p_1(k_0, x), p_2(k_0, x)),$$

in the variable k_1 alone, by computing the resultant of $p_1(k_0, k_1)$ and $p_2(k_0, k_1)$ with respect to k_0. This resultant is a degree 16 polynomial.

$$\begin{aligned} p_{D_2}(x) = \; & x^{16}(u_{21}^2 - 4u_{20})^5 + 16\,x^{15}(u_{21}^2 - 4u_{20})^4v_{21}+ \\ & 8\,x^{14}(u_{21}^2 - 4u_{20})^3(8f_2 - 12f_3u_{21} + 20u_{20}u_{21} - 15u_{21}^3)+ \\ & \vdots \\ & 16\,x^3(20u_{21}v_{20} + 16f_3v_{21} - 20u_{20}v_{21} + 35u_{21}^2v_{21})+ \\ & 8\,x^2(8f_2 - 12f_3u_{21} + 20u_{20}u_{21} - 15u_{21}^3)+ \\ & 16\,x\,(2v_{20} - u_{21}v_{21}) + (u_{21}^2 - 4u_{20}) \end{aligned}$$

Substituting the k_1 values in 2.5 by the roots of this polynomial the divisors D_1 are obtained such that $2D_1 = D_2$.

The degrees of the irreducible factors of $p_{D_2}(x)$ depend on the degrees of the irreducible factors of $f(x)$, which can be found in Table 2.1.

Factorization type of $f(x)$	Factorization types of $p_{D_2}(x)$
$[1,1,1,1,1]$	$[\underbrace{1,...,1}_{16}], [\underbrace{2,...,2}_{8}]$
$[1,1,1,2]$	$[\underbrace{1,...,1}_{8},2,2,2,2], [\underbrace{2,...,2}_{8}], [4,4,4,4]$
$[1,1,3]$	$[1,1,1,1,3,3,3,3], [2,2,6,6]$
$[1,2,2]$	$[1,1,1,1,6,6], [4,4,4,4]$
$[2,3]$	$[1,1,2,3,3,6], [4,12]$
$[1,4]$	$[1,1,2,4,4,4], [8,8]$

Table 2.1: Factorization types of $p_{D_2}(x)$

The equivalent polynomial $p_{D_2}(x)$ for weight 1 divisors can be obtained by a similar process.

2.3 Cryptographic protocols

In this section we introduce protocols that are necessary in our contributions and some of their properties. Firstly, in Section 2.3.1 some miscellaneous properties are introduced. Then, in Section 2.3.2 the ElGamal cryptosystem is explained, including a threshold variant, which is used in [GLMS] and [GMMS]. Digital signatures using elliptic curves and blind signatures using RSA can be found in Sections 2.3.3 and Sections 2.3.4 respectively, which are used in [GMS17]. Finally, a message authentication code using hashes can be found in Section 2.3.5, also used in [GMS17].

2.3.1 Some general properties

Here we give a brief introduction to some properties that a cryptographic protocol may have and that can be helpful to better understand our contri-

butions.

In public key cryptography, a protocol has *provable security* if its security requirements can be stated formally in an adversarial model, with some assumptions such that the adversary has access to the system and enough computational power. The security is proven by showing that, in order to break the security of the protocol, the attacker must solve an underlying problem that is considered to be hard. This is a useful way to prove the security of a system.

This concept was first introduced in [SM84] for *semantic security*, which formally means that, given a probabilistic, polynomial-time algorithm (PPTA) and a ciphertext of some message and its length, the algorithm cannot obtain any partial information about the message with higher probability than any other PPTA that only has the message length and not the ciphertext. In summary, information cannot be feasibly extracted from the ciphertext.

It is important to distinguish the difference between the concepts of anonymity and privacy. *Anonymity* refers to the possibility to communicate a message, hiding who the sender is, so that people knows some information without revealing the author of such communication. An example could be a political dissident that posts an anonymous blog online.

Privacy is about hiding some information, regardless that the author is known [Bra14]. This includes many scenarios, like email, private messages, web browsing or even personal habits: there is some information that is to be kept secret from other people or entities, even if its owner or sender is known.

2.3.2 ElGamal cryptosystem

ElGamal cryptosystem [ElG85] is an asymmetric key probabilistic algorithm, whose security depends upon the difficulty of computing discrete logarithms over a cyclic group, known as the *Discrete Logarithm Problem* (DLP).

Discrete Logarithm Problem

Let G be a cyclic group of order q and let g be a generator of G, and denote its group operation by multiplication. The discrete logarithm problem is as

follows:

Given an element $y \in G$, find an integer x such that $y = g^x$.

Solving a discrete logarithm is in general considered to be computationally intractable, if the group is chosen carefully. Especially, the group order q must have a large prime factor to prevent the use of the Pohlig-Hellman algorithm [PH78], a special-purpose algorithm for computing discrete logarithms in a finite abelian group whose order is a smooth integer that reduces the DLP over G into small instances over subgroups whose orders are factors of q.

Some groups that are used to implement the ElGamal cryptosystem are:

- The multiplicative group $\mathbb{F}_{p^m}^*$, with p prime, where $p^m - 1$ has a large prime factor q.
- The group of points of an elliptic curve $E(\mathbb{F}_{p^m})$, with a subgroup of prime order q.
- The Jacobian of a genus 2 curve C, with a subgroup of $\mathrm{Jac}(C)(\mathbb{F}_{p^m})$ with prime q.

The size of the group used is determined by the difficulty of solving the discrete logarithm in such group, depending on the best known algorithm. The best known general algorithm for computing discrete logarithms on any cyclic group is Pollard's rho [Pol78], with a time complexity of $O(\sqrt{q})$. In the case of $\mathbb{F}_{p^m}^*$, with p prime, the Index Calculus [COS86, How98] takes subexponential time. This algorithm is not applicable in general to other groups, however the MOV [MOV93] attack transfers the ECDLP in $E(\mathbb{F}_{p^m})$ to the DLP in $\mathbb{F}_{p^{km}}^*$, with k being the embedding degree. Recently some weaknesses [AMORH13, AMORH15] have been found in some supersingular elliptic curves used to implement pairing-based cryptosystems [Sha84, BF01].

For a 128-bit security level, that is, so that it would take 2^{128} operations to crack the key, an elliptic curve $E(\mathbb{F}_{p^m})$ with $p^m \approx 2^{256}$ is needed, while the key size required for $\mathbb{F}_{p^m}^*$ is 3072 bits. In the case of a Jacobian of a genus 2 curve $\mathrm{Jac}(C)(\mathbb{F}_{p^m})$, since its cardinality is of the order $(p^m)^2$, just 128 bits would be enough. In Table 2.2 the security equivalence among different public key cryptosystems and their key size in bits can be found.

DLP & RSA (bits)	ECDLP (bits)	HECDLP (bits)
1024	160	80
2048	224	112
3072	256	128
7680	384	192
15360	512	256

Table 2.2: NIST guidelines for security equivalence

In the following description of the ElGamal cryptosystem's steps [HMV06], multiplicative notation is used. Note however, that for elliptic and genus 2 curves, the same steps apply but with additive notation instead. The setup and key generation steps only have to be performed once.

Setup

A group G is chosen (a subgroup of $\mathbb{F}_p^*$ or$\mathbb{F}_{2^m}^*$, the points of an elliptic curve or the Jacobian of a genus 2 curve), with order a big prime q, and a generator $G = \langle g \rangle$. The values q, G and g are published.

Key generation

Each user generates her own private key by taking a random integer such that $x \in_R [1, q-1]$, and computes and publishes the corresponding public key $y = g^x$.

Encryption

The message to be sent is converted into an element of the group $m \in G$, and encrypted under the public key y as

$$E_y(m) = (c, d) = (g^r, m \cdot y^r),$$

with $r \in_R [1, q-1]$ being a random integer.

Decryption

Given a ciphertext (c, d), the original message is recovered using the corresponding private key x as

$$m = d \cdot c^{-x}.$$

Homomorphic property of ElGamal

In an homomorphic cryptosystem [RAD78], there exist at least two binary operations, that we denote by $\otimes$ and $\oplus$ (which may or may not be the same), where $\otimes$ is defined over the set of ciphertexts and $\oplus$ is defined over the cleartext space. Then, given two messages m_1 and m_2 encrypted under the same key y, we have that

$$E_y(m_1) \otimes E_y(m_2) = E_y(m_1 \oplus m_2).$$

This property allows computations to be performed on the ciphertext without decrypting the messages, and depending on the operation $\oplus$, a system can have for example, multiplicative or additive homomorphism. Some notable cryptosystems with such property are ElGamal and Paillier [Pai99].

ElGamal is a multiplicative homomorphic cryptosystem in which $\otimes$ is the component-wise product of the ciphertext, and $\oplus$ is the product of cleartext messages:

$$E_y(m_1) \otimes E_y(m_2) = E_y(m_1 \cdot m_2) = (g^{r_1+r_2}, m_1 \cdot m_2 \cdot y^{r_1+r_2}).$$

Over elliptic curves, it becomes additively homomorphic with $\otimes$ representing the component-wise addition, obtaining the encryption of the addition of the two points or divisors.

The ElGamal over $\mathbb{F}_p^*$ can be modified in order to provide additive homomorphism, by encrypting g^m instead of m. The corresponding ciphertext, would then be:

$$E_y(g^{m_1}) \otimes E_y(g^{m_2}) = E_y(g^{m_1+m_2}) = (g^{r_1+r_2}, g^{m_1+m_2} \cdot y^{r_1+r_2}).$$

Note that in this case, in order to retrieve the value $m_1 + m_2$, a discrete

logarithm must be performed. If the value falls in a known, relatively small range $[a, \ldots, b]$, it can be performed efficiently using Pollard's lambda algorithm [PH78], with an average complexity cost of $O(\sqrt{b-a})$.

Threshold ElGamal

The n-out-of-n threshold ElGamal is a variation in which the secret key is distributed (and may be generated) among n users, and all n users need to collaborate to decrypt a message [Ped91, Des92].

The setup and encryption steps remain unchanged, and the rest are modified as follows:

- *Key generation*: each user U_i generates (or receives from a trusted third party) a private and public key pair, $y_i = g^{x_i}$, and makes y_i public together with a zero-knowledge proof on $\log_g y_i$. The shared public key is computed as $y = \prod_{i=1}^{n} y_i$.

- *Decryption*: Given a ciphertext $E_y(m) = (c, d)$, each user U_i computes a *partial decryption* $T_i = c^{x_i}$ and sends it to all the users that are allowed to obtain the cleartext, that finally can obtain the plain text by computing $m = d \cdot (\prod_{i=1}^{n} T_i)^{-1}$.

2.3.3 Elliptic Curve Digital Signature Algorithm

The Elliptic Curve Digital Signature Algorithm (ECDSA) [ANS99] is a variant of the Digital Signature Algorithm (DSA) which uses elliptic curve cryptography. A digital signature is a scheme for proving the authenticity of a digital message. Some desired properties of a digital signature are:

- *Integrity*: to ensure that the message has not been altered in any way during transmission.

- *Authentication*: the recipient has reasons to believe that the message was created by a known sender.

- *Non-repudiation*: only the sender (signer) could have validly signed a message, so it cannot later deny having signed it.

A digital signature scheme typically consists of 3 algorithms: key generation, signature generation and signature verification.

Key Generation

Starting with the Setup step from Section 2.3.2, using elliptic curves: consider a group $G = E(\mathbb{F}_{p^m})$, with a subgroup of prime order q, a point $P \in E(\mathbb{F}_{p^m})$ or order q.

A random secret key $x \in_R [1, q-1]$ is selected, with the public key being $Q = xP$.

Signature generation

The steps required to sign a message M are the following:

1. Calculate $h = \mathrm{H}(M)$, where H is a cryptographic hash function that should output the same amount of bits as the bitlength of q (since the overall security of the signature scheme will depend on the smallest of the two values). The value h shall be converted to an integer.
2. Select a cryptographically secure random integer $k \in_R [1, q-1]$.
3. Compute the point $kP = (x_1, y_1)$.
4. Compute $r = x_1 \pmod q$. If $r = 0$, go back to step 3. Note that this is an utterly improbable occurrence.
5. Compute $s = (h + xr)k^{-1} \pmod q$. If $s = 0$, go back to step 3. This is similarly improbable.
6. The pair (r, s) is the signature of M.

Signature verification

Given the public key Q, the verifier first checks that $Q \in E(\mathbb{F}_{p^m})$, $Q \neq \mathcal{O}$ and $qQ = \mathcal{O}$. Then the verification steps are:

1. Check that both r and s are integers that lie in the interval $[1, q-1]$, otherwise reject the signature.

2. Compute $h = \mathrm{H}(M)$, using the same hash function as in the signature generation.

3. Compute $w = s^{-1} \pmod{q}$.

4. Compute $u_1 = hw \pmod{q}$ and $u_2 = rw \pmod{q}$.

5. Compute the point $(x_1, y_1) = u_1 P + u_2 Q$. If the resulting point is $\mathcal{O}$, the signature is invalid.

6. The signature is valid if and only if $r \equiv x_1 \pmod{q}$.

It is of paramount importance that the value k is not only secret and with good randomness, but different for different signatures. Otherwise, given two signatures (r, s) and (r, s') with the same unknown k for different known messages M and M', and attacker could obtain the private key x.

A possible approach to ensure that k is unique and different for every message is to generate deterministic signatures, by obtaining the value k using an HMAC (see Section 2.3.5) with the private key and the message, as defined in [Por13].

2.3.4 Blind signatures

A *blind signature* is a special digital signature in which a message is in some way disguised (blinded) before it is signed such that it can be publicly verified against the original, unblinded message, as it would be done with a regular digital signature. This kind of signatures are usually employed in privacy-related protocols where the author of the message and the signer are different entities, for example, in e-voting protocols or digital cash. This concept was first introduced in [Cha83].

In a more formal way, it can be described as a cryptographic protocol involving two entities, a user U that wants to obtain a signature on a message m, and a signer S with the secret key for signing, who signs m but without her learning anything about the message.

The main security requirements of a blind signature scheme are blindness and unforgeability. The *blindness* condition implies that it must be infeasible

for a malicious signer S^* to decide which of two messages m_0 and m_1 has been signed first in two executions with an honest user U. For *unforgeability*, an efficient adversary U^* should not be able to generate $k+1$ valid signature pairs with different messages after at most k completed interactions with the honest signer [PS00]. Another, mores strict definition, *honest-user unforgeability* is proposed in [SU12], that states that after performing k interactions with the signer, an adversary that requests signatures for the messages $m_1, \cdots, m_n$ from the user (which produces these signatures by interacting with the signer), then the adversary cannot produce signatures for pairwise distinct message $m_1^*, \cdots, m_{k+1}^*$ with $\{m_1^*, \cdots, m_{k+1}^*\} \cap \{m_1, \cdots, m_n\} = \varnothing$.

There are many blind singing schemes using different cryptosystems, but here we are going to introduce only the one based on RSA, as it is the one used in one of our contributions.

RSA blind signature

RSA [RSA78] is a well known public key cryptosystem, the security of which holds on the assumed intractability of the Integer Factorization Problem.

Key generation

Following an RSA key generation, we have the public modulus $N = pq$, with p and q secret primes of similar size, the public key e such that $1 < e < \lambda(N)$ and $\gcd(e, \lambda(N)) = 1$, and the private key $d \equiv e^{-1} \pmod{\lambda(N)}$, where $\lambda(N)$ is Carmichael's totient function, which is defined as the smallest positive integer t such that

$$a^t \equiv 1 \pmod{N}, \quad \text{for every integer } a \text{ coprime with } N.$$

In other words, it is the exponent of the multiplicative group of integers modulo N, and it divides the order of the group, $\phi(N)$. It can be seen as a sharpening of Euler's theorem.

Blind signature generation

A traditional RSA signature would be computed as $m^d \pmod N$. In the blinded version, a random r is chosen such that is coprime with N, its encryption

$$r^e \pmod N$$

computed, and then used as a a blinding factor. The user with the message m to be signed computes the blinded message as

$$m' \equiv mr^e \pmod N$$

and then sends m' to the signer, who then calculates the blinded signature as:

$$s' \equiv (m')^d \pmod N.$$

The blinded signature is sent back to the user, who then removes the blinding factor to obtain s, the actual valid RSA signature of m:

$$s \equiv s' \cdot r^{-1} \pmod N.$$

Signature verification

The signer, to accept the validity of a signature, it just needs to check whether

$$s \equiv m^d \pmod N.$$

Notice that since RSA keys satisfy $r^{ed} \equiv r^{ed \pmod{\lambda(N)}} \equiv r \pmod N$, the following shows that s is indeed the signature of the message m:

$$s \equiv s' \cdot r^{-1} \equiv (m')^d \cdot r^{-1} \equiv m^d r^{ed} r^{-1} \equiv m^d r r^{-1} \equiv m^d \pmod N.$$

The main advantage of using RSA to perform a blind signature is that it is simpler to implement and faster to execute than, for example, the equivalent blind signature using elliptic curve cryptography. However, it is important to remark that, since the signing process is equivalent to that of decrypting, an attacker that intercepts a ciphered message may blind it and send it to

sign, and in turn it would be able to obtain the corresponding plain text. Therefore, it is essential that the same keys are never used to both encrypt and sign messages.

2.3.5 Hash-based Message Authentication Codes

A hash-based message authentication code (HMAC) [BCK96] is a keyed cryptographic one-way hash function, used for calculating an authentication code of a message m given a secret key K. We denote by $\text{HMAC}_K(m)$ the resulting code.

It shares the main properties of a hash (one-way function and collision-secure) as well as incorporating the property that $\text{HMAC}_K(m)$ can only be computed if the secret key K and the message m are known. Therefore, for an HMAC to be secure, it is required that given $\text{HMAC}_K(m)$ and K, it is infeasible to obtain m, and given $\text{HMAC}_K(m)$ and m, it is also infeasible to find K.

The cryptographic strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output, and the size and quality of the key.

The message must be sent together (encrypted or not) with $\text{HMAC}_K(m)$, and then, whoever has the key K, can validate the message's *integrity* (that it has not been modified during transmission) and *authenticity* (since the code can only have been computed by someone who has the secret key).

The following definition, taken from the RFC 2104 standard, [KCB97], defines how to compute an $\text{HMAC}_K(m)$:

- Denote by B the byte-length of the blocks of data of the hash function H (for MD5, SHA-1 and SHA-256, $B = 64$) and by L the byte-length of the hash output (for SHA-256, $L = 32$).

- The secret key K can be of any length but it needs to be converted to a B bytes key K':

  - If K is smaller than B bytes, append zero bytes 0x00 to the end of K to obtain K'.

  - If K is bigger, then $K' = H(K)$.
  - Otherwise, $K' = K$.

- We define two fixed different numbers, ipad and opad ('i' and 'o' for inner and outer):
  - $ipad$ = the byte 0x36 repeated B times.
  - $opad$ = the byte 0x5c repeated B times.

Finally, to compute the authentication code of a message m:

$$\mathrm{HMAC}_K(m) = H\Big((K' \oplus opad) \ \| \ H(\ (K' \oplus ipad) \| m\ ) \Big),$$

where $\|$ denotes concatenation and $\oplus$ denotes bitwise exclusive OR (XOR).

The exact values of *ipad* and *opad* are not critical, but were defined in such a way to have a large Hamming distance from each other and so the inner and outer keys will have fewer bits in common. They need to be different in at least one bit.

Chapter 3

Contributions

The proposals of this thesis are presented as articles published or submitted to journals or conferences, following a thematic ordering: the first two contributions are related to genus 2 curves, studying properties of the cardinality of their Jacobian varieties and offering improvements for some necessary cryptographic operations; the next one introduces a novel parking system that improves drivers' privacy; finally the last two describe enhancements on privacy on some smart metering systems.

A brief description of each article is presented next, highlighting the most relevant aspects of our contributions. For copyright reasons, the articles can not be shown in this thesis.

1. [GMPT18] Ricard Garra, Josep M. Miret, Jordi Pujolàs, and Nicolas Thériault. "The 2-adic valuation of the cardinality of Jacobians of genus 2 curves over quadratic towers of finite fields". Accepted for publication in *Journal of Algebra and its Applications*.

   In the first contribution we focus on genus 2 curves over fields of odd characteristic. We study how the 2-adic valuation of the cardinality of the Jacobian of a curve increases when performing successive quadratic extensions, taking into account all types of curves, including supersingular ones. By carefully looking at the curve equation $f(x)$ and the coefficients of the Frobenius endomorphism, as well as the initial 2-adic valuation, we are able to show and proof how that valuation increases at each extension, with an exact value if it exists or its bounds oth-

erwise. This information could be of use in some steps of the SEA algorithm for point counting.

2. [GMP16] Ricard Garra, Josep M. Miret, and Jordi Pujolas. "Halving in some genus 2 curves over binary fields". *IEEE Latin America Transactions*, 14(6):2885–2889, 2016.

   In the second contribution we show all the process of how we obtained explicit formulae for computing the halving of each possible divisor class, for type II genus 2 curves over binary fields. Our formulae are, to the best of our knowledge, the fastest known for the studied type of curves, improving the ones given by Birkner and Thériault in [BT08]. The halving operation is useful for computing scalar multiplication, which is the main operation in cryptographic protocols based on the discrete logarithm on algebraic curves.

3. [GMS17] Ricard Garra, Santi Martínez, and Francesc Sebé. "A Privacy-Preserving Pay-by-Phone Parking System". *IEEE Transactions on Vehicular Technology*, 66(7):5697–5706, 2017.

   In this article we study and discuss existing solutions that use a mobile phone application in order to pay for parking time, as well as their drawbacks, especially privacy-wise, and we introduce a novel system that preserves the privacy of drivers, all while maintaining the flexibility that a phone app offers and the ability to complain in case of an unfair fine, with the ability to prove that a payment for the corresponding car and time slot had been made. The privacy that the system offers ensures that an attacker (or a corrupt company running it) can not collect more information out of the system than a person patrolling the city would, checking parked cars in the street. Several protocols are used as building blocks, including elliptic curve cryptography, blind digital signatures and HMAC.

4. [GLMS] Ricard Garra, Dominik Leibenger, Josep M. Miret and Francesc Sebé. "Repairing an aggregation-based smart metering system". *Submitted for publication.*

While researching several smart metering proposals, particularly aggregation based ones, we found a security flaw in an existing protocol [BPS+16] that would allow a corrupted substation to obtain all the readings of any one individual meter in the neighborhood. We decided to repair the mentioned security flaw and modify the protocol so it can ensure meter's privacy against a corrupted substation. The reasons to repair this proposal are many, but in general we find that it has several advantages, making it a very good option among the literature, since it offers low communication costs, it does not require a trusted third party for generating or storing secret keys and it offers data integrity without digital signatures or message authentication codes.

5. [GMMS] Ricard Garra, Santi Martínez, Josep M. Miret, and Francesc Sebé. "Improving a smart metering system using elliptic curves and removing the trusted dealer". Accepted for publication in the proceedings of *Reunión Española sobre Criptologia y Seguridad de la Información 2018*.

   Finally, in the last contribution we combined different approaches in order to obtain a smart metering system with many desirable advantages: we eliminate the need of a trusted third party to establish the keys used in the protocol, only a single message is required to be sent to the substation by each smart meter when transmitting consumption readings and no communication among meters is needed, and therefore it is easily scalable to thousands of meters. It uses homomorphic addition to aggregate the readings in such a way that only the sum of the honest smart meters could be obtained by a corrupt substation. The protocol uses elliptic curve cryptography to reduce communication and key storage costs.

Chapter 4

Conclusions and future work

In this thesis we study some properties of genus 2 curves related to the cardinality of their Jacobians as well as the halving operation, due to their usefulness in cryptographic protocols used in modern society since they offer the smallest key size for a given security level compared to other options. We also address possible privacy problems that arise from new technologies used in the e-society, proposing new protocols to address those concerns.

In the elliptic and hyperelliptic versions of ElGamal, it is important to carefully choose the curve such that it meets some specific criteria, and in particular it is necessary to know the cardinality of the group of points or divisors. We started with genus 2 curves over a field with odd characteristic, and we a studied of how the 2-adic valuation of their Jacobian grows after successive quadratic extensions [GMPT18]. This can help in a step required in the SEA algorithm, which is the best known algorithm to compute the cardinality of a Jacobian to date. Since this can be a time-consuming task for cryptographic level curves, it is desirable to lower its complexity.

When a suitable curve has been found and we want to set up and use an ElGamal cryptosystem, the basic operation needed for every computation is scalar multiplication. There exist several algorithms for performing that operation, with the simplest one probably being the classic double-and-add strategy, which is equivalent to the double-and-add used to compute integer modular exponentiations. We focused on a different approach, halve-and-add, which instead of the double of a divisor, it uses its halved one: from a

divisor D, we need D' such that $2D' = D$. In [GMP16] we give improved algorithms for computing that divisor D' for all possible divisors of type II curves over binary fields. These can be used to enhance halve-and-add versions of the scalar multiplication in those curves, reducing their cost and execution time.

We then shifted our focus from more theoretic approaches to more practically applied protocols. After pointing out the main drawbacks of existing solutions for pay-by-phone parking systems, especially regarding privacy, we propose a new system that ensures the privacy of the drivers [GMS17]. By removing the possibility of collecting parking information and behavior of each car, like having a record of exactly where and when a car has been parked, the system offers a level of privacy equivalent to a person patrolling the city and seeing what cars are parked and where. Additionally, in case a parking officer issues an unfair fine, the driver has the possibility of complaining by proving that a payment had been made for the time slot of the fine.

After finding a security flaw in an existing smart metering protocol, but that otherwise had some desirable properties, we decided to present a new system that fixes the aforementioned flaw while retaining the other main properties [GLMS]. More specifically those advantages are: no trusted third party deals with or stores secret keys, and therefore the privacy can not be compromised if that party is corrupted; the communication cost per round is $O(n)$, where n is the number of smart meters, and the individual readings can only be obtained if the substation and all the other meters are corrupt; and finally it offers data integrity without the use of digital signatures or message authentication codes.

Inspired by the previous proposal and some other interesting approaches to aggregative solutions for smart metering, we devised a protocol [GMMS] that does not need a trusted third party either, and it requires only one message to be sent by the meters for each round of communication, when transmitting energy consumption readings, with no communication among meters. Additionally, the protocol uses elliptic curve cryptography to reduce even more the communication cost as well as the size of the keys.

Possible further research could be done in some related areas, like study-

ing the 2-adic valuation of genus 2 curves but over binary fields this time, or even using other primes to analyze their ℓ-adic valuation after different field extensions, like it has been done in [MPV15] for elliptic curves, which would further help to improve the efficiency of the SEA algorithm.

The proposed mobile application for pay-by-phone parking could be expanded and improved with, for example, functionalities to allow it to help to find a parking spot, like the proposal in [CMC15] but extended to outdoors.

In regards to aggregation-based smart metering systems, they have the inherent problem that if enough meters are corrupted, the system's privacy could be compromised. There are other approaches, like the ones based on perturbation that add noise to the reading before sending it, but they need to be carefully tuned so that they more or less cancel out. It could be interesting to research and develop a system like that with the benefits we achieved in our contributions, like the avoidance of a trusted third party, while providing good aggregation accuracy without sacrificing privacy nor efficiency.

Bibliography

[ÁC11] Gergely Ács and Claude Castelluccia. I have a dream! (differentially private smart metering). In *International Workshop on Information Hiding*, pages 118–132. Springer, 2011.

[AMORH13] Gora Adj, Alfred Menezes, Thomaz Oliveira, and Francisco Rodríguez-Henríquez. Weakness of $\mathbb{F}_{3^{6\cdot 509}}$ for Discrete Logarithm Cryptography. In *International Conference on Pairing-Based Cryptography*, pages 20–44. Springer, 2013.

[AMORH15] Gora Adj, Alfred Menezes, Thomaz Oliveira, and Francisco Rodríguez-Henríquez. Weakness of $\mathbb{F}_{3^{6\cdot 1429}}$ and $\mathbb{F}_{2^{4\cdot 3041}}$ for Discrete Logarithm Cryptography. *Finite Fields and Their Applications*, 32:148–170, 2015.

[ANS99] ANSI. Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). *American National Standard X9.62-1999*, 1999.

[Atk88] Arthur O L Atkin. The number of points on an elliptic curve modulo a prime. *Preprint*, 1988.

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In *Annual International Cryptology Conference*, pages 1–15. Springer, 1996.

[BD04] Bertrand Byramjee and Sylvain Duquesne. Classification of genus 2 curves over $\mathbb{F}_{2^n}$ and optimization of their arithmetic. *IACR Cryptology ePrint Archive*, 2004:107, 2004.

[BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In *Annual international cryptology conference*, pages 213–229. Springer, 2001.

[Bir06] Peter Birkner. Efficient divisor class halving on genus two curves. In *International Workshop on Selected Areas in Cryptography*, pages 317–326. Springer, 2006.

[BPS+16] Núria Busom, Ronald Petrlic, Francesc Sebé, Christoph Sorge, and Magda Valls. Efficient smart metering based on homomorphic encryption. *Computer Communications*, 82:95–101, 2016.

[Bra14] Danny Bradbury. Anonymity and privacy: a guide for the perplexed. *Network Security*, 2014(10):10–14, 2014.

[BSS99] Ian Blake, Gadiel Seroussi, and Nigel Smart. *Elliptic curves in cryptography*, volume 265. Cambridge university press, 1999.

[BSU10] Jens-Matthias Bohli, Christoph Sorge, and Osman Ugus. A privacy model for smart metering. In *Communications Workshops (ICC), 2010 IEEE International Conference on*, pages 1–5. IEEE, 2010.

[BT08] Peter Birkner and Nicolas Thériault. Faster halvings in genus 2. In *International Workshop on Selected Areas in Cryptography*, pages 1–17. Springer, 2008.

[Can87] David G Cantor. Computing in the Jacobian of a hyperelliptic curve. *Mathematics of computation*, 48(177):95–101, 1987.

[CFA+05] Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren. *Handbook of elliptic and hyperelliptic curve cryptography*. CRC press, 2005.

[Cha83] David Chaum. Blind signatures for untraceable payments. In *Advances in cryptology*, pages 199–203. Springer, 1983.

[CMC15] Cándido Caballero, Jezabel Molina, and Pino Caballero. Low-cost service to predict and manage indoor parking spaces. In *International Conference on Ubiquitous Computing and Ambient Intelligence*, pages 225–236. Springer, 2015.

[CMT05] Claude Castelluccia, Einar Mykletun, and Gene Tsudik. Efficient aggregation of encrypted data in wireless sensor networks. 3rd Intl. In *Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Sensor Networks, Italy*, 2005.

[COS86] Don Coppersmith, Andrew M Odlzyko, and Richard Schroeppel. Discrete logarithms in $GF(p)$. *Algorithmica*, 1(1-4):1–15, 1986.

[CY02] YoungJu Choie and D Yun. Isomorphism Classes of Hyperelliptic Curves of Genus 2 over $\mathbb{F}_q$. In *Australasian Conference on Information Security and Privacy*, pages 190–202. Springer, 2002.

[Des92] Yvo Desmedt. Threshold cryptosystems. In *International Workshop on the Theory and Application of Cryptographic Techniques*, pages 1–14. Springer, 1992.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[EK10] Costas Efthymiou and Georgios Kalogridis. Smart grid privacy via anonymization of smart metering data. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 238–243. IEEE, 2010.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[Elk98] Noam D Elkies. Elliptic and modular curves over finite fields and related computational issues. *AMS IP STUDIES IN ADVANCED MATHEMATICS*, 7:21–76, 1998.

[Esp] Gobierno de España. Plan de susitución de contadores. `https://www.boe.es/diario_boe/txt.php?id=BOE-A-2012-2538`. Accessed: 2018-06-01.

[EYS] EYSA Mobile. `http://www.eysamobile.com`.

[FB13] Soren Finster and Ingmar Baumgart. Pseudonymous smart metering without a trusted third party. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, pages 1723–1728. IEEE, 2013.

[FHLM04] Kenny Fong, Darrel Hankerson, Julio López, and Alfred Menezes. Field inversion and point halving revisited. *IEEE Transactions on Computers*, 53(8):1047–1059, 2004.

[Ful72] William Fulton. *Curvas algebraicas*. Reverté, 1972.

[Gal01] Steven D Galbraith. Supersingular curves in cryptography. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 495–513. Springer, 2001.

[GLMS] Ricard Garra, Dominik Leibenger, Josep M Miret, and Francesc Sebé. Repairing an aggregation-based smart metering system. *Submitted for publication.*

[GMMS] Ricard Garra, Santi Martínez, Josep M Miret, and Francesc Sebé. Improving a smart metering system using elliptic curves and removing the trusted dealer. Accepted for publication in the proceedings of *Reunión Española sobre Criptologia y Seguridad de la Información 2018.*

[GMP16] Ricard Garra, Josep M Miret, and Jordi Pujolas. Halving in some genus 2 curves over binary fields. *IEEE Latin America Transactions*, 14(6):2885–2889, 2016.

[GMPT18] Ricard Garra, Josep M Miret, Jordi Pujolàs, and Nicolas Thériault. The 2-adic valuation of the cardinality of Jacobians of genus 2 curves over quadratic towers of finite fields. *Journal of Algebra and Its Applications*, 2018.

[GMS17] Ricard Garra, Santi Martínez, and Francesc Sebé. A Privacy-Preserving Pay-by-Phone Parking System. *IEEE Transactions on Vehicular Technology*, 66(7):5697–5706, 2017.

[GS04] Pierrick Gaudry and Éric Schost. Construction of secure random curves of genus 2 over prime fields. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 239–256. Springer, 2004.

[GS12] Pierrick Gaudry and Éric Schost. Genus 2 point counting over prime fields. *Journal of Symbolic Computation*, 47(4):368–400, 2012.

[Has33] Helmut Hasse. Beweis des analogons der riemannschen vermutung für die artinschen und fk schmidtschen kongruenzzetafunktionen in gewissen elliptischen fällen. vorläufie mitteilung. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1933:253–262, 1933.

[HMV06] Darrel Hankerson, Alfred J Menezes, and Scott Vanstone. *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.

[How98] Jason S Howell. The Index Calculus Algorithm for Discrete Logarithms. Masters degree thesis, Clemson University, 1998.

[KCB97] Hugo Krawczyk, Ran Canetti, and Mihir Bellare. RFC 2104: HMAC: Keyed-Hashing for Message Authentication, 1997.

[KKT05] Izuru Kitamura, Masanobu Katagi, and Tsuyoshi Takagi. A complete divisor class halving algorithm for hyperelliptic curve cryptosystems of genus two. In *Australasian Conference on Information Security and Privacy*, pages 146–157. Springer, 2005.

[Knu99] Erik Woodward Knudsen. Elliptic scalar multiplication using point halving. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 135–149. Springer, 1999.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[Kob89] Neal Koblitz. Hyperelliptic cryptosystems. *Journal of cryptology*, 1(3):139–150, 1989.

[Kob12] Neal Koblitz. *Algebraic aspects of cryptography*, volume 3. Springer Science & Business Media, 2012.

[LLMP93] Arjen K Lenstra, Hendrik W Lenstra, Mark S Manasse, and John M Pollard. The number field sieve. In *The development of the number field sieve*, pages 11–42. Springer, 1993.

[LLZS10] Rongxing Lu, Xiaodong Lin, Haojin Zhu, and Xuemin Shen. An intelligent secure and privacy-preserving parking scheme through vehicular communications. *IEEE Transactions on Vehicular Technology*, 59(6):2772–2785, 2010.

[Mil85] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.

[MMPR09] Josep M Miret, Ramiro Moreno, Jordi Pujolàs, and Anna Rio. Halving for the 2-Sylow subgroup of genus 2 curves over binary fields. *Finite Fields and Their Applications*, 15(5):569–579, 2009.

[MMRV05] Josep M Miret, Ramiro Moreno, Anna Rio, and Magda Valls. Determining the 2-Sylow subgroup of an elliptic curve over a finite field. *Mathematics of computation*, 74(249):411–427, 2005.

[MMRV09] Josep M Miret, Ramiro Moreno, Anna Rio, and Magda Valls. Computing the ℓ-power torsion of an elliptic curve over a finite field. *Mathematics of Computation*, 78(267):1767–1786, 2009.

[MOV93] Alfred J Menezes, Tatsuaki Okamoto, and Scott A Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on information Theory*, 39(5):1639–1646, 1993.

[MPR09] Josep M Miret, Jordi Pujolas, and Anna Rio. Bisection for genus 2 curves in odd characteristic. *Proceedings of the Japan Academy, Series A, Mathematical Sciences*, 85(4):55–61, 2009.

[MPR10] Josep M Miret, Jordi Pujolàs, and Anna Rio. Explicit 2-power torsion of genus 2 curves over finite fields. *Adv. in Math. of Comm.*, 4(2):155–168, 2010.

[MPV15] Josep M Miret, Jordi Pujolàs, and Javier Valera. On the ℓ-adic valuation of the cardinality of elliptic curves over finite extensions of $\mathbb{F}_q$. *Archiv der Mathematik*, 105(3):261–269, 2015.

[MSP+13] Félix Gómez Mármol, Christoph Sorge, Ronald Petrlic, Osman Ugus, Dirk Westhoff, and Gregorio Martínez Pérez. Privacy-enhanced architecture for smart metering. *International journal of information security*, 12(2):67–82, 2013.

[MWZ96] Alfred Menezes, Yi-Hong Wu, and Robert Zuccherato. *An elementary introduction to hyperelliptic curves*. Faculty of Mathematics, University of Waterloo, 1996.

[NZLS16] Jianbing Ni, Kuan Zhang, Xiaodong Lin, and Xuemin Sherman Shen. EDAT: Efficient data aggregation without TTP for privacy-assured smart metering. In *Communications (ICC),*

2016 IEEE International Conference on, pages 1–6. IEEE, 2016.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 1999.

[Ped91] Torben Pryds Pedersen. A threshold cryptosystem without a trusted party. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 522–526. Springer, 1991.

[Pet10] Ronald Petrlic. A privacy-preserving concept for smart grids. *Sicherheit in vernetzten Systemen*, 18:B1–B14, 2010.

[PH78] Stephen Pohlig and Martin Hellman. An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance (Corresp.). *IEEE Transactions on information Theory*, 24(1):106–110, 1978.

[Pol78] John M Pollard. Monte Carlo methods for index computation (mod p). *Mathematics of computation*, 32(143):918–924, 1978.

[Por13] Thomas Pornin. RFC 6979: Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA), 2013.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of cryptology*, 13(3):361–396, 2000.

[RAD78] Ronald L Rivest, Len Adleman, and Michael L Dertouzos. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[Sch95] René Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, 7(1):219–254, 1995.

[Sch00] Richard Schroeppel. Elliptic curve point halving wins big. In *2nd Midwest Arithmetical Geometry in Cryptography Workshop, Urbana, Illinois*, 2000.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Workshop on the theory and application of cryptographic techniques*, pages 47–53. Springer, 1984.

[Sil09] Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009.

[SM84] Goldwasser Shafi and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.

[SU12] Dominique Schröder and Dominique Unruh. Security of blind signatures revisited. In *International Workshop on Public Key Cryptography*, pages 662–679. Springer, 2012.

[TFHA+11] Jonathan Taverne, Armando Faz-Hernández, Diego F Aranha, Francisco Rodríguez-Henríquez, Darrel Hankerson, and Julio López. Speeding scalar multiplication over binary elliptic curves using the new carry-less multiplication instruction. *Journal of Cryptographic Engineering*, 1(3):187, 2011.

[Thé03] Nicolas Thériault. Index calculus attack for hyperelliptic curves of small genus. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 75–92. Springer, 2003.

[Wei49] André Weil. Numbers of solutions of equations in finite fields. *Bulletin American Mathematical Society*, 55(5):497–508, 1949.

[YYRO11] Gongjun Yan, Weiming Yang, Danda B Rawat, and Stephan Olariu. SmartParking: A secure and intelligent parking system. *IEEE Intelligent Transportation Systems Magazine*, 3(1):18–30, 2011.