Identity Based Cryptography From Bilinear Pairings

Manuel Bernardo Barbosa
mbb@di.uminho.pt

TR-CCTC/DI-2005-37
June 2005

Centro de Ciências e Tecnologias da Computação
Departamento de Informática da Universidade do Minho
Campus de Gualtar — Braga — Portugal

TR-CCTC/DI-2005-37
Identity Based Cryptography From Bilinear Pairings
by Manuel Bernardo Barbosa

Abstract

This report contains an overview of two related areas of research in cryptography which have been prolific in significant advances in recent years. The first of these areas is pairing based cryptography. Bilinear pairings over elliptic curves were initially used as formal mathematical tools and later as cryptanalysis tools that rendered supersingular curves insecure. In recent years, bilinear pairings have been used to construct many cryptographic schemes. The second area covered by this report is identity based cryptography. Digital certificates are a fundamental part of public key cryptography, as one needs a secure way of associating an agent's identity with a random (meaningless) public key. In identity based cryptography, public keys can be arbitrary bit strings, including readable representations of one's identity.

Contents

Chapter 1

Introduction

This report contains an overview of two related areas of research in cryptography which have been prolific in significant advances in recent years. The first of these areas is pairing based cryptography. Bilinear pairings over elliptic curves were initially used as formal mathematical tools and later as cryptanalysis tools that rendered supersingular curves insecure. In recent years, bilinear pairings have been used to construct many cryptographic schemes, the most notorious of which is Boneh and Franklin's identity based encryption algorithm.

The second area covered by this report is identity based cryptography. Digital certificates are a fundamental part of public key cryptography, as one needs a secure way of associating an agent's identity with a random (meaningless) public key. In identity based cryptography, public keys can be arbitrary bit strings, including readable representations of one's identity. Therefore, digital certificates are not necessary in an identity based scenario. The number of identity based cryptographic primitives and schemes that have been proposed in the last five years is also astounding. Whether or not large-scale identity based cryptographic infrastructures will replace classical public key infrastructures, and in which areas this is more likely to occur, is something that remains to be seen.

This work was carried out during a four months visit with the Cryptography and Information Security Group, at the Department of Computer Science, University of Bristol.

Acknowledgement Work funded by scholarship SFRH/BPD/20528/2004, awarded by the Fundação para a Ciência e Tecnologia, Ministério da Ciência e do Ensino Superior, Portugal.

Chapter 2

Basic Notions

2.1 Hard problems

In this section, we list a set of hard problems used in cryptography. By hard we mean problems for which no polynomial time algorithm is known.

2.1.1 General Problems

Factoring large numbers *Given a large integer number n, find its prime factorisation, such that $n = \Pi p_i^{e_i}$.* The best known algorithm for factoring a number with no small prime factors is the General Number Field Sieve, which executes in sub exponential time.

RSA problem *Given an RSA public key (n, e), such that $n = p \cdot q$ is the product of two random large primes, and a ciphertext $C = M^e \pmod n$, to compute M* [39]. Clearly, this problem can be no harder than factoring the modulus, since this reveals $\phi(n) = (p-1)*(q-1)$ and allows for the calculation of the private key $d = e^{(-1)} \pmod{\phi(n)}$. The inverse relationship, i.e. is factoring the modulus any harder than the RSA problem, is an open issue. However, it is believed that the best attack on an RSA cryptosystem is to factor n.

Discrete Logarithm Problem (DLP) *Let $(G, *)$ be a multiplicative group of order n, and let $h \in G$ such that $h = g^x$ for some unknown $x \in \mathbb{Z}_n$. Given g and h, the discrete logarithm problem is to find x.* For additive groups, such as an elliptic curve, the problem definition is slightly different, since exponentiation is effectively a scalar multiplication. If $G = \mathbb{Z}_p$, where p is prime, then the best known algorithm is the index calculus method, which executes in sub-exponential time. Over elliptic curves, and until recently, no sub-exponential time algorithm was known to exist. However, recent developments, which also led to the appearence of pairing based cryptography, have shown that for particular types of curves the DLP can be reduced to the same problem over a different

group, over which the Index Calculus Method applies (see Section 4.2). P1363 [24] already includes an algorithm for avoiding such curves.

Diffie-Hellman Problem (DHP) *Let $a, b \in \mathbb{Z}_n^*$, $(G, *)$ a multiplicative group of order n, and $g \in G$. Given g, g^a, and g^b, the Computational Diffie-Hellman problem (CDHP) is to find $h \in G$ such that $h = g^{ab}$.* Clearly, this problem can be no harder than the DLP, since solving the latter for a or b automatically permits obtaining h. The reverse relation, i.e. is the DLP any harder than the CDHP, is still an open problem, but work done in this area indicates that the two problems might be equivalent.

Decision Diffie-Hellman Problem (DDHP) *Let $a, b \in \mathbb{Z}_n^*$, $(G, *)$ a multiplicative group of order n, and $g, h \in G$. Given g, g^a, g^b, and h, the Decision Diffie-Hellman problem (DDHP) is to determine whether $h = g^{ab}$.* This ia a weaker variant of the CDHP. Again, it is clear that the DDHP can be no harder than the CDHP. However, there are some groups for which it is known that the CDHP is hard, but the DDHP is easy. Such groups are called *Gap Diffie-Hellman Groups*. Nevertheless, for the general case, it is not clear whether the DDHP is always easier than the CDHP.

Gap Diffie-Hellman Problem (GDHP) *Let $a, b \in \mathbb{Z}_n^*$, $(G, *)$ a cyclic multiplicative group of prime order n, and $g \in G$. Given g, g^a, and g^b, the Gap Diffie-Hellman problem (GDHP) is to solve the CDHP, possibly with help of a decision Diffie-Hellman oracle.* Note that this problem formulation is natural in gap Diffie-Hellman groups, where the GDHP and the CDHP are equivalent, since the DDHP oracle is readily available.

2.1.2 Problems Over Bilinear Groups

Consider a map $\hat{t} : G_1 \times G_2 \rightarrow G_T$, from multiplicative groups $(G_1, *)$ and $G_2.*)$ to another multiplicative group $(G_T, *)$, with all groups of prime order q. This map is said to be a *bilinear pairing*, if it has the following properties:

- Linearity in the first argument: $\hat{t}(x * x, y) = \hat{t}(x, y) * \hat{t}(x, y)$.
- Linearity in the second argument: $\hat{t}(x, y * y) = \hat{t}(x, y) * \hat{t}(x, y)$.
- Non-degeneracy (strong): $\hat{t}(x, y) = 1 \Leftrightarrow (x = 1) \vee (y = 1)$.

If the pairing is defined such that $G_1 = G_2$ then it is said to be *symmetric*, and *asymmetric* otherwise.

Consider the simpler case of a symmetric pairing. If there is an efficient way of calculating such a bilinear map from G_1 to G_T, then the Decision Diffie-Hellman problem in G_1 is easy, although the Computational Diffie-Hellman problem may still be hard [11]. In other words, we are able to build a gap Diffie Hellman group.

This is because we can calculate $\hat{t}(g^a, g^b)$ and simply compare it to $\hat{t}(g, h)$. Due to the bi-linearity property, these values will be equal if and only if $h = g^{ab}$. Note that because G_T is cyclic, and the pairing is strongly non-degenerate, $\hat{t}(g, g)$ must be a generator for G_T.

Bilinear Diffie-Hellman Problem (BDHP) *Let $a, b, c \in \mathbb{Z}_n^*$, $(G_1, *)$ be a cyclic multiplicative group of prime order q, $(G_T, *)$ be a cyclic multiplicative group of prime order q, $\hat{t} : G_1 \times G_1 \rightarrow G_T$ be a symmetrical bilinear pairing, and $g \in G_1$ be a generator such that $\hat{t}(g, g)$ is also a generator of G_T. Given g, g^a, g^b and g^c, the Bilinear Diffie-Hellman Problem (BDHP) is to compute $e(g, g)^{abc}$.* The BDHP can be no harder than the CDHP in either group, since the ability to solve the latter in any of them provides a trivial solution to the former. See Section 6 for recent developments on this issue.

When dealing with asymmetric pairings of the form $\hat{t} : G_1 \times G_2 \rightarrow G_T$, the previous definitions of the Diffie-Hellman and Gap-Diffie-Hellman problems are not adequate [12]. This is because the domain of the pairing is formed out of two different groups. Hence, the relevant issue is not solving DH in G_1, but the following problem over G_1 and G_2.

Co-Diffie-Hellman Problem (Co-DHP) *Let $a \in \mathbb{Z}_n^*$, $(G_1, *)$ and $(G_2, *)$ be cyclic multiplicative groups of prime order q, and $g_1 \in G_1$ and $g_2 \in G_2$ be generators. Given g_1, g_1^a, g_2, g_2^a, and $h \in G_1$ the Co-Diffie-Hellman Problem (Co-DHP) is to compute h^a.*

If an isomorphism $\rho : G_2 \rightarrow G_1$ exists and is efficiently computable, which is usually the case, then this assumption can be relaxed, by ommiting g_2^a, and making $g_1 = \rho(g_2)$.

Co-Decision-Diffie-Hellman Problem (Co-DHP) *Let $a, b \in \mathbb{Z}_n^*$, $(G_1, *)$ and $(G_2, *)$ be cyclic multiplicative groups of prime order n, and $g_1 \in G_1$ and $g_2 \in G_2$ be generators. Given g_1, g_1^a, g_2 and g_2^b the Co-Decision-Diffie-Hellman Problem (Co-DDHP) is to determine if $a = b$.*

One calls a pair (G_1, G_2) a *co-Gap-Diffie-Hellman pair*, if the co-DHP is hard, and the co-DDHP is easy for these groups. If we have a bilinear map such as the one described above, then one must consider the problem of finding pre-images of pairing values.

One Way Bilinearity (OWB) *Given a random element $Q \in G_T$, find a pair (x, y) such that $\hat{t}(x, y) = Q$*[10]. It is not clear how hard this problem actually is in the case of the pairings used in identity based cryptography. However, it is definitely not harder than a stricter version of this notion in which one of the points in G_1 or G_2 if fixed.

This notion has been disregarded until recently, despite being, as Joux noted in is revised version of [27] fundamental to the understanding the security of pairing based systems.

Let us assume, momentarily, that this problem is not hard in a particular setting $(G_1, G_T, \hat{t} : G_1 \times G_1 \rightarrow G_T, P)$, where P is a generator of G_1. In other words, assume that given $g \in G_T$ we are able to compute ϕg such that $\hat{t}(P, \phi(g)) = g$.

Suppose now that we want to solve the CDH problem in G_T: we are given g^a and g^b, and we want to calculate g^{ab}. By the pairing's bilinearity, then we must have $\phi(g^a) = a\phi(g)$ and $\phi(g^b) = b\phi(g)$.

Now forgetting about P, we have $\hat{t}(\phi(g), \phi(g)) = g^\lambda$, and it is easy to calculate $g^{ab\lambda} = \hat{t}(\phi(g^b), \phi(g^a))$. If q is prime, then we have $\lambda^{q-3} = \lambda^{-2} \pmod q$. Additionally, it is easy to calculate $g^{\lambda^{q-3}}$ using an addition chain for $q-3$ and the pairing to construct the powers of g. Finally, one could calculate $g^{ab} = \hat{t}(\phi(g^{ab\lambda}), \phi(g^{\lambda^{-2}}))$.

This leads to the conclusion that the latter form of OWB is at least as hard as CDH in G_2 (and consequently G_1) when there is a bilinear pairing between the two.

2.2 Security models

Central to any cryptographic primitive or protocol is an assessment of the security guarantees that it provides. The theoretical validation of these properties based on formal security requirements and adversary models, anchoring the proofs of security on the assumption that it is unfeasible to solve a hard problem is usually called *provable security*.

Provable security gained strength in public key cryptography, where the internal structure of cryptographic primitives is a lot simpler, and security conjectures are usually based on arguments such as "assuming that the discrete logarithm problem is hard".

The first work in this area introduced security proofs very similar to complexity theory reductions, which in addition to being quite elaborate, were only applicable to cryptographic schemes that were not usable in practice.

In 1998. Bellare et al. [9] introduced the Random Oracle model, which allows for simpler proofs, and simpler schemes. In this model, all parties have access to a random function (the Random Oracle). Proofs carried out Random Oracle model only apply to the real world if one assumes that cryptographic hash functions are a valid implementation of a truly random function (or if one considers only attacks where hash functions are viewed as random functions).

Today, opinions diverge as to the validity of proofs obtained in the RO model, although the majority of cryptosystems for which security proofs are presented are secure only in this setting. Nevertheless, it is generally accepted that a proof in the *standard model* is preferrable, but if it is not attainable, a proof in the RO model is better than no proof at all.

Public Key Encryption In [5], Bellare et al. sum up previous work to establish the notions of security for public key encryption and the relations between these notions. [1]

The security of a public key encryption scheme is defined along two different axis: one of them measures the security requirements for the scheme, or goals; on the other one are the capabilities of the adversary, i.e. the nature of the attack. Regarding the goals, we can have:

- *Indistinguishability (IND)* – this notion captures the intuition of a privacy requirement. In a scheme satisfying this requirement it is unfeasible for an adversary, given a cryptogram created from one of two possible clear texts, to distinguish which clear text was actually used. This is also called *polynomial security*, and it has been shown to be equivalent to *semantic security*[2].

- *Non-malleability(NM)* – this notion captures the intuition that a cryptographic scheme must be tamper-proof. Given a cryptogram, an adversary must be unable to produce another one, such that the corresponding clear text is *meaningfully-related* with the original one (unknown to the adversary).

Three types of attack are usually considered:

- *Chosen Plaintext Attack (CPA)* – The adversary can obtain ciphertexts for plaintexts of her choice. In public key cryptography this is always possible in practice, due to the public knowledge of the encryption key.

- *Non-adaptive Chosen Ciphertext Attack (CCA1)* – Same as CPA, but the adversary has the additional capability of obtaining decryptions of chosen ciphertexts, at some point prior to the time of the actual attack.

- *Adaptive Chosen Ciphertext Attack (CCA2)* – Same as CCA1, only that the adversary has access to the decryption oracle at all times during the attack (obviously it cannot ask for the decryption of the ciphertext on which she is being challenged).

For a particular scheme, one can combine goals and attacks arbitrarily, which leads to six different possibilities. Bellare et al. identify the relations between these notions, both inside and outside the Random Oracle model. The most interesting conclusion is that a scheme satisfying the IND-CCA2 requirement will satisfy all others i.e. a scheme satisfying indistinguishability under an adaptive chosen ciphertext attack automatically satisfies all other combinations of goals/attacks.

There is another type of goal, called One Way Encryption (OWE) [18], in which the adversary has to gain advantage in a game where she is challenged

[1] We are not going to describe the symmetric counterparts of these notions, since they do not apply to pairing-based cryptography. A discussion of this can be found in [7].

[2] The notion that a cryptogram reveals nothing of the clear text, apart from its length.

with a specific ciphertext, and must present the corresponding cleartext. This notion is weaker than the previous ones, since an adversary winning this game will certainly be powerful enough to break the other types of attack.

These notions can also be considered in a multi-user setting [6].

Digital Signatures For digital signatures, the strongest and most widely used security model is that of existential unforgeability under an adaptive chosen message attack (UF-CMA) [21].

The concept of existential unforgeability means that it is infeasible for the adversary to present a valid signature for any message, under a particular set of parameters, namely a public key.

In an adaptive chosen message attack, the adversary is provided with a signing oracle, from which she may request an arbitrary number of signatures, for all messages except the one that she presents to win the game.

Identification Secure identification schemes are based on interactive (possibly zero knowledge) proofs of knowledge, whereby one party convinces another of her identity, by demonstrating knowledge about a secret.

The security of identification schemes is defined based on the concept of impersonation (IMP), for different types of attacks: passive attacks (PA), active attacks (AA) and concurrent attacks (CA) [8].

Impersonation, means that the prover (the party being identified) is able to make the verifier (the party performing the identification) accept a false identity claim with non-negligible probability.

In a passive attack (PA), the adversary is not allowed to interact with the system before attempting an impersonation. In an active attack (AA), the adversary is allowed to interact with the prover, several times, posing as verifier. Finally, in concurrent attacks (CA), the attacker is able to perform several active attacks in parallel, i.e. several sessions may occur simultaneously.

2.3 Elliptic Curves

Informally, an elliptic curve is the locus of points in the x-y plane that satisfy an algebraic equation of the form $y^2 = Ax^3 + Bx^2 + Cx^+D$. The values x and y may represent different things, namely elements of the sets $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Q}$, or a finite field $\mathbf{F}_p$.

These curves are not ellipses. Their name comes from a particular type of integral that arises from calculating the length of arcs in ellipses, called elliptic integrals. An elliptic integral is of the form

$$\int \frac{dx}{\sqrt{4x^3 - g_2x - g_3}} \tag{2.1}$$

Elliptic integrals are generalisations of inverse trigonometric functions. Over complex numbers, elliptic integrals are multivalued, and are only well-defined

modulo a period lattice i.e. there are two independent periods involved in the repetition pattern. This means that the values taken by elliptic integrals can be considered to be on a torus, just like the joint positions of the pointers in a clock.

This implies that the inverse function of an elliptic integral is a doubly periodic function. This type of function is called an elliptic function.

Elliptic functions, like every doubly-periodic function with periods that are independent over $\mathbb{R}$, satisfy an equation of the form

$$\wp'^2 = 4\wp^3 - g_2\wp - g_3 \tag{2.2}$$

for some constants g_2 and g_3.

If we consider the pair $(\wp', \wp)$ to be a point, then the solutions to Eq. 2.2 provide a map from a torus to the curve

$$Y^2 = 4X^3 - g_2X - g_3 \tag{2.3}$$

which is an example of an elliptic curve.

General elliptic curves are defined over a field K. Let $\bar{K}$ be its algebraic closure and K^* its multiplicative group. An elliptic curve over K is defined as the set of solutions in the projective plane $\mathbb{P}^2(\bar{K})$ of a homogeneous *Weierstrass equation* of the form

$$E : Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3 \tag{2.4}$$

with $a_1, a_2, a_3, a_4, a_6 \in K$.

In other words, we've got a cubic homogeneous (all terms with the same degree) equation with three variables. Solutions are seen as homogeneous co-ordinates in a projective plane over the algebraic closure of K.

A projective plane is an abstract concept in which points can be located using homogeneous co-ordinates. In terms of the solutions to the equation, a projective plane over $\bar{K}$ provides a locus for all possible solutions, including points at infinity. [3]

Considering a field $\hat{K}$ such that $K \subseteq \hat{K} \subseteq \bar{K}$, a point (X, Y, Z) (with $X, Y, Z \in \bar{K}$) on the curve is said to be $\hat{K}$-rational if the projection of the solution back to Euclidean co-ordinates gives a point in $\hat{K}$. For that we must have $(X, Y, Z) = \alpha(\hat{X}, \hat{Y}, \hat{Z})$ with $\hat{X}, \hat{Y}, \hat{Z} \in \hat{K}$.

Note that the curve has only one $\hat{K}$-rational point at infinity. By setting $Z = 0$ we get $X = 0$ meaning that all points in the Y axis map to the same $\hat{K}$-rational point, which is called the point at infinity $\mathcal{O}$.

[3] In order to capture the concept of infinity in the solution to polynomial equations, a "trick" is used. Instead of using (X, Y) co-ordinates in Euclidean space, which do not allow for infinity representation, a homogeneous co-ordinate system is used, which includes a third co-ordinate and looks like $(X/W, Y/W, W)$. Whenever $W = 0$ we've got infinity (actually it is a line in infinity where parallel planes meet). In order to obtain a one-to-one relationship to the original co-ordinate system, all points in the homogeneous co-ordinate system whose co-ordinates are proportional, are considered to be equivalent: points become lines.

When working with Euclidean co-ordinates, we use the affine form for the equation:

$$E : Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6 \tag{2.5}$$

This equation can be further simplified if the characteristic of K is not 2 or 3. In this case, it can be rewritten as:

$$E : Y^2 = X^3 + aX + b \tag{2.6}$$

From this point on, and for the sake of simplicity, we will stick to the use of affine coordinates, and to fields of characteristic greater than 3 (finite fields), so that we can use Equation 2.6.

For a particular curve, the *j-invariant* is defined as

$$j(E) = \frac{2^8 3^3 a^3}{4a^3 + 27b^2}$$

this quantity is preserved, even when the curve is subject to a change of variables.

A related quantity is the *curve descriminant*, which is defined as

$$\Delta(E) = -16(4a^3 + 27b^2)$$

When this in non-zero, then the left side of Equation 2.6 has three distinct roots, and the curve is non-singular.

If the field underlying an elliptic curve is algebraic closed, a straight line will intercept the curve at three points (counting tangents as multiple roots). If two of these points are known, then the third can be found. Furthermore, if a straight line intercepts an elliptic curve in two K-rational points, then the third point is also K-rational.

Let $P_1 = (x_1, y_1)$ and $P_2(x_2, y_2)$ be two points in a non-singular elliptic curve. Then, the third point in which the line through P_1 and P_2 intercepts the curve is given by $P_3 = (x_3, y_3)$, such that

$$\begin{aligned} x_3 &= \lambda^2 - x_2 - x_2 \\ y_3 &= \lambda(x_3 - x_1) + y_1 \\ \lambda &= \begin{cases} \frac{y_1 - y_2}{x_1 - x_2} & \text{for } x_1 \neq x_2 \\ \frac{y_1 - y_2}{x_1 - x_2} & \text{for } x_1 = x_2 \end{cases} \end{aligned} \tag{2.7}$$

The set of K-rational points of an elliptic curve, including the point at infinity $\mathcal{O}$ form an additive group, which we denote by $E(K)$. The addition operation is constructed based on the previous observations, by defining that

$$P_1 + P_2 + P_3 = \mathcal{O} \Leftrightarrow P_1 + P_2 = -P_3$$

Removing the minus from the previous equation, all of this can be written as:

$$\begin{aligned} &P_3 = (x_3, y_3) = (x_1, y_1) + (x_2, y_2) = P_1 + P_2 \\ &x_3 = \lambda^2 - x_2 - x_2 \\ &y_3 = \lambda(x_1 - x_3) - y_1 \\ &\lambda = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2} & \text{for } x_1 \neq x_2 \\ \frac{y_1 - y_2}{x_1 - x_2} & \text{for } x_1 = x_2 \end{cases} \end{aligned} \tag{2.8}$$

Finally, we have $-P = (x, -y)$ and $P + (-P) = \mathcal{O}$.

A natural extension of the addition operation is scalar multiplication. We define the *multiplication-by-m* $[m]P$ operation as

$$[m]P = \underbrace{P + P + \cdots + P}_{m} \text{ for } m > 0,$$

$$[0]P = \mathcal{O} \text{ and } [-m]P = [m](-P).$$

This operation is equivalent to exponentiation in multiplicative groups, and it is the basis for the problem underlying elliptic curve cryptography: the *discrete logarithm problem over elliptic curves* (ECDLP). This is the problem of, given a point $Q = [d]P$ and P, find the scalar d. As mentioned in Section 2.1, over carefully chosen elliptic curves, the ECDLP is a hard problem for which no sub-exponential time algorithm is known. This makes it possible to implement elliptic curve cryptosystems using smaller parameter sizes, for the same security level.

Based on the multiplication-by-m operation, we can also define the *order* of an elliptic curve point P as the smallest positive integer m such that $[m]P = \mathcal{O}$. If no such integer exists, then the order is said to be *infinite*.

This, in turn leads to the definition of the *subgroup of n-torsion points* in an elliptic curve. A point P for which $[n]P = \mathcal{O}$ is said to be an n-torsion point. Note that this happens only for points whose order divides n. The set of n-torsion points in an elliptic curve is given by

$$E(K)[n] = \{P \in E(K) \,:\, [n]P = \mathcal{O}\}$$

and it is a subgroup of $E(K)$.

Consider the case in which $K = \mathbb{F}_q$. The order of the group of points of the elliptic curve $E(\mathbb{F}_q)$, written $\#E(\mathbb{F}_q)$, satisfies the following equation

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

where t is called the *trace of Frobenius* (or simply the trace of the curve), satisfying the so-called *Hasse bound*:

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

An elliptic curve is said to be *supersingular* if the characteristic of the base field p divides the trace t.

The group structure of $E(\mathbb{F}_q)$ is rather complex. An interesting result concerns the sub-groups of n-torsion points. If n and q are relatively prime, then $E[n]$ is isomorphic to the $\mathbb{Z}_n \oplus \mathbb{Z}_n$. If $n = p^e$, where p is the characteristic of $\mathbb{F}_q$, then either $E[n] = \{\mathcal{O}\}$ if the curve is supersingular, or $E[n]$ is isomorphic to Z_{p^e} otherwise [44].

This result has two important consequences. Firstly, if n and q are relative prime then $|E[n]| = n^2$. Secondly, if n is prime, $E[n]$ is generated by any two linearly independent n-torsion points.

Finally, the *Frobenius Endomorphism* or Frobenius map in an elliptic curve $E(\mathbb{F}_q)$ is defined as

$$\begin{aligned} \Phi : E &\rightarrow E \\ \Phi(x, y) &= (x^q, y^q) \\ \Phi(\mathcal{O}) &= \mathcal{O} \end{aligned}$$

Note that if we have $\mathbb{F}_q \subseteq K$ then, for a point $P \in E(K)$, we have

$$\Phi(P) = P \Leftrightarrow P \in E(\mathbb{F}_q).$$

Chapter 3

Pairings

3.1 Introduction

In this chapter we describe how bilinear maps such as the ones defined in the previous chapter are implemented over elliptic curves. There are two constructions that can be used for this purpose: the Weil Pairing and the Tate Pairing. Their definitions are based on divisor theory [44, 32], so we will briefly describe the relevant aspects of this in the first part of this chapter.

3.2 Divisor Theory: The Very Basics

Take an elliptic curve $E(K)$ where $K = \mathbb{F}_q$ in Weierstrass form

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

The *coordinate ring* $K[E]$ of E over K is defined as the quotient ring

$$K[x, y]/I(E)$$

where $I(E)$ is the ideal formed by all polynomials with coefficients in K which are zero at all points of E.

Another way to put this is the following. Define the function $r \in K[x, y]$ as

$$r(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6.$$

The ideal $I(E)$ is now the ideal in $K[x, y]$ generated by r. By considering fractions of $K[E]$, we obtain the function field $K(E)$.

These definitions can also be extended to the algebraic closure of the field K, denoted by $\overline{K}$.

We call an element of $\overline{K}(E)$ a *rational function*. A rational function f is said to be *defined at a point* P if its denominator does not evaluate to 0 at point P. In this case, the evaluation of the f at point P is obtained by taking the

quotient between its numerator and its denominator when they are evaluated at the coordinates of P. If a function is not defined at a point P, then we say it has a *pole* in P. If the function is defined in P and its numerator evaluates to 0, then we say that the function has a *zero* in P. A rational function has a finite number of poles and zeros.

Evaluation of a rational function at infinity ($\mathcal{O}$) if obtained by comparing the degrees of the numerator and denominator, as usual. However, in this case, because of the relation between variables x and y established by the curve equation, their degrees are set at 2 and 3 respectively.

In order to define the concept of a divisor, one needs first to address the problem of pole and zero *multiplicity*. For each point in the elliptic curve $P \in E$ there exists a rational functions u such that $u(P) = 0$ and every rational function $f \in \overline{K}(E)$ can be written as $f = u^d s$. The function f is called a *uniformizing parameter at* P.

The interesting thing is that d does not depend on the actual function u and its value is called the *order* of f at P, denoted $d = \mathtt{ord}_P(f)$. For zeros, the order will be positive. For poles, the order will be negative. For all other points, the order will be zero. The *multiplicity of a pole or zero* P in f is the absolute value of the order of f at P.

The poles and zeros of a rational function completely define it up to a constant factor. To keep track of the poles and zeros of rational function, we use divisors. A *divisor* D is a formal sum of points of the form

$$D = \sum_{P \in E} n_P \langle P \rangle$$

with $n_P \in \mathbb{Z}$ and non-zero for only a finite number of points. Note that the sum is *not* associated with elliptic curve addition, and the n_P value is not elliptic curve scalar multiplication. The purpose of a divisor is not to be evaluated to a single point in the elliptic curve, but to associate "weights" to elements in lists of elliptic curve points. In some cases, divisors represent the orders of rational functions at relevant points. The *divisor of a function* f is defined as

$$\mathtt{div}(f) = \sum_{P \in E} \mathtt{ord}_P(f) \langle P \rangle$$

The set of all divisors generated by the points in a curve E is denoted $\mathtt{Div}(E)$. One can naturally define divisor addition over this set as

$$\sum_{P \in E} n_P \langle P \rangle + \sum_{P \in E} m_P \langle P \rangle = \sum_{P \in E} (n_P + m_P) \langle P \rangle$$

obtaining an abelian group.

Other useful notions associated with divisors are the following:

- The *support* of a divisor is the set of points for which n_P is not zero.
- The *degree* of a divisor is the sum of all n_P.

- The set of all divisors of degree 0 form a subgroup under divisor addition, denoted by $\texttt{Div}^0(E)$.
- A divisor D is called *principal* if $D = \texttt{div}(f)$ for some rational function f.
- Two divisors D_1 and D_2 are said to be *(linearly) equivalent*, denoted $D_1 \sim D_2$, if $D_1 - D_2$ is principal. Intuitively, D_1 is D_2 away from representing the divisor of some rational function f.
- A divisor D is principal if and only if it has degree 0 and the weighted sum (using n_P as scalar point multiplications) of the points in its support add up to $\mathcal{O}$.
- The set of all principal divisors in $\texttt{Div}(E)$ is denoted $\texttt{Prin}(E)$.
- The set of all divisors in $\texttt{Div}(E)$ which are not principal is represented by the *Picard group* or *divisor class group* defined as

$$\texttt{Pic}(E) = \texttt{Div}(E)/\texttt{Prin}(E).$$

- Every degree zero divisor is equivalent to $\langle P \rangle - \langle \mathcal{O} \rangle$ for some $P \in E$.
- The evaluation of a rational function f in a divisor D satisfying the restriction that the support of D and the support of the divisor of f share no common points, is defined as follows:

$$f(D) = \prod_{P \in D} f(P)^{n_P}$$

- Weil's reciprocity law states that $f(\texttt{div}(g)) = g(\texttt{div}(f))$.
- Constant factors do not affect the evaluation of a function at a divisor, i.e. if $f_2 = cf_2$, then $f_2(D) = f_1(D)$.

3.3 The Weil Pairing

There are two different, although related definitions of the Weil pairing. The first definition, as given in [44] serves a theoretical role in literature. The second definition, as given in [11] for example, allows for an efficient implementation and is therefore more suitable for pairing based cryptography. Here, we will focus on the latter. For a good description of both definitions and the relation between them refer to [32].

Let r be an integer co-prime to the field characteristic p, and let S and T be r-torsion points. Let also A and B be divisors such that $A \sim \langle S \rangle - \langle \mathcal{O} \rangle$ and $B \sim \langle T \rangle - \langle \mathcal{O} \rangle$, and A and B have disjoint support.

Note that because S and T are in $E[r]$, this means that rA and rB are principal divisors, so there are $f_A, f_B \in \overline{K}(E)$ such that

$$\texttt{div}(f_A) = rA \text{ and } \texttt{div}(f_B) = rB.$$

In this setting, the Weil pairing $\hat{e}_r : E[r] \times E[r] \to \mu_r$, where μ_r denotes the group of r-th roots of unity in $\overline{K}$, is given by

$$\hat{e}_r(S,T) = \frac{f_A(B)}{f_B(A)}.$$

The value of the Weil pairing is independent of the choice of divisors A and B, and functions f_A and f_B. The Weil pairing satisfies the following properties:

- Linearity in the first factor: $\hat{e}_r(S_1 + S_2, T) = \hat{e}_r(S_1, T)\hat{e}_r(S_2, T)$.
- Linearity in the second factor: $\hat{e}_r(S, T_1 + T_2) = \hat{e}_r(S, T_1)\hat{e}_r(S, T_2)$.
- Identity: $\hat{e}_r(S, S) = 1$.
- Alternation: $\hat{e}_r(S, T) = \hat{e}_r(T, S)$.
- Non-degeneracy: $\hat{e}_r(S, T) = 1, \forall S \in E[r] \Rightarrow T = \mathcal{O}$.

3.4 The Tate Pairing

Just as we can construct $K = \mathbb{F}_q$ as the quotient group $\mathbb{Z}/q\mathbb{Z}$, we can apply the same principle to $E(K)$, and construct $E(K)/rE(K)$. The elements of this quotient group are cosets of $E(K)$.

We call $C_\mathcal{O}$ the coset obtained by multiplying every point in $E(K)$ by an r such that $E[r] \neq \{\}$, because this coset is guaranteed to contain $\mathcal{O}$. All other cosets in the quotient group can be generated by adding a point that is not in $C_\mathcal{O}$ to all the points in this coset. The number of elements in $C_\mathcal{O}$ is $\#E(K)/r$ and, in this way, it is possible to split $E(K)$ into r distinct cosets of order r, which are the elements of $E(K)/rE(K)$.

If r is prime, then we have r^2 r-torsion points in $E(\overline{K})$. However, we may have either r or r^2 of these points in the curve over K. If $E[r] \nsubseteq E(K)$, then each co-set will have exactly one r-torsion point. If $E[r] \subseteq E(K)$, then each co-set will have exactly r^2 r-torsion point.

The Tate pairing definition also uses divisors and the evaluation of rational functions over devisors. Let $P \in E(F_{q^k})[r]$, where k is a suitable extension size called the *Tate embedding degree*. In practice, r is usually a large prime dividing the order of the curve, and k is the smallest integer such that $r|(q^k - 1)$. As before, the divisor $r\langle P\rangle - r\langle\mathcal{O}\rangle$ is principal, and we can find a rational function g for it.

Now let Q be a point representing a coset in $E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$, and D a degree zero divisor with support disjoint than that of $\texttt{div}(g)$, such that $D \sim \langle Q\rangle - \langle\mathcal{O}\rangle$.

In this setting, the Tate pairing is defined as

$$\hat{t}_r : E(F_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \rightarrow \mathbb{F}^*_{q^k}/(\mathbb{F}^*_{q^k})^r$$

$$\hat{t}_r(P, Q) = g(D).$$

Note that the codomain of the Tate pairing is also a quotient group where each element is a co-set. This means that a result produced by the Tate pairing, although constrained to a given coset, may not be unique. Each coset, must be seen as an equivalence class. The equivalence relation is given by

$$a \equiv b \Leftrightarrow a = bc^r \text{ , with } a,\ b,\ c \in \mathbb{F}_{q^k}.$$

In cryptographic applications it is usually the case that one needs pairing results to be exactly reproducible. For this reason, Tate pairing calculations are usually followed by an exponentiation to the power of $(q^k - 1)/r$ that removes the r-th power ambiguity and produces an r-th root of unity. An alternate definition for the Tate pairing is therefore

$$\hat{t}'_r : E(F_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \rightarrow \mu_r$$

$$\hat{t}'_r(P, Q) = g(D)^{(q^k-1)/r}.$$

Both definitions of the Tate pairing are well-defined and the result does not depend on the choice of g and D. The Tate pairing satisfies the following properties:

- Linearity in the first factor: $\hat{t}_r(P_1 + P_2, Q) = \hat{t}_r(P_1, Q)\hat{t}_r(P_2, Q)$.
- Linearity in the second factor: $\hat{t}_r(P, Q_1 + Q_2) = \hat{t}_r(P, Q_1)\hat{t}_r(P, Q_2)$.
- Identity: $\hat{t}_r(\mathcal{O}, Q) = 1$.
- Non-degeneracy:

$$\forall P \in E[r]\ ,\ \exists Q \in E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \text{ such that } \hat{t}_r(P, Q) \neq 1.$$

Note that, contrary to what occurs with the Weil pairing, $\hat{t}_r(P, P)$ may not be the identity. However, if $k > 1$, then this will always be the case [19]. Furthermore, if r is prime, $P \neq \mathcal{O}$ and Q is linearly independent from P, then the Tate pairing is non-trivial.

3.5 Embedding Degrees

The definitions of the Weil and Tate pairings over an elliptic curve $E(\mathbb{F}_q)$ imply extensions of the field F_q.

In the case of the Weil pairing, the domain requires two independent r-torsion points and the co-domain is the group of r-th roots of unity in the algebraic closure of F_q. So, the minimum extension size we need is the smallest

extension such that the curve $E(\mathbb{F}_{q^{k_w}}$ contains all the r-torsion points and the field extension $\mathbb{F}_{q^{k_w}}$ contains all r-th roots of unity. We call this minimum extension size the *Weil embedding degree* and denote it k_w.

For the Tate pairing, r is required to be co-prime to q, and the domain and co-domain require an extension of degree k, such that k is the smallest integer satisfying $m|(q^k - 1)$. We call this value of k the *Tate embedding degree*, and denote it k_t.

It is interesting that the definition of the Tate embedding degree is also a necessary and sufficient condition [32] for $\mu_r \subseteq \mathbb{F}_{q^{k_t}}$. The Weil embedding degree may be larger than this, as we need all the r-torsion points, but this is only possible happens for $k_t = 1$. For all other values of k_t we have $k_w = k_t = k$. For this reason, one usually just talks about the embedding degree of the curve.

3.6 Distortion Maps

The properties of both the Tate pairing and the Weil pairing imply that in order to have a non-trivial pairing value different from unity, one needs an efficient way of finding linearly independent r-torsion points. Supersingular curves provide an interesting feature in this context, which permits doing just that, as they permit the construction of *distortion maps* [46].

Let P be an r-torsion point in the elliptic curve $E(\mathbb{F}_q)$ with r co-prime to q. A distortion map with respect to P is an endomorphism ϕ that maps P to a point $\phi(P)$ which is linearly independent from P. Note that, since we are dealing with an endomorphism, the point at infinity does not admit a distortion map. Also, through the endomorphism structure we know that the value of $\phi(P)$ is still an r-torsion point.

Distortion maps do not exist for non-supersingular elliptic curves. However, when implementing pairing based protocols using supersingular curves, distortion maps are quite handy. In particular, they allow for the construction of *modified pairings* (or symmetric pairings) which take both parameters from the same group G_1, simply by using

$$\hat{e}(P, \phi(Q))$$

where both P and Q are in the same group G_1.

Chapter 4

Efficient implementation

4.1 Background

The fact that the Weil pairing is usually assumed to require a workload equivalent to that of two Tate pairing calculations, has relegated the former to a secondary role in the implementation of pairing-based cryptosystems. Nevertheless, as will be pointed out in Section 6 Koblitz and Menezes have recently suggested that the Weil pairing is preferable for higher security levels, since it does not require the final exponentiation to obtain a unique result.

Efficiency in pairing-based cryptosystems comes down to a trade-off between three parameters: security level, computational load and bandwidth. These parameters can be directly linked to the sizes of the groups over which the pairings are calculated, as well as the sizes of the base field and its extension.

The following conditions must be met:

- For the DLP and CDH problems to be hard in $E(\mathbb{F}_q)$, we must have $log(q) \geq 160$ [4]. Note that we have $q = p^g$, where p is the characteristic of the field.
- For the DLP and DH problems to be hard in $\mathbb{F}_{q^k}$, we must have $log(q) * k \geq 1024$ [4].
- For the BDHP to be hard, the size r of the groups over which the pairing is operating is typically a large prime divisor of the order of the elliptic curve, or the order itself. This must be a prime of at least 160 bits [20], and it is usually expected to be of the same magnitude as q [37].

Nevertheless there is some flexibility in choosing adequate values for q and k. In general, a small value for k implies a larger base field, which will bring added costs, not only to the pairing calculations, but also to the elliptic curve implementation. On the other hand, larger values of k imply additional complexity in dealing with the field extensions. Finally, bandwidth issues are very

dependent on the characteristics of the target cryptosystem. However, it is usually the case that because larger values of k allow for smaller elliptic curve point representations, this will also bring overall bandwidth gains.

4.2 To go or not to go supersingular

Supersingular elliptic curves have very rich internal structure, and are very easy to construct. This has made them an obvious choice for the implementation of elliptic curve cryptosystems. However, their particular characteristics made some authors suspicious that this type of curve might be an easier target for attacks. This was confirmed when the MOV attack was published in 1993 [34] and later, in 1994, when when the F-R attack [17] appeared.

These attacks use pairings to reduce the DLP over an elliptic curve, to the DLP over an extension of the base field, for which sub-exponential time algorithms are known. The attack is only viable on particular types of curves, namely those over which the size of the required extension field is small. However, for supersingular curves this is always the case, which means that for acceptable levels of security, supersingular curves require larger base field sizes.

When pairing-based cryptosystems appeared, supersingular curves were once again considered for the implementation of cryptographic algorithms. The fact that they are easy to construct and that distortion maps exist only for supersingular curves makes them ideal for this type of application.

The above mentioned distrust for supersingular curves, and the need for higher values of the embedding degree has led researchers to look at the problem of generating suitable non-supersingular curves for pairing based cryptosystems. MNT curves [36] [42] are elliptic curves of prime order, which are built using the Complex Multiplication (CM) method, and that can be generated with specific embedding degree values.

These curves have the advantage that the prime order subgroup that is used for pairing calculations is the curve itself, thus minimizing the associated co-factor. They have, however, the disadvantage that it is not possible to obtain modified versions of the pairings, where non-degeneracy is guaranteed, and where both pairing parameters come from the same group. This, in turn, makes it less straightforward to implement pairing-based cryptosystems over MNT curves [37].

4.3 Base field and field extensions

The size of the extension field over which the pairing is calculated is central to the efficiency of the pairing calculation. Recall from the previous chapter that, for the Tate pairing, the required extension field must contain all the l^{th} roots of unity. A necessary and sufficient condition for this is that k be the smallest integer such that $l|(q^k - 1)$ [1]. In the case of the Weil pairing, the condition that must be met by the extension field is that it contains all elements of order l.

The required extension size is generally greater or equal than that required for the Tate pairing. However, the results in [1] indicate that the extension fields required for both pairings are usually the same [19], and this is always true if $k > 1$.

The implementation of elliptic curve cryptosystems is usually based on two types of base fields: fields of characteristic 2, i.e. extensions of $\mathbb{F}_{2^n}$; or fields of large prime characteristic, i.e. $\mathbb{F}_p$. The most efficient implementations operate over binary fields, and take advantage of well-known optimisations that are particularly suited for hardware implementation. On the other hand, the efficient implementation of arithmetic modulo large primes has been extensively studied for the implementation of cryptosystems not based on elliptic curves. It was natural to take advantage of these developments in the implementation of elliptic curve cryptosystems.

Extensions of $\mathbb{F}_3$ had not been considered interesting enough until pairing-based cryptography came into the picture. This new interest is to the fact that, for supersingular curves over $\mathbb{F}_{2^n}$, it is not possible to obtain values of $k > 4$. However, if one considers supersingular curves over $\mathbb{F}_{3^n}$, it is possible to maximize the value of k, obtaining an optimum value of 6 [1].

This has prompted several authors to investigate the adaptation of optimisations used over other types of fields, to the particular case of fields of characteristic three [4] [37]. Examples of this are efficient point tripling algorithms, triple-and-add adaptations for efficient scalar multiplication, square root extraction, etc.

Galbraith et al. [20] propose additional optimisation techniques, particularly in what concerns the implementation of extension fields. These authors identify as most relevant cases the implementation of extension fields $\mathbb{F}_{2^{4m}}$ and $\mathbb{F}_{3^{6m}}$, as these are the cases where the embedding degree is maximised. They propose the following constructions:

- $F = \mathbb{F}_{2^m}$ is extended using a tower of two quadratic extensions: $F_1 = F[x]/(x^2 + x + 1) \cong \mathbb{F}_{2^{2m}}$ and $F_2 = F_1[y]/(y^2 + (x + 1)y + 1) \cong \mathbb{F}_{2^{4m}}$.
- $F = \mathbb{F}_{3^m}$ is extended using a tower of a cubic extension and a quadratic extension: $F_1 = F[a]/(a^3 - a + 1) \cong \mathbb{F}_{3^{3m}}$ and $F_2 = F_1[b]/(b^2 + 1) \cong \mathbb{F}_{3^{6m}}$.

In both cases, the trivial multiplication procedure is modified so as to minimise the number of multiplications over the underlying field. Division computations are optimised by using conjugates.

The work by Barreto et al. [2] in 2003 introduced a new approach to the choice of groups forpairing calculation over non-supersingular curves. The authors propose a mechanism for generating the point $Q \in \mathbb{F}_{q^k}$ through a construction that uses the twist of the curve over $\mathbb{F}_{q^d}$ where k is even, and $d = k/2$.

The method consists in working over the twist of the curve in $\mathbb{F}_{q^d}$ to obtain a random point Q'. For a curve E given by $y^2 = x^3 + ax + b$, the twist over $\mathbb{F}_{q^d}$ is given by $E'(\mathbb{F}_{q^d}) : y^2 = x^3 + v^2ax + v^3b$, for some quadratic non-residue

[1] For supersingular curves, $k \leq 6$. [34]

$v \in \mathbb{F}_{q^d}$. Since $d = k/2$, v will be a quadratic residue in $\mathbb{F}_{q^k}$, which means that the map $\Psi : (X, Y) \rightarrow (v^{-1}X, (v\sqrt{v})^{-1}Y)$ is an isomorphism that maps the group of points of $E'(\mathbb{F}_{q^d})$ to a subgroup of points of $E(\mathbb{F}_{q^k})$. The point Q is generated as $Q = \Psi(Q')$.

The fact that the x-coordinate of Q is in $E'(\mathbb{F}_{q^d})$ by construction, means that the optimisations of Miller's algorithm described in the next section, and taken from [4], can be applied in this setting as well. Additionally, bandwidth and processing can be saved by only explicitly calculating Q when the pairing is evaluated. All other operations such as key generation, hashing and point transmission can be carried out using only $\mathbb{F}_{q^d}$ arithmetic.

4.4 Miller's algorithm

All pairing implementations derive from an algorithm invented by Miller [35] in 1985. This algorithm is basically a double-and-add procedure applied to the construction of the rational functions required by the Weil and Tate pairings.

Recall that both the Weil and the Tate pairing definitions require the evaluation of $f(D)$, where f is a rational function with divisor of the form $\mathtt{div}(f) = r\langle P\rangle - r\langle\mathcal{O}\rangle$, and D is a divisor with support disjoint than that of $\mathtt{div}(f)$.

Let f_c be a rational function such that $\mathtt{div}(f_c) = c\langle P\rangle - \langle[c]P\rangle - (c-1)\langle\mathcal{O}\rangle$. Miller's algorithm is based on the observation that the desired function f can be constructed using the following recurrence

$$f_{a+b} = f_a \cdot f_b \cdot h_{[a]P,[b]P}/h_{[a+b]P,-[a+b]P}$$

where $h_{U,V}$ is the rational function given by the line through U and V. As always, if $U = V$ the line $h_{U,U}$ is the tangent in this point, and if either $U = \mathcal{O}$ or $V = \mathcal{O}$, then $h_{U,V}$ is the vertical line passing through the other point.

In the simpler case of the Tate pairing, where the aim is to construct a function g with divisor $r\langle P\rangle - r\langle\mathcal{O}\rangle$, and P is an r torsion point, f_r is exactly what we need. [2] Using the previous recursion directly, one would need to maintain representations of rational functions while constructing the required function g. However, given that we only need the evaluation of this function at the divisor D, this is not necessary. In fact, in Miller's algorithm, the evaluation is accumulated in each step of the recurrence, allowing for a very efficient implementation. Figure 4.4 shows Miller's algorithm for the Tate pairing evaluation.

4.5 Optimisation of Miller's algorithm

In [4], Barreto et al. propose several optimisations to Miller's algorithm for the Tate pairing evaluation.

The first of these optimisations are based on the observation that for the *interesting* supersingular elliptic curves in Table 4.5, $(q-1)$ is a factor of $(q^k -$

[2] In the Weil pairing, several rational functions with this structure must be combined to obtain the required f_A and f_B.

Figure 4.1: Miller's algorithm for the Tate pairing

$f \leftarrow 1$
$V \leftarrow P$
for $i \leftarrow t-1, t-2, ..., 1, 0$ do
  $f \leftarrow f^2 \cdot g_{V,V}(D)/g_{2V,-2V}(D)$
  $V \leftarrow 2V$
  if $r_i = 1$
    $f \leftarrow f \cdot g_{V,P}(D)/g_{V+P,-(V+P)}(D)$
    $V \leftarrow V + P$
  end if
end for
return f

Table 4.1: Barreto et al.'s interesting supersingular curves

Curve	**Base field**	**Curve order**	**k**
$y^2 = x^3 + (1-b)x + b, b \in \{0,1\}$	$\mathbb{F}_p$	p+1	2
$y^2 + y = x^3 + x + b, b \in \{0,1\}$	$\mathbb{F}_{2^m}$	$2^m + 1 \pm 2^{(m+1)/2}$	4
$y^2 = x^3 - x + b, b \in \{-1,1\}$	$\mathbb{F}_{3^m}$	$3^m + 1 \pm 3^{(m+1)/2}$	6

$1)/r$, the exponent used to convert the result of the Tate pairing into an actual root of unity. This leads to a theorem stating that in the calculation of the Tate pairing for these curves, one can simply evaluate $g(Q)$, followed by the usual exponentiation. In [2], this optimisation is extended to the friendly groups described in Section 4.3.

The second optimization applies to a subset of the curves in Table 4.5, and to the modified version of the Tate pairing using distortion maps. Under these special circumstances, Barreto et al. demonstrate that the denominators of the rational functions used in Miller's double-and-add algorithm can be ignored, since they eventually evaluate to 1. In [2], this optimisation is extended to the friendly groups described in Section 4.3, in the case where k is even.

Finally, Barreto et al. propose the use of a Solinas prime for r, which minimises the number of non-doubling operations.

Galbraith et al. [20] also endorse this recommendation.. Alternatively, these authors point out that the Tate pairing may be calculated, not in respect to r, but in respect to the order of the curve, which may be advantageous if the prime r does not have a low hamming weight. Note that, in this case, the pairing values will be different, and so all users of a cryptosystem must follow the same convention in this respect. Galbraith et al. [20] also provide a version of Miller's algorithm optimised for fields of characteristic three, and propose the selection of the random point[3] S, not from the curve over extended field, but over the base field. This allows carrying out part of the calculations directly over the base field, thereby speeding up the process.

[3]The point used to build a divisor equivalent to $(Q) - (O)$

Other observations by these authors include the fact that all the line coefficients are over the base field (if P's coordinates are in the base field), and that the calculation of the rational function may be performed so as to postpone all divisions until the very end, at which point a single division provides the final result.

Izu and Takagi [26] further elaborate on how to build on top of the previous improvements, by proposing optimisations for implementations over generic curves i.e. not necessarily supersingular. These optimisations are based mostly on clever manipulations of coordinate systems, and careful re-utilisation of precomputed quantities.

Scott and Barreto [41] propose an optimization that applies to the final exponentiation in Tate pairing calculations when k is even. The exponent, $(q^2-1)/r = (q+1)(q-1)/r$ can be partitioned into two factors $(q-1)$ and $(q+1)/r$.

Suppose that $\mathbb{F}_{q^k} = \mathbb{F}_{k^{2d}}$ is implemented as a quadratic extension of $\mathbb{F}_{q^d}$, then we can represent an element of $\mathbb{F}_{q^k}$ as $x+iy$, where $\delta = i^2 \in \mathbb{F}_{q^d}$ is some quadratic non residue.

Scott and Barreto note that part of the final exponentiation can be efficiently calculated as

$$(x+iy)^{q-1} = (x+iy)^q/(x+iy) = (x-iy)/(x+iy).$$

Furthermore, a value $a+ib = (x+iy)^{q-1}$ calculated in this way has a very interesting property. These values are *unitary*, i.e. their norm, $a^2-\delta b^2$, is equal to 1, and this means that exponentiations can be calculated very efficiently using Lucas sequences.

These authors also propose a method for compressing pairing values that leads to a bandwidth gain of roughly 50% when transferring pairing values. This method operates roughly as elliptic curve point compression whereby only one of the coordinates is transmitted and the other one is only defined up to a sign.

In [16], Duursma and Lee generalise the work by Barreto et al. and by Galbraith et al. to the hyperelliptic case, providing further optimisation for the characteristic 3 case. Their algorithm is currently believed to be the most efficient.

Restricting their work to the elliptic curve case, Duursma and Lee's improvements apply to the following family of supersingular curves over fields of characteristic three: $E(\mathbb{F}_q) : y^2 = x^3 - x + b, b \in -1, 1, q = 3^n$. The orders of the groups in these curves N are well known, and have a very small ternary expansion of the form $3^{2m-1} \pm 3^m + 1, 2m-1 = n$. For $l = N$, the (Tate and Weil) embedding degree is $k = 6$.

All the optimizations proposed by Barreto et al. [4] also apply to these curves. The field extension is constructed using a cubic and a quadratic extension: $\mathbb{F}_{q^3} = \mathbb{F}_q[\rho]/(\rho^3 - \rho - b)$ and $\mathbb{F}_{q^6} = \mathbb{F}_{q^3}[\sigma]/(\sigma^2+1)$. The distortion map is given by: $\phi(Q) = \phi(x,y) = (\rho - x, \sigma y)$. Note that the x-coordinate is in $\mathbb{F}_{q^3}$.

The real contribution of Duursma and Lee [16] is that they found a closed

Figure 4.2: The Duursma-Lee algorithm for the Tate pairing

Input: point $P = (x_1, y_1)$, point $Q = (x_2, y_2)$
Output: $f_P(\phi(Q)) \in \mathbb{F}^*_{q^6}/(\mathbb{F}^*_{q^6})^{3^{3n}+1}$

$f \leftarrow 1$
for $i = 1$ **to** m
    $x_1 = x_1^3, y_1 = y_1^3$
    $\mu \leftarrow x_1 + x_2 + b, \lambda \leftarrow -y_1 y_2 \sigma - \mu^2$
    $g \leftarrow \lambda - \mu\rho - \rho^3, f \leftarrow f * g$
    $x_2 \leftarrow x_2^{1/3}, y_2 \leftarrow y_2^{1/3}$

form for the Tate pairing, which permits breaking away from Miller's algorithm standard implementation.

The main observation in their work is that setting $l = 3^{3n} + 1$, we have $e_l(P,Q)^{3^{3n}-1} = e_N(P,Q)^{(3^{6n-1}-1)/N}$. This turns l into a trivial ternary expansion, which elliminates additions in Miller's algorithm, but it makes the final exponentiation a bit awckward. Duursma and Lee went on to show that the increase in l's bit length can be compensated by processing three trits at a time, and providing a closed form for the Tate pairing, which greatly simplifies implementations.

The algorithm is described in Figure 4.5, as found in [22]. Note that two cubings and two cubic roots are required in each iteration.

Granger et al. [23] tackle the problem of finding the unique representative of the quotient group avoiding the final exponentiation by $3^{3n} - 1$, which requires a division. What is achieved is a compression technique, whereby the representative of the quotient group is stored using only a value in $\mathbb{F}_{q^3}$. This method has the advantage that subsequent operations over the pairing value, computed over this representation, provide a significant efficiency gain.

In [22] the same authors investigate optimal implementations of the cubing and cubic root operations over fields of characteristic three required by the algorithm in Figure 4.5 by using normal bases.

Chapter 5

Identity-based Cryptography

5.1 Background

Identity based cryptography was proposed by Shamir [43] in 1985. The problem that this author addressed was the need for multiple interactions between users using public key cryptography.

Typical use of digital signatures and asymmetric ciphers implies the exchange of public key certificates between the parties, since that is the only way random public keys can be trustfully associated with one's identity. Shamir's idea consists of using a readable representation of the identity as the public key, thereby eliminating the need for public key certificates.

For this to be possible, Shamir came up with a three party paradigm. It involves the two parties (A and B) who are communicating, and a *Key Generation Center* (KGC), who is responsible for providing A and B with their private keys. The KGC derives the private keys from the readable public keys, and must be the only party capable of doing so.

This type of system heavily relies on the trustworthiness of the KGC, since it knows everyone's private key, and on the viability of transferring private keys to users using a secure channel (Shamir advocates smartcards).

Note that, under these assumptions, A and B will be able to communicate securely using a single exchange, as soon as they are provided with their own private keys.

The requirements for this type of scheme, as stated by Shamir, are the following:

- In addition to the properties expected for any public key system,
- the KGC must be able to generate private keys efficiently from identities, using a given *master key*, which it keeps secret;

- not knowing this master key, it must be infeasible for any party to obtain it, given an arbitrary number of key pairs and instances of the identity based algorithm execution; and,
- not knowing this master key, it must also be infeasible to recover an agent's private key.

Is his original work, Shamir proposed a signature scheme that satisfies these requirements (see Section 5.3). He was, however, unable to present a suitable encryption scheme, and this was an open problem for a few years (see Section 5.4).

5.2 Security Models

The security models defined in Section 2.2 are not applicable in an identity based scenario, since the interactive key generation process may leak information about the challenge.

Boneh and Franklin [11] presented an adapted version of the security model for encryption in the identity based scenario. The goals of indistinguishability and non-malleability have exactly the same meaning, but the attack models must be refined:

- The attacker is given access to a private key generation oracle from which she may obtain private keys for public keys of her choosing.
- The public key on which the adversary is challenged is of her choosing (and obviously she may not ask the oracle for the corresponding private key).

The security models for other identity based cryptographic primitives are derived from their non-identity based counterparts using similar adaptations.

5.3 Signatures

The concept of an identity based signature, as a dual for identity based encryption must be taken with care. This is because the purpose of a digital signature is authentication, and the escrow functionality inherent to identity-based systems is somewhat unnatural in this context.

Basically, it is difficult to argue that a system in which the signer must contact a Key Generation Center (KGC) to obtain is private signing key is any use at all in terms of authentication. At any time the KGC can generate valid signatures on behalf of the signer without any possibility of this being detected. This is somewhat different for identity based encryption, in which the private key must only be derived when a decryption is going to take place, and where key escrow functionality can actually be included as a desirable feature [25]. One way to solve this is, of course, to resort to multiple key generation authorities such that forgery is infeasible up to a threshold of colluding authorities.

Public Parameters	(n, e, f)	A RSA modulus n, a large number e relative prime to $\phi(n)$, and a one way function f taking two arguments.
KGC Master Key	(p, q)	The prime factorisation of $n = p \cdot q$.
Priv. Key Extraction	$g = \sqrt[e]{i} \pmod n$	The private key is the number g such that $i = g^e \pmod n$, where i is the identity of the private key owner. The security of the scheme relies on the assumption that this calculation is only feasible knowing the factorisation of n, which directly relates it to the security of RSA.
Signature	(s, t)	First, one calculates $t = r^e \pmod n$, where r is a randomly generated number, and then $s = g \cdot r^{f(t,m)} \pmod n$. Again, the security assumption is related to that of RSA e.g. selecting t at random and calculating a suitable s implies the factorisation of n.
Verification	$s^e = i \cdot t^{f(m,t)} \pmod n$	Because e is relative prime to n one can cancel it in the exponents on both sides of the equation.

Table 5.1: Shamir's original identity based signature scheme

Public Parameters	$(G_1, G_T, \hat{e}, H_1, H_2, P, P_{Pub})$	Two cyclic groups of prime order q and a bilinear map between them $\hat{e} : G_1 \times G_1 \rightarrow G_T$, such that the one-more CDH problem is hard in G_1. Two hash functions H_1 and H_2 that take identity representations and $\{0,1\}^* \times G_1 \times \{0,1\}^*$ tuples into G_1, respectively. A random generator P of G_1, and the KGC's public key $P_{Pub} \in G_1$.
KGC Master Key	s	The discrete logarithm of P_{Pub} to the base P, or $P_{Pub} = sP$.
Priv. Key Extraction	d_{ID}	Given a finite bit string representing the ID, the private key is $d_{ID} = sQ_{ID} = sH_1(ID)$.
Signature	(Y, Z)	First, one calculates $Y = yP$, where y is a randomly generated number, and then $Z = yH_2(ID, Y, M) + d_{ID}$.
Verification	$\hat{e}(Z, P) = \hat{e}(H_2(ID, Y, M), Y) \cdot \hat{e}(H_1(ID), P_{Pub})$	From the pairing's bilinearity.

Table 5.2: Sakai et al.'s pairing based and identity based signature scheme

The first identity based signature scheme was included in Shamir's first work on identity based cryptography [43], in 1985. It is a close relative of RSA, but it was proposed informally, as an example, and no formal security claims were stated for it. A description is included in Table 5.1 for reference purposes.

After Boneh and Franklin's identity based encryption came out in 2001 [11], several authors proposed signature schemes, operating on the same parameter set, in order to provide a complete identity based PKI. Here we refer a few: SOK-IBS by Sakai et al [40], CC-IBS by Cha and Cheon [14], the scheme by Hess [25] and the scheme by Paterson [38].

In this report we will present the first three schemes as taken from Bellare et al.'s [8] work on the provable security of identity based signature and identification schemes. This is particular important for the SOK-IBS scheme, since it is modified to obtain improved security properties.

Table 5.2 shows the details of the SOK-IBS algorithm. In their original

Public Parameters	$(G_1, G_T, \hat{e}, H_1, H_2, P, P_{Pub})$	Two cyclic groups of prime order q and a bilinear map between them $\hat{e} : G_1 \times G_1 \rightarrow G_T$, such that the one-more CDH problem is hard in G_1. Two hash functions H_1 and H_2 that take identity representations into G_1 and $\{0,1\}^* \times G_2 \times \{0,1\}^*$ tuples into $\mathbb{Z}_q$, respectively. A random generator P of G_1, and the KGC's public key $P_{Pub} \in G_1$.
KGC Master Key	s	The discrete logarithm of P_{Pub} to the base P, or $P_{Pub} = sP$.
Priv. Key Extraction	d_{ID}	Given a finite bit string representing the ID, the private key is $d_{ID} = sQ_{ID} = sH_1(ID)$.
Signature	(α, Z)	First, one calculates $\alpha = \hat{e}(P,P)^y$, where y is a randomly generated number, and then $Z = d_{ID} H_2(ID, \alpha, M) + yP.$
Verification	$\hat{e}(Z,P) = \alpha \cdot \hat{e}(H_1(ID), P_{Pub})^{H_2(ID,Y,M)}$	From the pairing's bilinearity.

Table 5.3: Hess's pairing based and identity based signature scheme

work, Sakai et al. proposed this algorithm in a slightly different form: instead of using $H_2(Y, M)$ they simply used $H_2(M)$. By introducing this change, Bellare et al. were able to demonstrate that this scheme is UF-CMA secure under the one-more CDH assumption in the RO model.

The one-more CDH assumption states that it is infeasible for an adversary to gain advantage in a game in which it has access to a CDH oracle and a challenge oracle. The goal of the game is to solve the CDH problem for at least one more challenge than the queries made to the CDH oracle.

Libert and Quisquater [31] recently proved that, although it is less efficient, this scheme provides better security guarantees than its counterparts for the same security levels, since it allows for more efficient reductions to the underlying security problem. These authors have also pointed out that this algorithm is an identity based adaptation of a randomized version of Boneh et al.'s short signatures described below.

Table 5.3 shows the identity based signature algorithm proposed by Hess [25]. This scheme is also UF-CMA secure under the one-more CDH assumption in the RO model [8].

Table 5.4 shows the identity based signature algorithm proposed by Cha et al. [14]. This scheme is also UF-CMA secure under the one-more CDH assumption in the RO model [8].

Table 5.5 shows the identity based signature algorithm proposed by Paterson [38]. The security of this scheme is left in [8] as an open problem.

Finally, one of the most cited pairing-based algorithms is a short signature algorithm by Boneh et al. (BLS) [12]. It is not identity-based, but it is included here for reference purposes.

Public Parameters	$(G_1, G_T, \hat{e}, H_1, H_2, P, P_{Pub})$	Two cyclic groups of prime order q and a bilinear map between them $\hat{e} : G_1 \times G_1 \rightarrow G_T$, such that the one-more CDH problem is hard in G_1. Two hash functions H_1 and H_2 that take identity representations into G_1 and $\{0,1\}^* \times G_1 \times \{0,1\}^*$ tuples into $\mathbb{Z}_q$, respectively. A random generator P of G_1, and the KGC's public key $P_{Pub} \in G_1$.
KGC Master Key	s	The discrete logarithm of P_{Pub} to the base P, or $P_{Pub} = sP$.
Priv. Key Extraction	d_{ID}	Given a finite bit string representing the ID, the private key is $d_{ID} = sQ_{ID} = sH_1(ID)$.
Signature	(Y, Z)	First, one calculates $Y = yP_{Pub}$, where y is a randomly generated number, and then $Z = d_{ID}(H_2(ID, Y, M) + y)$.
Verification	$\hat{e}(Z, P) = \hat{e}(H_1(ID), Y) \cdot \hat{e}(H_1(ID), P_{Pub})^{H_2(ID,Y,M)}$	From the pairing's bilinearity.

Table 5.4: Cha et al.'s pairing based and identity based signature scheme

Public Parameters	$(G_1, G_T, \hat{e}, H_1, H_2, P, P_{Pub})$	Two cyclic groups of prime order q and a bilinear map between them $\hat{e} : G_1 \times G_1 \rightarrow G_T$, such that the CDH problem is hard. Three hash functions H_1, H_2 and H_3 that take identity representations into G_1, messages into $\mathbb{Z}_q$, and elements of G_1 to $\mathbb{Z}_q$, respectively. A random generator P of G_1, and the KGC's public key $P_{Pub} \in G_1$.
KGC Master Key	s	The discrete logarithm of P_{Pub} to the base P, or $P_{Pub} = sP$.
Priv. Key Extraction	d_{ID}	Given a finite bit string representing the ID, the private key is $d_{ID} = sQ_{ID} = sH_1(ID)$.
Signature	(Y, Z)	First, one calculates $Y = yP$, where y is a randomly generated number, and then $Z = y^{-1}(d_{ID}H_3(Y) + (H_2(M)P)$.
Verification	$\hat{e}(Z, Y) = \hat{e}(H_1(ID), P_{Pub})^{H_3(Y)} \cdot \hat{e}(P, P)^{H_2(M)}$	From the pairing's bilinearity.

Table 5.5: Paterson's pairing based and identity based signature scheme

Public Parameters	$(G_1, G_2, G_T, \hat{e}, H, P, P_{Pub})$	Three cyclic groups of prime order q and a bilinear map between them $\hat{e} : G_1 \times G_2 \rightarrow G_T$, such that (G_1, G_2) is a gap-co-Diffie-Hellman pair. A hash function H that takes n-bit-long messages to G_1. A random generator P of G_2, and the public key $P_{Pub} \in G_2$.
Private Key	x	The private key is the number $x \in \mathbb{Z}_q$ such that $P_{Pub} = xP$
Signature	σ	The signature for a message M is calculated as $\sigma = x \cdot H(M)$.
Verification	$\hat{e}(\sigma, P) = \hat{e}(H(M), P_{Pub})$	The verification procedure is trivial by the pairing's bilinearity.

Table 5.6: Boneh et al.'s short signature algorithm from pairings

The BLS algorithm is very simple. It is described as being based on a co-Gap-Bilinear-Diffie-Hellman group, i.e. using asymmetric pairings, and we present it in Table 5.6. The scheme is shown to be existential unforgeability secure under an adaptive chosen message attack (UF-CMA), in the RO model.

The short-signature characteristic arises from the implementation that is proposed by the authors. Note that the signature is a single element of G_1. The smallest the representation that can be found for such an element, the shorter the resulting signatures.

Boneh et al. propose an implementation based on the Weil or the Tate pairing over MNT elliptic curves, using embedding degree six [1].

Given that a signature is simply a point of the curve over the base field, Boneh et al. relax the verification procedure in order to be able to use only the x-coordinate of this point as signature. For this to be possible, the verification process must accept signatures generated by σ or $-\sigma$. By the bilinearity property of the pairing, this just means that you must test for the inverse as well. Using this observation, this algorithm produces signatures of 160 bits, at the same security level as an El Gamal signature of 320 bits, and an RSA signature of 1024 bits.

5.4 Encryption

In 2001, Boneh and Franklin [11] published a paper describing an identity based encryption scheme based on the Weil Pairing. This was ground breaking, since it had been a few years since Shamir [43] had identified the problem, and left it open. The system was also ground breaking because it confirmed the potential of pairings to be used for "good" in the construction of cryptographic schemes, and not just as a cryptanalysis tool.

The identity based encryption scheme in [11] works over any bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_T$ between two groups G_1 and G_T.

[1] Supersingular curves of characteristic three are also recognised to be a solution, but said not to be an optimal solution for bandwith saving since Coppersmith's DLP method means longer bit-lengths are required for the field extensions.

Public Parameters	$(G_1, G_T, \hat{e}, H_1, H_2, P, P_{Pub})$	Two cyclic groups of prime order q and a bilinear map between them $\hat{e} : G_1 \times G_1 \rightarrow G_T$, such that the BDH problem is hard. Two hash functions H_1 and H_2 that take identity representations to G_1, and elements of G_T into the n-bit-long message space, respectively. A random generator P of G_1, and the KGC's public key $P_{Pub} \in G_1$.
KGC Master Key	s	The discrete logarithm of P_{Pub} to the base P, or $P_{Pub} = sP$.
Priv. Key Extraction	d_{ID}	Given a finite bit string representing the ID, the private key is $d_{ID} = sQ_{ID} = sH_1(ID)$.
Encryption	(y_1, y_2)	A pair,in which the first element $y1 = rP$ hides a random $r \in \mathbb{Z}_q^*$, and the second element $y_2 = M \oplus H_2(g_{ID}^r))$ masks the plain text with a hashed version of $g_{ID} = \hat{e}(Q_{ID}, P_{Pub})$, to the power r.
Decryption	$M = y_2 \oplus H_2(g_{ID}^r)$	Where the mask is reconstructed from $g_{ID}^r = \hat{e}(y_1, d_{ID})$

Table 5.7: Boneh and Franklin's identity based encryption scheme

The proofs of security presented by the authors for the following schemes are given in the random oracle model, under the assumption that the BDHP is hard.

Table 5.7 shows a simplified version of the algorithm, which is shown to be OWE secure.

The complete identity based encryption algorithm described in [11] corresponds to a transformed version of the one-way secure encryption scheme. The complete version provides IND-ID-CCA2 security, and the transformation is the one in [18].

The concrete imlementation proposed by Boneh and Franklin goes as follows:

- The elliptical curve used is a $y^2 = x^3 + 1$ over $\mathbb{F}_p$, with $p = 2 \pmod 3$, which is supersingular with embedding degree $k = 2$. The number of points in the curve is $\#E(\mathbb{F}_p) = p + 1$.
- The group of points of order $q = (p + 1)/l$ in $E/\mathbb{F}_p$ is used as group G_1, where q is a large prime factor of the curve order but q^2 does not divide it[2].
- A non-trivial root of $x^2 - 1$, $\zeta \in \mathbb{F}_{p^2}$ is used to build a distortion map $\phi(x, y) = (\zeta x, y)$, that for each point in G_1 permits finding a linear independent point in $E/\mathbb{F}_{p^2}$, of the same order q.
- The bilinear map is built from the Weil pairing, and this is used in the transformed version, $\hat{e}(P, \phi(Q))$, with $P, Q \in G_1$, which is non-degenerate.
- Group G_T is the subgroup of order q of $\mathbb{F}_{p^2}^*$, which is isomorphic to the subgroup of $\mathbb{Z}_p^2$ that contains all the q^{th} roots of unity.

[2]This condition guarantees that $E[q] \nsubseteq E(\mathbb{F}_p)$.

- Hashing from identities into G_1 is achieved by taking advantage of an interesting property of this curve. The y coordinates of its points span all of $\mathbb{Z}_p$, i.e. for each $y \in \mathbb{Z}_p$ there is only one point that has this y-coordinate, and this point has x-coordinate $x = (y^2 - 1)^{1/3}$. Hence, hashing is performed in two steps. First the identity is hashed into $\mathbb{Z}_p$, and this value is taken as y-coordinate of the target point. An x-coordinate is then easily calculated as $x = (y^2 - 1)^{(2p-1)/3}$. The resulting point is not known to be of order q, therefore it is multiplied by $l = (p + 1)/q$ to produced the desired point in G_1.

- Hashing from G_T into the message space is straightforward.

Curiously, around the same time the Boneh and Franklin scheme was published, Cocks [15] published another identity based encryption algorithm that solved the same problem. We don't discuss it here, as it it is not pairing based.

Since 2001, many pairing based encryption algorithm variants have been proposed. Here we will mention only two: Libert and Quisquater's identity based signcryption algorithm [30], and Boyen's *swiss army knife* multipurpose identity based cryptosystem [13].

The concept of signcryption was proposed in 1997 by Zheng [47] as a primitive that combines the functionality of a digital signature and encryption scheme (integrity, authentication, nonrepudiation and confidentiality) at a much lower cost than separately conducting these operations one after the other. Signcryption schemes are made out for algorithms: *Setup*, *Key Generation*, *Signcryption*and *Unsigncryption*, where the latter returns bottom ($\perp$) in case of a verification error.

The security model for the signcryption primitive is a combination of the models for encryption and digital signatura. In the case of identity based signcryption, the important security notions are semantical security against an adaptive chosen ciphertext attack, and existential unforgeability against a chosen message attack. The only difference in this latter case, is that instead of producing a valid message/signature pair, the adversary must produce a valid signcryption message [33].

Libert and Quisquater's identity based signcryption scheme is described in Table 5.8 It is proven to be IND-CCA2 secure (in its identity based variant) in the RO model, and under the assumption that the Decision BDH problem is hard. Since it is based in Hess's identity based signature scheme [25], it is also UF-CMA secure in the RO model, under the CDH assumption.

Boyen's work [13] takes a more general approach to the signcryption problem. The author proposes a set of cryptographic algorithms that allow for efficient identity based digital signatures, encryption and signcryption, over a common set of global parameters. The primitives that comprise the scheme are: *Setup*, *Private key extraction*, *Signature*, *Encryption*, *Decryption* and *Verification*.

This author also proposes a broader security model, that elliminates ambiguities in the interactions between the different primitives. The important concepts are:

Public Parameters	$(G_1, G_T, \hat{e}, H_i, P, P_{Pub})$	Two cyclic groups of prime order q and a bilinear map between them $\hat{e} : G_1 \times G_1 \rightarrow G_T$, such that the DBDH problem is hard. Three hash functions H_1, H_2 and H_3 that take identity representations to G_1, arbitrary bit strings to $\mathbb{Z}_q^*$, and elements of G_T into the n-bit-long message space, respectively. A random generator P of G_1, and the KGC's public key $P_{Pub} \in G_1$
KGC Master Key	s	The discrete logarithm of P_{Pub} to the base P, or $P_{Pub} = sP$.
Priv. Key Extraction	d_{ID}	Given a finite bit string representing the ID, the private key is $d_{ID} = sQ_{ID} = sH_1(ID)$.
Signcryption	$C = (U, V, c)$	$U = xP$, with x random, $r = H_2(U\|\|m)$, $V = xP_{Pub} + r * d_{ID_A}$, $c = m \oplus H_3(\hat{e}(P_{Pub}, Q_{ID_B})^x)$
Unsigncryption	m or $\perp$	$m = c \oplus H_3(\hat{e}(U, d_{ID_B}))$, $r = H_2(U\|\|m)$ **if** $\hat{e}(V, P) = \hat{e}(Q_{ID_A}, P_{Pub})^r \hat{e}(U, P_{Pub})$ **then** m **else** $\perp$

Table 5.8: Libert and Quisquater's identity based signcryption scheme

- **message confidentiality** – the formal definition is similar to IND-CCA2, in its identity based variant, but with *insider security*. This means that the adversary knows the private key associated with the signing identity.
- **signature non-repudiation** – the formal definition is relative only to the cleartext message, and is equivaltent toEUF-CMA in an identity based setting. Again, for insider-security, the adversary knows the recipient's private key.
- **ciphertext unlinkability** – this guarantees that non-repudiation applies only to the cleartext message, and not to the ciphertext. The formal definition requires only that it is easy to construct a valid ciphertext that is indistinguishable from a challenge cryphertext for which the encrypted signature and message are known.
- **ciphertext authentication** – this, in combination with the previous property, means that the encryption scheme includes MAC-like functionality. The formal definition is relative to the whole ciphertext, and it is based on UF-CMA with *outsider security* i.e. the adversary does not know the recipient's private key.
- **ciphertext anonymity** – this means that only the legitimate recipient can find out who the sender is. The formal definition is based on an IND-CCA2-like game, where the adversary must choose between sender identities, rather than cleartexts.

The details of the scheme are shown in Table 5.9. Boyen also provides an intuitive description of the scheme:

- The signature is a pair, where j is a commitment to a random r, and v is a value that depends on both r and the message m.

Public Parameters	$(G_1, G_T, \hat{e}, H_0..H_4, P, P_{Pub})$	Two cyclic groups of prime order q and a bilinear map between them $\hat{e} : G_1 \times G_1 \rightarrow G_T$, such that the BDH problem is hard. Five hash functions: $H_0 : \{0,1\}^* \rightarrow G_1^*$, $H_1 : G_1^* \times \{0,1\}^* \rightarrow \mathbb{F}_q^*$, $H_2 : G_T^* \rightarrow \{0,1\}^{\lfloor \log p \rfloor}$, $H_3 : G_T^* \rightarrow \mathbb{F}_q^*$ and $H_4 : G_1^* \rightarrow \{0,1\}^*$. A random generator P of G_1, and the KGC's public key $P_{Pub} \in G_1$.
KGC Master Key	s	The discrete logarithm of P_{Pub} to the base P, or $P_{Pub} = sP$.
Priv. Key Extraction	d_{ID}	Given a finite bit string representing the ID, the private key is $d_{ID} = sQ_{ID} = sH_0(ID)$.
Sign	$S = \langle (j, v); r \rangle$	$j = rQ_{ID_A}$, with r random, $h = H_1(j, m)$ $v = d_{ID_A}^{r+h}$, where (j, v) is the signature, and r is passed on to subsequent stages.
Encrypt	(x, y, z)	$u = \hat{e}(d_{ID_A}, Q_{ID_B})$, $k = H_3(u)$, $x = j^k$, $w = u^{kr}$, $y = H_2(w) \oplus v$ $z = H_4(v) \oplus (ID_A \| \| m)$
Decrypt	$(ID_A, \hat{m}, \hat{j}, \hat{v})$	$\hat{w} = \hat{e}(x, d_{ID_B})$, $\hat{v} = H_2(\hat{w}) \oplus \hat{y}$, $(ID_A \| \| \hat{m}) = \hat{z} \oplus H_4(\hat{v})$, $\hat{u} = \hat{e}(Q_{ID_A}, d_{ID_B})$ $\hat{k} = H_3(\hat{u})$, $\hat{j} = \hat{x}^{\hat{k}^{-1}}$.
Verify	$\top$ or $\bot$	$\hat{h} = H_1(\hat{j}, \hat{m})$ **if** $\hat{e}(P, \hat{v}) = \hat{e}(P_{Pub}, \hat{j} \cdot Q_{ID_A}^{\hat{h}})$ **then** $\top$ **else** $\bot$

Table 5.9: Boyen's identity based swiss army knife signcryption scheme

- Encryption is performed in two layers.
- The inner layer encrypts j, the commitment, into x using a basic key agreement that uses the sender's private key and the recipient's public key.
- The outer layer encrypts signature and the bit string comprised of the sender's identity and the message itself into y and z respectively. y is obtained using an identity based encryption mechanism using pairings, but re-using the randomization of r. z is obtained masking the cleartext with an image of the signature component v.

This scheme is proven to be secure with respect to all the requirements layed out above, in the RO model, and under the BDH hardness assumption.

Chapter 6

Recent Developments

6.1 Koblitz and Menezes, 2005

In [28], Koblitz and Menezes discuss the future of pairing based cryptography. This is done taking into account an increase in the security standards for public key cryptosystems that, in their view, is to be expected in order to achieve security levels equivalent to 128-, 192-, or 256-bit AES keys.

The main problems with pairing based cryptography pointed out in this work are the following:

- The fact that the DLP problem must be hard in the target group means that q^k (with q the size of the field underlying the elliptic curve, and k embedding degree of the curve) must be approximately the size of an RSA module which, at high security levels, can be very large (at least 15360 bits for security equivalent to 256-bit AES). This is very costly in terms of efficiency.

- The BDHP is a new problem. The assumption that it is hard is essencial to the security claims in most pairing-based protocols. So far there is no evidence that this problem is not easier than the DH and DL problems in the source elliptic curve. Similarly, no evidence exists that the BDHP is not easier than the DHP in the finite field $\mathbb{F}_{q^k}$. If anything, the evidence points in the opposite direction. For example, Koblitz and Menezes point out that if, as a consequence of a theorem by Verheul [29], if the BDHP is ever proven to be equivalent to the DHP in $\mathbb{F}_{q^k}$, then this will render both problems easy.

The estimated parameter sizes for pairing based cryptosystems at high security levels, are shown in Table 6.1. They are calculated based on the required sizes for the target field and the source sub-group, in order to achieve a particular security level. Note that, in typical applications of pairing-based crytography, we will have k approximately equal to the ratio between q^k and r (also shown

Security level	80-bit AES	128-bit AES	192-bit AES	256-bit AES
r bit-length	160	256	384	512
p^k bit-length	1024	3072	8192	15360
r^k/r ratio	6.4	12	21.33	30

Table 6.1: Koblitz and Menezes' estimates for required parameter sizes at high security levels.

in Table 6.1) for optimal performance. This is the case, for example in the BLS short signatures discussed in Section 5.3.

Koblitz and Menezes draw the following conclusions from this data:

- For higher efficiency, both q and r should be Solinas primes. Therefore, there is a need for curve generation methods that achieve the desired values for k under this constraint for high security levels. No such method exists for supersingular curves, for $k \geq 2$ (apart from a result by Barreto et al. [4] that produces near-optimal solutions for 128-bit security).

- For the higher security levels, and $k \neq 2$, it is not advisable to use supersingular curves, since the number of choices for the embedding degree are small, and it may simply not be possible to find a suitable curve.

- At high security levels, there is a threshold, above which the additional exponentiations required by the Tate pairing render it less efficient than the Weil pairing.

- Curves with an embedding degree of one have never been seriously considered for pairing-based systems, but they constitute a valid solution. Koblitz and Menezes propose a family of curves with $k = 1$, for which it is possible to find satisfactory curves at high security levels. They have the disadvantage, however, that the arithmetic must be done in large fields, for which there are no efficiency shortcuts.

- For $k = 2$, supersingular curver provide highly efficient solutions for pairing based systems, and there is no fundament to rule them out as not secure.

- For $k > 2$ it is advisable to stick to curves over fields $\mathbb{F}_q$, where the field extension $\mathbb{F}_{q^k}$ is a *pairing-friendly field*. This means that $q = 1 \pmod{12}$, and k is of the form $2^i 3^j$. The structure of this type of field permits constructing the extension as a tower of quadratic and cubic extensions which, in turn, allows for significant speed-ups in multiplications over the full extension.

6.2 Smart and Vercauteren, 2005

In [45], Smart and Vercautered present a clarification of two issues that are critical in pairing based systems: the hard problems underlying pairing-based systems, and the need for a computable isomorphism between the source groups of the pairing. In their work, these authors focus on the general case of an asymmetric pairing $\hat{p} : G_1 \times G_2 \rightarrow G_T$, and base their definitions on the notion of a *pairing problem instance*: $\Gamma = (q, G_1, G_2, G_T, P_1, P_2, \hat{p})$, where q is the (prime) order of the groups, and P_i are generators for groups G_i.

The hard problems that Smart and Verceuteren identify, based on a particular pairing problem instance Γ, are variations of the Bilinear Diffie Hellman problem and Computational Diffie Hellman problem:

- $BDH_{i,j,k}$ **problem:** *Given* Γ *and* $i, j, k \in 1, 2$*, and also* aP_i*,* bP_j *and* cP_k*, with* $a, b, c \in \mathbb{F}_q$*, compute* $\hat{p}(P_1, P_2)^{abc}$*.*

- $co - BDH_{j,k}$ **problem:** *Given* Γ *and* $j, k \in 1, 2$*, and also* aP_1*,* aP_2*,* bP_j *and* cP_k*, with* $a, b, c \in \mathbb{F}_q$*, compute* $\hat{p}(P_1, P_2)^{abc}$*.*

- $BDH^{\phi}_{i,j,k}$ **problem:** *Given* Γ *and* $i, j, k \in 1, 2$*, and also* aP_i*,* bP_j *and* cP_k*, with* $a, b, c \in \mathbb{F}_q$*, compute* $\hat{p}(P_1, P_2)^{abc}$*. The difference to* $BDH_{i,j,k}$ *is that the adversary has access to an oracle which computes an isomorphism* $\phi : G_2 \rightarrow G_1$ *(but not necessarily* ϕ^{-1}*).*

- $CDH_{i,j,k}$ **problem:** *Given* Γ *and* $i, j, k \in 1, 2$*, and also* aP_i *and* bP_j*, with* $a, b \in \mathbb{F}_q$*, compute* cP_k*.*

- $CDH^{\phi}_{i,j,k}$ **problem:** *Given* Γ *and* $i, j, k \in 1, 2$*, and also* aP_i*,* bP_j *and* cP_k*, with* $a, b, c \in \mathbb{F}_q$*, compute* $\hat{p}(P_1, P_2)^{abc}$*. Again, the difference to* $CDH_{i,j,k}$ *is that the adversary has access to an oracle which computes an isomorphism* $\phi : G_2 \rightarrow G_1$ *(but not necessarily* ϕ^{-1}*).*

The ways in which these problems relate to each other are not obvious:

- In the presence of a computable isomorphism, the $BDH_{i,j,k}$ and the $BDH^{\phi}_{i,jk}$ problems are equivalent. In its absence, the former is at least as hard as the latter.

- Instances of $BDH_{i,j,k}$ where $i + j + k$ are the same (the number of points taken from each group is the same) are obviously equivalent. There appears to be no relationship between instances of this problem where the permutations differ.

- In the case of $BDH^{\phi}_{i,j,k}$, the situation is similar, apart from the fact that the availability if the isomorphism oracle permits mapping points of G_2 onto G_1. This implies that instances where $i + j + k$ is smaller (i.e. more points taken from G_1) are at least as hard as permutations with higher $i + j + k$.

- For $co - BDH_{j,k}$, since we have the two powers of a in both groups, it is easy to see that it is not harder than $BDH_{i,j,k}$. On the other hand, it is at least as hard as $BDH^{\phi}_{2,j,k}$.
- Relations between the CDH variants are similar to the previous ones.
- Finally, the $CDH_{i,j,k}$ problem is at least as hard as the $BDH_{i,j,k'}$ problem, as long as $k \neq k'$. This is because if we are able to calculate abP_k and $cP_{k'}$ in different groups, then the pairing gives us the solution to $BDH_{i,j,k}$.

A similar approach can be taken to defining underlying gap-Diffie-Hellman variants.

These definitions allow for a more general analysis of pairing based cryptosystems. Namely they permit defining at a higher level of abstraction what are the restrictions on groups G_1 and G_2, and taking implementation decisions based on that. The authors present such an analysis for Boneh et al.'s identity based encryption scheme [11] and Boneh et al.'s short signature scheme [12], which exposes the essence of these cryptosystems, and explores the different implementation options that they allow for.

The need for a computable isomorphism between G_2 and G_1 is said to arise mostly in security proofs. On the other hand, Smart and Vercauteren point out that, for the most promising Tate pairing implementation (Barreto et al.'s work on MNT curves [2]) such a computable isomorphism does not exist. It is therefore still an open problem to find an efficient pairing implementation, suitable for cryptographic applications, that meets all the requirements imposed by the pairing based cryptographic schemes in literature.

In addition to computable isomorphisms, there is another restriction on these groups that is very important for cryptographic applications: whether a group is *randomly samplable*. This means that it is possible to randomly choose an element of the group without using a power of the generator (i.e. not knowing its discrete log). Only under this condition is it possible to obtain hash functions that map onto these groups.

Smart and Vercauteren propose a new choice for group G_2 that will allow for both a computable isomorphism and an efficient MNT-curve-based implementation. However, this group requires a larger representation, and appears not to be randomly samplable.

6.3 Barreto and Naehrig, 2005

Barreto and Naehrig [3] propose a surprisingly simple method of constructing MNT curves of prime order with embedding degree 12. This is significant for schemes such as the BLS short signatures where the size of the underlying field must be kept to a minimum. With an embedding degree of 12 it is possible to setup a set of parameters providing 128-bit security working over a base field size of 256 bits, and an extension field size of 3072 bits. This optimizes the bandwidth/security ratio. In fact, every pairing based cryptographic protocol

implemented over this type of set-up would benefit from the decrease in parameter sizes (this would be noticeable in all parameters except the extended field size).

The problem with this method is that one can not control the Hamming weight of the system parameters, namely the curve order. This severely limits the efficiency that can be achieved in pairing calculations, even though the proposed curves allow for the usual Tate pairing calculation optimizations, as well as elliptic curve point and pairing value compression techniques.

Bibliography

[1] R. Balasubramanian and Neal Koblitz. The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 11(2):141–145, Spring 1998.

[2] Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. On the selection of pairing-friendly groups. Cryptology ePrint Archive, Report 2003/086, 2003.

[3] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. Cryptology ePrint Archive, Report 2005/133, 2005.

[4] Paulo S.L.M. Barreto, Hae Y. Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. Cryptology ePrint Archive, Report 2002/008, 2002.

[5] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. *Lecture Notes in Computer Science*, 1462:26–??, 1998.

[6] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In *EUROCRYPT*, pages 259–274, 2000.

[7] Mihir Bellare, Anand Desai, Eron Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *FOCS '97: Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS '97)*, page 394. IEEE Computer Society, 1997.

[8] Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Security proofs for identity-based identification and signature schemes., 2004.

[9] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[10] Susan Hohenberger Ben Adida and Ronald L. Rivest. Separable identity-based ring signatures: Theoretical foundations for fighting phishing attacks, 2005.

[11] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *Lecture Notes in Computer Science*, 2139:213–??, 2001.

[12] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. *J. Cryptology*, 17(4):297–319, 2004.

[13] Xavier Boyen. Multipurpose identity-based signcryption : A swiss army knife for identity-based cryptography. Cryptology ePrint Archive, Report 2003/163, 2003.

[14] Jae Choon Cha and Jung Hee Cheon. An identity-based signature from gap diffie-hellman groups. Cryptology ePrint Archive, Report 2002/018, 2002.

[15] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 360–363. Springer-Verlag, 2001.

[16] Iwan Duursma and Hyang-Sook Lee. Tate-pairing implementations for tripartite key agreement. Cryptology ePrint Archive, Report 2003/053, 2003.

[17] G. Frey, M. Muller, and H.-G. Ruck. The tate pairing and the discrete logarithm applied to elliptic curve cryptosystems, 1999.

[18] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 537–554, 1999.

[19] Steven D. Galbraith. Supersingular curves in cryptography. *Lecture Notes in Computer Science*, 2248:495–??, 2001.

[20] Steven D. Galbraith, Keith Harrison, and David Soldera. Implementing the tate pairing. In *ANTS-V: Proceedings of the 5th International Symposium on Algorithmic Number Theory*, pages 324–337. Springer-Verlag, 2002.

[21] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[22] R. Granger, D. Page, and M. Stam. Hardware and software normal basis arithmetic for pairing based cryptography in characteristic three. Cryptology ePrint Archive, Report 2004/157, 2004.

[23] R. Granger, D. Page, and M. Stam. On small characteristic algebraic tori in pairing-based cryptography. Cryptology ePrint Archive, Report 2004/132, 2004.

[24] P1363 Working Group. P1363 - standard specifications for public-key cryptography.

[25] Florian Hess. Efficient identity based signature schemes based on pairings. In *SAC '02: Revised Papers from the 9th Annual International Workshop on Selected Areas in Cryptography*, pages 310–324, London, UK, 2003. Springer-Verlag.

[26] Tetsuya Izu and Tsuyoshi Takagi. Efficient computations of the tate pairing for the large mov degrees. In *Information Security and Cryptology - ICISC 2002, 5th International Conference Seoul, Korea, November 28-29, 2002, Revised Papers*, Lecture Notes in Computer Science, pages 283–297. Springer, 2002.

[27] Antoine Joux. A one round protocol for tripartite diffie-hellman. In *ANTS-IV: Proceedings of the 4th International Symposium on Algorithmic Number Theory*, pages 385–394, London, UK, 2000. Springer-Verlag.

[28] Neal Koblitz and Alfred Menezes. Pairing-based cryptography at high security levels. Cryptology ePrint Archive, Report 2005/076, 2005.

[29] Arjen K. Lenstra and Eric R. Verheul. The xtr public key system. In *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, Lecture Notes in Computer Science, pages 1–19. Springer, 2000.

[30] Benoît Libert and Jean-Jacques Quisquater. New identity based signcryption schemes from pairings. Cryptology ePrint Archive, Report 2003/023, 2003.

[31] Benoît Libert and Jean-Jacques Quisquater. The exact security of an identity based signature and its applications. Cryptology ePrint Archive, Report 2004/102, 2004.

[32] M. Maas. Pairing-based cryptography. Master Thesis, 2004.

[33] John Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002.

[34] Alfred 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.

[35] Victor S. Miller. The weil pairing, and its efficient calculation. *J. Cryptology*, 17(4):235–261, 2004.

[36] Miyaji, Nakabayashi, and Takano. New explicit conditions of elliptic curve traces for FR-reduction. *TIEICE: IEICE Transactions on Communications / Electronics / Information and Systems*, 2001.

[37] D. Page, N.P. Smart, and F. Vercauteren. A comparison of mnt curves and supersingular curves. Cryptology ePrint Archive, Report 2004/165, 2004.

[38] Kenneth G. Paterson. Id-based signatures from pairings on elliptic curves. Cryptology ePrint Archive, Report 2002/004, 2002.

[39] Ronald L. Rivest and Burt Kaliski. The rsa problem. *To appear in Encyclopedia of Cryptography and Security*, 2003.

[40] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairings. In *SCIS 2001*, 2001.

[41] Michael Scott and Paulo S. L. M. Barreto. Compressed pairings. Cryptology ePrint Archive, Report 2004/032, 2004.

[42] Michael Scott and Paulo S.L.M Barreto. Generating more mnt elliptic curves. Cryptology ePrint Archive, Report 2004/058, 2004.

[43] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53. Springer-Verlag New York, Inc., 1985.

[44] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*. Springer Verlag, 1986.

[45] Nigel Smart and Frederik Vercauteren. On computable isomorphisms in efficient pairing based systems. Ecrypt Draft, 2005.

[46] Eric R. Verheul. Evidence that xtr is more secure than supersingular elliptic curve cryptosystems. In *EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pages 195–210, London, UK, 2001. Springer-Verlag.

[47] Yuliang Zheng. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer, 1997.