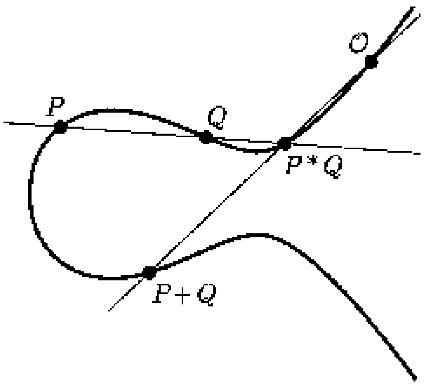
E.N.S. LYON - MMA2

PRACTICAL PUBLIC KEY CRYPTOSYSTEMS

It is possible to write endlessly on elliptic curves. **S.Lang**



Annie Chateau
Training period of summer 1998

Under direction of Ivan Damgård
Aarhus University
Denmark

Subject: The purpose of this project is to study the construction and applicability of public key cryptosystems based on concepts from various kinds of discrete mathematics. The most directly relevant cases are algebraic structures that give rise in various ways to large finite Abelian groups. The simplest examples come from the ring of integers modulo large prime or composite numbers, leading to well-known systems like RSA. But perhaps a more interesting example, both from a mathematical and implementation point of view, is the group of rational points on an elliptic curves over a finite field. Also class groups could be relevant. The construction of elliptic curves with prescribed properties is currently the subject of intense studies worldwide, due both to the immediate relevance from an implementation point of view, and to the pure mathematical interest. This subject forms a natural starting point for the project.

Contents

List of Figures

Introduction

The problematic and the actors of cryptography are the following:
Alice and Bob are good friends, and want to communicate. Unfortunately, their worse enemy Oscar tries to know what they say. So they must use a cryptosystem to communicate.
We will particularly be interested in one aspect of the cryptography, the public key systems. "Public key" means that some information is available in public area, and can be used to encrypt a message. But the decryption key is secret, so that only the person who receives the message can decrypt it. This concept is developed in section 1 .

What is a cryptosystem? In a formal way, we can describe it with the following definition:

Definition 1 Cryptosystem
A cryptosystem is a five-tuple ($\mathcal{P}$, $\mathcal{C}$, $\mathcal{K}$, $\mathcal{E}$, $\mathcal{D}$) where:
- $\mathcal{P}$ is the set of possible plaintexts (texts that are not encrypted)
- $\mathcal{C}$ is the set of possible ciphertexts (texts that are encrypted)
- $\mathcal{K}$ is the set of possible keys
- $\mathcal{E}$ is the set of encryption rules depending on a key: if K is an element of $\mathcal{K}$,
$e_K\, \mathcal{P} \to \mathcal{C}$ is the corresponding encryption rule
- $\mathcal{D}$ is the set of decryption rules depending on a key: $d_K\, \mathcal{C} \to \mathcal{P}$
and which verifies: $\forall K \in \mathcal{K}\ \forall x \in \mathcal{P}\ d_K(e_k(x)) = x$

It is simpler to say that Alice has a method to encrypt the messages that she wants to send to Bob, and Bob has a method to decrypt the messages that Alice sent to him. The channel by which they communicate is not really sure, and Oscar can get the encrypted message. Alice and Bob don't know what sort of computer Oscar has, so they must find the securest method to encrypt their messages. Oscar is said to be a cryptanalyst. Cryptanalyse is the dark side of cryptology, but it is necessary to search ways of breaking a system: if you succeed in breaking it, you are sure that it cannot be used with security, if you don't and if a lot of people who tried have not succeeded too, you can reasonably think that the system is quite secure for the moment.
In private-key cryptography, the same key is used to encipher and to decipher. Private-key cryptography is also called symmetric cryptography. In public-key cryptography, there are two keys: a private key owned by the recipient and a public key that everyone can use. So in the definition above, we have to separate two fields in the key K when we talk about public-key cryptography

or asymmetrical cryptography. The keys are then of the form:

$$K = (k_{private}, k_{public})$$

So by e_K we mean $e_{k_{public}}$ and by d_K we mean $d_{k_{private}}$.

Another aspect of public-key cryptography is authentication of the identity of the sender. How could Bob be sure that the message is Alice's one, and not a false message from Oscar? This is the purpose of signature schemes, that use quite the same methods as encrypting. Both can be juxtaposed in order to obtain a secure and origin-certified communication.
Formally, a signature scheme can be defined by:

Definition 2 Signature scheme
A signature scheme is a five-tuple ($\mathcal{P}$, $\mathcal{A}$, $\mathcal{K}$, $\mathcal{S}$, $\mathcal{V}$) where
- $\mathcal{P}$ is the set of possible messages
- $\mathcal{A}$ is the set of possible signatures
- $\mathcal{K}$ is the set of possible keys $K = (k_{private}, k_{public})$
- $\mathcal{S}$ is the set of signing algorithms depending on a key: if K is an element of $\mathcal{K}$, $sig_{k_{private}}$ $\mathcal{P} \to \mathcal{A}$ is the corresponding signing algorithm
- $\mathcal{V}$ is the set of verification algorithms: $ver_{k_{public}}$ $\mathcal{P} \times \mathcal{A} \to \{True,\ False\}$ and which verifies: $\forall x \in \mathcal{P}\ \forall y \in \mathcal{A}$

$$ver(x,y) = \begin{cases} \text{true} & \text{if } y = sig(x) \\ \text{false} & \text{if } y \neq sig(x) \end{cases}$$

1 Public key cryptography

1.1 Principles

In classical private key cryptosystems, Bob and Alice need to communicate the key by a very secure system, before they send any encrypted message. Public key cryptosystems resolve this problem since the encoding key is public. The security is provided by the fact that the decryption rule d_K is very hard, or very long if we consider the computational aspect, to find, even if we know the key k_{public}. One drawback of this system is that Oscar could try the key on all the possible plaintexts and compare it with the encrypted message he got during the transfer, and so that he can determine the original plaintext. But it can take a so long time that it would be very hard to decrypt long messages.
The original idea of public key cryptosystem is due to Diffie and Hellman in 1976. It led to the famous RSA cryptosystem by Rivest, Shamir and Adleman in 1978.
This idea seemed to be so efficient that there were a great development of public key cryptography in the following years. Cryptosystems that are more secure have been found, and some other has been proved to be not so secure because there have been found efficient attacks against them. (We will develop this point on 2.2.2) . We will describe some of these systems that are still considered as secure.

1.2 Trapdoor one-way functions and computational security

The public key cryptosystems mainly use the idea of "one-way" functions. A one-way function is a function that is easy to compute but very hard to invert, so that it is nearly impossible without more informations. Those informations are called a trapdoor. If such a function exists, Alice uses the function to encrypt messages and Bob, who has supplementary informations, invert the function the get the plaintext. But Oscar, even with his big computer, cannot invert the function nor find the plaintext.
It is easy to see that a one-way function without any trapdoor is totally useless in cryptosystems, because it would take the same time to Bob and Oscar to decrypt the message! But it could be used in signature schemes, for example in the one-way signature schemes.(See B.1.2)
A lot of systems where proposed using a certain function which is supposed to be one-way: the discrete logarithm. If n is an integer, and b a positive integer, the b^{th}-power of $x \in \mathbb{Z}_n$ is $f(x) = x^b \mod n$. The value of x is chosen,

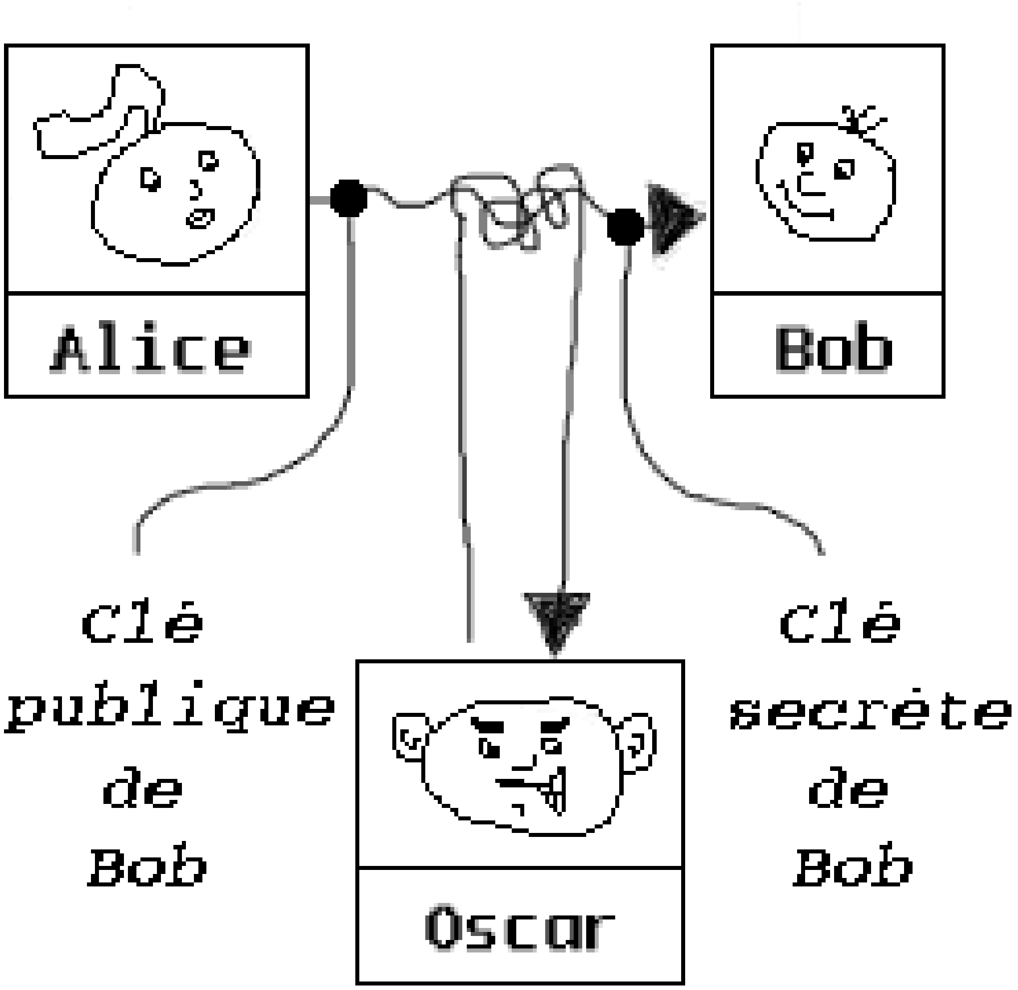


Figure 1: Public key cryptography protocol

and the function takes b as an argument. The invert of this function is called the discrete logarithm and if $y = x^b \mod n$, then $b = \log_x y$. It is supposed to be really difficult to find b when you know $f(x)$, x and n in certain cases. There is no known proof of the fact that this function is one-way, but the most part of cryptosystems work on this hope.
Systems using discrete logarithm are said to be based on the computational difficulty of the discrete logarithm. Another security criteria is the factorisation of large numbers with large prime factors. For example, we can say that the RSA cryptosystem is based on the difficulty of factorising large numbers. (See 2.1) Although the word "difficulty" could seem vague, it is based on the precise theory of complexity of the algorithms.

1.3 Complexity of algorithms

The speed of an algorithm is based on an operation that is supposed to take a time 1 to be executed. And it is defined as an asymptotic behaviour depending on the size of the input of the number of executions of this instruction (which can be a multiplication, of a comparison, or something else ...) when the size of the input raise. For example we say that an algorithm is polynomial when is complexity is in $O(P(n))$ where $P(n)$ is a polynomial function of the size n of the input. In practice, we consider that polynomial algorithms don't take "a lot of time", unlike exponential algorithms.
In the following, we will meet some algorithms that run in "sub-exponential time". It means their asymptotic running time is of the form $O(e^{((c+o(1))(\ln n)^{\alpha}(\ln \ln n)^{1-\alpha})})$, where $0 < \alpha < 1$ and c are constants. It is faster than any exponential algorithm but slower than any polynomial algorithm.
A probabilistic polynomial (resp. sub-exponential) time algorithm is an algorithm where a random factor plays a role, and whose expected running time is bounded by a polynomial (resp. sub-exponential) function depending on the size of the input.

1.4 Broken cryptosystems

We won't talk a lot about broken cryptosystems. A broken cryptosystem is proved to be insecure, it means someone has found an algorithm which is in practice easy to compute (a polynomial algorithm for example). However it is one interesting aspect of cryptography that can give the feeling of what is secure or not.

2 Tools of public key cryptography

2.1 The RSA cryptosystem and signature scheme: the integer factorisation problem

This system is one of the best known cryptosystem, and usually the first introduced when it comes to talk about cryptography. It is one of the reason we won't develop a lot about it. One other reason is that there now exist more secure systems.

2.1.1 Principles

Bob chooses a large number of the form $n = pq$, where p and q are large prime numbers. The plaintexts and the ciphertexts are coded in the finite ring $\mathbb{Z}_n$. The keys are elements of: $\mathcal{K} = \{(a,b) \backslash\ ab \equiv 1 \pmod{\Phi(n)}\}$, where Φ is the Euler function.
For a given key $K = (a,b)$, the encryption function is:

$$e_K(x) = x^b \bmod n$$

and the decryption function:

$$d_K(y) = y^a \bmod n$$

The public key consists of (n,b) and the values p,q,a are Bob's secret trapdoor. To find such a key, Bob firstly find p and q, computes $n = pq$ and $\Phi(n) = (p-1)(q-1)$, he chooses a random b $(0 < b < \Phi(n))$ relatively prime to $\Phi(n)$. He computes $a \equiv b^{-1} \bmod \Phi(n)$ with the euclidian algorithm, and finally publishes his public key (n,b).

2.1.2 Security

If Oscar can factorise the number n into pq, he can easily compute the number $\Phi(n)$ and the secrete trapdoor a. So he can decrypt the messages sent to Bob. A lot of factorisation algorithm were developed in order to crack the RSA system. Some are now very efficient (see 2.1.3), but they are not enough powerful to factorise very large numbers (that have more than 200 decimal digits) with the recent computers.
It is enough for Oscar to have the number $\Phi(n)$, but it is as difficult to get this number as to factorise n.

2.1.3 Attacks

There are two major axes concerning factorisation algorithms:

- multi-polynomial quadratic sieve: the basic quadratic sieve take its origins in a idea of M.Kraitchik. It has been studied and developed by C.Pomerance in 85, and improved by R.D.Silverman in 87. The idea and the algorithm can be found in [19]. The basic idea is the following: if you can find integers A and B such that, for a number N to be factorised: $A^2 \equiv B^2 \mod N$, $A \not\equiv B \mod N$ and $A \not\equiv -B \mod N$, then you get two proper factors of N: $gcd(A+B, N)$ and $gcd(A-B, N)$. It runs in time $O(e^{\sqrt{\ln N \ln \ln N}})$. This algorithm is a general purpose algorithm, it means that the running time depends only on the size of N.

- elliptic curves: It was first discovered by H.Lenstra Jr. in 1985. The basic idea is the following: You can compute the same operation on the group of an elliptic curve defined over a ring $\mathbb{Z}_N$ as over a field $\mathbb{Z}_{p^m}$, if you don't have an non invertible denominator.(See 3.1.3). You choose a random elliptic curve and a random point P on it. You compute $k \times P$ where k is divisible by each prime number less than a bound. If one denominator is non invertible, it is probably a factor of N.It runs in time $O(e^{\sqrt{2 \ln p \ln \ln p}})$. The ECM is a special purpose algorithm, it means it exploit special features of the number to be factorised, for example the fact that it has small prime factors.

Other methods are efficient but run slower, and so are not convenient for cryptanalysists. For example, there are the Pollard ρ method, the Pollard $p-1$ method, the number field sieve, the continued fraction method ...

2.1.4 Signature scheme

The RSA signature scheme is based on the same principle as the RSA cryptosystem: let $n = pq$ a big integer, product of two big integer factors, and $\mathcal{K} = \{(a, b) \backslash\ ab \equiv 1 \pmod{\Phi(n)}\}$.
The values n and b are public and the values p, q, a are secret.
The signature algorithm is

$$sig_K(x) = x^a \mod n.$$

The verification algorithm is

$$ver_K(x, y) = true \Leftrightarrow x = y^b \mod n.$$

2.1.5 Rabin-Miller cryptosystem

The Rabin-Miller cryptosystem is also based on the factorisation problem. Let $n = pq$ be the product of two distinct (large) primes such as $p \equiv 3 \mod 4$ and $q \equiv 3 \mod 4$.Let $\mathcal{K} = \{(n, p, q, B) \backslash 0 \le B \le n-1\}$. For a given key $K = (n, p, q, B)$ the encryption function is:

$$e_k(x) = x(x + B) \mod n$$

and the decryption function:

$$d_k(y) = \sqrt{\frac{B^2}{4} + y} - \frac{B}{2} \mod n$$

The public key consists of (n, B), while p and q are kept secret.
Finding x knowing y is equivalent to a mere problem of extracting the quadratic root of a number modulo n, which is possible when you know p and q, but is very difficult when you don't.
One drawback of this system is that there are four possible plaintexts for a given ciphertext. Indeed you have a choice between four solutions when you solve the equation

$$z^2 \equiv \frac{B^2}{4} + y \mod n$$

So the recipient must choose one of them ...

2.2 The discrete logarithm problem

We will describe different types of discrete logarithm problems. They follow all the same scheme, but the world where they live is quite different, and this can generate important computation differences, especially with the introduction of the elliptic curves.

2.2.1 Discrete logarithm in $\mathbb{Z}_p$

It is very simple to formulate: the world where we live is the field $\mathbb{Z}_p$ (of course, p is prime). There exists a generator of the multiplicative group $\mathbb{Z}_p^*$, we choose one, α. For each element β of the group $\mathbb{Z}_p^*$, there exists a unique integer $0 \le l \le p-2$ such as $\beta \equiv \alpha^l \mod p$.
The integer l is called the discrete logarithm of β to the base α.
Computing the lth power of α modulo p is not very difficult and is can be done in polynomial time (see figure 2).
However it is a more complicated thing to compute l knowing α, β and p.

Let $l = \sum_{i=0}^{r} v_i 2^i$ the binary decomposition of l
Set $result \leftarrow \alpha$
For $i = r - 1$ to 0 do
    set $result \leftarrow result^2 \mod p$
    if $v_i = 1$ then set $result \leftarrow result \times \alpha \mod p$

Figure 2: Exponentiation algorithm modulo p

For some ideas of this, see 2.2.2.

The most famous cryptosystem using the discrete logarithm problem in $\mathbb{Z}_p$ is the El-Gamal cryptosystem.
In this system, we define $\mathcal{K} = \{(p, \alpha, a, \beta) \backslash \beta \equiv \alpha^a \mod p\}$.
For a given key $k = (p, \alpha, a, \beta)$ and a random number $k \in \mathbb{Z}_p{}^*$, the encryption function is:

$$e_K(x, k) = (y_1, y_2)$$

where

$$\begin{aligned} y_1 &= \alpha^k \mod p \\ y_2 &= x\beta^k \mod p \end{aligned}$$

and the decryption function is:

$$d_K(y_1, y_2) = y_2(y_1^a)^{-1} \mod p.$$

The value a is kept secret, as the random number k, and the value p, α, β are public.

2.2.2 Some attacks on the discrete logarithm cryptosystems

We will describe the main ideas of some known attacks against the discrete logarithm cryptosystems. All of them have on purpose to compute the discrete logarithm of an element of $\mathbb{Z}_p^*$ to a given base.

Schank's algorithm:baby-step, giant-step method

This method is based on a precomputation of some couple of the form (i, α^i), for $0 \leq i < m$. m is the value $\sqrt{p-1}$. The complete algorithm is given in figure 3. All operations are done modulo p. If we can ob-

Compute (i, α^i), for $0 \leq i < m$
Sort them by second component to obtain a list of pairs L
Compute $\beta\alpha^{-jm}$ for $0 \leq j < m$ and search if it is in the list L
If for one i and one j, we have $\beta\alpha^{-jm} = \alpha^i$, output $i + jm$.

Figure 3: The Baby-step Giant-step algorithm

tain a couple (i, j) for which the equality $\beta\alpha^{-jm} = \alpha^i$ is verified, then $\beta = \alpha^{i+jm}$ and then $\log_\alpha \beta = i + jm$.
This algorithm runs in time $O(m \log m)$ and is said to be a "square-root" method.

Pollig-Hellman algorithm

This method uses the factorisation of the order of the multiplicative group:

$$p - 1 = \prod_{i=1}^{t} p_i^{\lambda_i}.$$

If we can compute each values $a \mod p^{\lambda_i}$, where $a = \log_\alpha \beta$, then it is possible to compute $a \mod (p-1)$ by the Chinese remainder theorem.
The first step is to calculate $z_i \equiv a \mod p^{\lambda_i}$ for $0 \leq i \leq t$. We can write:

$$z_i = \sum_{j=0}^{\lambda_i - 1} z_{(i,j)} p_i^j.$$

where $0 \leq z_{(i,j)} \leq p_i - 1$. If $\gamma_i = \alpha^{p-1/p_i}$ is a p_i^{th} root of the unity in the group, we have:

$$\beta^{p-1/p_i} = \alpha^{a(p-1)/p_i} = \gamma_i^a = \gamma_i^{z(i,0)}.$$

It is possible to find the value of $z_{(i,0)}$ using a square-root method like the preceeding one.
If $\lambda_i > 1$, it is possible to find consecutively the $z_{(i,j)}$ by applying the same method with $(\beta\alpha^{-z_{(i,0)}})^{(p-1)/p_i^2}$, $(\beta\alpha^{-z_{(i,0)}-z_{(i,1)}p_i})^{(p-1)/p_i^3}$ ...
The complete algorithm is found in figure 4.
The factorisation of the order $p - 1$ is supposed to be known. If it is

For $i = 1$ to t
    Compute $z_i \equiv a \mod p^{\lambda_i}$ for $0 \leq i \leq t$
    Compute γ_i a p_i^{th} root of unity
    Let $\delta \leftarrow \beta$
    For $j = 0$ to $\lambda_i - 1$
        Compute $\gamma^{(p-1)/p_i^{j+1}}$ and find its logarithm to the base γ modulo p_i
        Let $z_{(i,j)} \leftarrow$ the result of this computation
        Let $\delta \leftarrow \delta\alpha^{-z_{(i,j)}p_i^j}$
    Let $z_i = \sum_{j=0}^{\lambda_i - 1} z_{(i,j)}p_i^j$
Using the chinese remainder theorem, compute a.

Figure 4: The Pollig-Hellman algorithm

not the case, a factorisation algorithm can be used, but it can increase the running time. If it is known, the running time is $O(\sum_{i=1}^t \lambda_i(\log(p-1) + \sqrt{p_i}\log p_i))$.

The index-calculus method

It is one of the most efficient method to compute the discrete logarithm is $\mathbb{Z}_p{}^*$. Adleman gave a complete study of it in 1979. The main idea is to use a factor base. It consists of a set of elements of the group, here in $\mathbb{Z}_p{}^*$ we could take all the prime numbers less than a chosen value m. Once the set is determined, we have to find all the logarithms of the elements of the set by writing a sufficient number of powers of α to obtain a linear congruence that can give them. Thereafter it is easy to compute the logarithms of all the elements that can be written in the factor base.
The complete algorithm with formal notation can be seen in figure 5. This algorithm runs in subexponential time (see 1.3).
The choice of the factor base is crucial: it must be large enough to be representative of a majority of the elements in the group, but it must be small enough to limitate the size of the congruence sytem.

Let $S = \{p_1, p_2, \ldots, p_t\}$ the factor base
Until you have at least t independent congruences or you have tried all the possibilities
1. Choose a random integer $0 \leq a < p-1$ no chosen yet
2. Try to write α^a in the factor base: $\alpha^a = \prod_{i=1}^{t} p_i^{\lambda_i}$
3. If it is not possible return to step 1.
   Else store the linear congruence $a \equiv \sum_{i=1}^{t} \lambda_i \log_\alpha p_i \mod (p-1)$

If you have not a sufficient number of congruences $\rightarrow$ failure
Else resolve the congruence system and compute the $\log_\alpha p_i$ for $1 \leq i \leq t$
Until you succeed or you have tried all the possibilities
  Choose a random $0 \leq s < p-1$
  Try to write $\alpha^s\beta$ in the factor base: $\alpha^s\beta = \prod_{i=1}^{t} p_i^{b_i}$
If you have not found a convenient $s \rightarrow$ failure
Else compute $\log_\alpha \beta \equiv \sum_{i=1}^{t} b_i \log_\alpha p_i - s \mod (p-1)$

Figure 5: The Index Calculus Method

2.2.3 Generalisation to finite fields

The discrete logarithm problem is quite the same in a finite field F_q.
The algorithms described in 2.2.2 can be used in finite fields too.
In fact the discrete logarithm problem can be formalized in every finite cyclic group G. Let α be a generator of the group G. Each element β of G can be written as a power of α: $\beta = \alpha^x$. So we call x the discrete logarithm of β to the base α.
The El-Gamal cryptosystem is directly adapted from the case in $\mathbb{Z}_p{}^*$:
Let $\sharp G$ be the order of the group.
We define $\mathcal{K} = \{(\alpha, a, \beta) \backslash \beta \equiv \alpha^a \mod \sharp G\}$.
For a given key $k = (\alpha, a, \beta)$ and a random number $k \in G$, the encryption function is:

$$e_K(x, k) = (y_1, y_2)$$

where

$$\begin{aligned} y_1 &= \alpha^k \mod \sharp G \\ y_2 &= x\beta^k \mod \sharp G \end{aligned}$$

and the decryption function is:

$$d_K(y_1, y_2) = y_2 (y_1^a)^{-1} \mod \sharp G.$$

The value a is kept secret, as the random number k, and the value $\sharp G$, α, β are public as well as the group G.
In cryptography, the cyclic groups used are often finite fields F_{p^m}.
The Baby-step Giant-step and the Pollig-Hellman algorithms can be directly adapted by changing "$p-1$" in "$\sharp G$". The Index-Calculus method is a little more delicate. It is convenient to represent the elements of F_{p^m} as the polynomials of degree less than m and coefficients in F_p. The operation are done modulo a fixed irreducible polynomial of degree m. The factor base can be the set of all irreducible polynomials of degree at most a fixed value.
A few words about general cyclic groups: every cyclic group of order n is isomorphic to the finite ring $\mathbb{Z}_n$, in which it is easily possible to compute the discrete logarithm with the preceeding algorithm (using the factorisation of n and the Chinese remainder theorem). The problem is that the isomorphism is not always easy to compute.

2.2.4 Elliptic curves

As described in the case of finite groups, it is possible to adapt the El-Gamal cryptosystem to the group of rational points of an elliptic curve. All the mathematic and cryptographic aspects of it are studied in 3. We already met elliptic curves about factorisation. In addition to factorisation and secure cryptosystem, elliptic curve are used to obtain error-correcting codes.

2.2.5 El-Gamal signature scheme and DSS

The most known signature scheme using a public-key is the El-Gamal signature scheme (1985). It led to one of the most used standards concerning signature schemes, the DSS (Digital Signature Standard). El-Gamal signature scheme and DSS are based on the discrete logarithm problem. The El-Gamal signature scheme is the following: (we describe it in $\mathbb{Z}_p$ but it can be adapted to finite fields)
The messages are in $\mathbb{Z}_p^*$ and the signature are in $\mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$.
Let $\mathcal{K} = \{(p,\alpha,a,\beta) \backslash \beta \equiv \alpha^a \mod p\}$.
For a given key $K = (p,\alpha,a,\beta)$, and for a (secret) random number $k \in \mathbb{Z}_{p-1}^*$, the signature function is:

$$sig_K(x,k) = (\gamma,\delta)$$

where

$$\begin{aligned} \gamma &\equiv \alpha^k \mod p \\ \delta &\equiv (x - a\gamma)k^{-1} \mod (p-1) \end{aligned}$$

and the verification function is:

$$ver(x,\gamma,\delta) = \text{ true } \Leftrightarrow \beta^{\gamma}\gamma^{\delta} \equiv \alpha^{x} \mod p.$$

The values p, α and β are public and the values a and k are kept secret.
The DSS is more recent (1994). The modifications that have been done to the El-Gamal signature scheme give a long term reliability to this signature scheme. Indeed when you have to send a message, it has to be sent once. But when you want to sign a message, you want that the signature stays verifiable a certain time. The DSS allows to take longer moduli, because they comparatively need shorter signature length than El-Gamal signature scheme (in which the signature length is the double of the modulus length). The main change is in the definition of δ, that is to say:

$$\delta \equiv (x + \alpha\gamma)k^{-1} \mod (p-1)$$

If $gcd(x + \alpha\gamma, p-1) = 1$, then $\delta^{-1} \mod (p-1)$ exists and the verification condition can be modified as follows:

$$\alpha^{x\delta^{-1}}\beta^{\gamma\delta^{-1}} \equiv \gamma \mod p.$$

The complete description of the DSS stands here: Let p be a 512-bit prime and q be a 160-bit prime that divides $p-1$. Let α be a q^{th} root of unity modulo p.
The messages are in $\mathbb{Z}_p^*$ and the signature are in $\mathbb{Z}_q \times \mathbb{Z}_q$.
Let $\mathcal{K} = \{(p,q,\alpha,a,\beta)\backslash\beta \equiv \alpha^a \mod p\}$.
For a given key $K = (p,q,\alpha,a,\beta)$, and for a (secret) random number $1 \leq k \leq q-1$, the signature function is:

$$sig_K(x,k) = (\gamma,\delta)$$

where

$$\begin{aligned} \gamma &\equiv (\alpha^k \mod p) \mod q \\ \delta &\equiv (x + a\gamma)k^{-1} \mod q \end{aligned}$$

and the verification function is:

$$ver(x,\gamma,\delta) = \text{ true } \Leftrightarrow (\alpha^{e_1}\beta^{e_2} \mod p) \equiv \gamma \mod q.$$

where

$$\begin{aligned} e_1 &\equiv x\delta^{-1} \mod q \\ e_2 &\equiv \gamma\delta^{-1} \mod q. \end{aligned}$$

The values p, q, α and β are public and the values a and k are kept secret.
The signature length is here 320 bits for a 512-bit modulo. One drawback of this method is that the verification algorithm is computed slower than the signature algorithm. Both are fortunately relatively fast to compute.

2.3 The knapsack cryptosystems

Knapsack cryptosystems are linked with the subset sum problem: You have a vector of integers $\mathbf{a} = (a_1, \ldots, a_n)$ and a target value K and you want to find, if it is possible, a 0-1 vector $\mathbf{x}$ of length n for which: $\mathbf{a} \cdot \mathbf{x} = K$. It is a NP-complete problem, it means it can be done in non-deterministic polynomial time.
The principle of a knapsack-based cryptosystem is the following:
Bob's public key is the vector $\mathbf{a}$. Alice send to Bob the message M(of length n) encrypted by sending him the value $K = \mathbf{a} \cdot M$. Oscar must resolve a knapsack problem if he wants to get M. But Bob must have a trapdoor. the problem is that when you choose particular forms of vectors $\mathbf{a}$, in order to have a one-to-one encryption, it is not NP-complete anymore. In fact, knapsack cryptosystems have almost all be broken, except the Chor-Rivest. The McEliece system was broken in 1997 (see [2]).

Similarly, we can imagine that a given NP-complete problem can generate a cryptosystem. In fact it is difficult to find such problem with a trapdoor, because a trapdoor would reduce the running time perhaps down to polynomial time, which would mean that P=NP. However, we don't know that.

2.4 Is it all?

Of course public key cryptography is not reduced to cryptosystems and signature schemes, and there are a lot of goals in cryptography:

- Secure message exchange: this is the role of **cryptosystems**
- Convincing everyone that a message you have received is from the good sender: **signature scheme**
- Convincing yourself of the identity of the sender of a message you received: **authentication scheme**
- Collaborate with more than two persons whatever the number of "means" is: **multi-party protocols**
- Convince someone of something without giving him additional informations: **zero-knowledge protocols**

Some of the tools used are described in B.

3 Elliptic curves and cryptography

After some words about what is an elliptic curve and what it is necessary to know about them to understand the EC cryptosystem, we will go further in the algorithmic problems and the generating of efficient curves. However, an interesting and complete study of elliptic curves can be found in [17].

3.1 Generalities about elliptic curves

The term elliptic comes from the research of the perimeter of ellipses, which has led to find such curves.

3.1.1 Definition. Weierstrass equations

Elliptic curves are defined on a field. In our case, this field will be a finite field F_q (where q is a power of a prime number).
The best known form for a definition of elliptic curves is the Weierstrass equations.

Definition 3 Weierstrass equation
Let K be a field, and $\overline{K}$ its algebraic closure.
A Weierstrass equation is a homogeneous equation of degree 3 of the form:

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$$

where the a_i coefficients are in $\overline{K}$.
The equation is said to be smooth if it has no singular point, and singular otherwise.

In the case of $K = F_q$, the algebraic closure $\overline{K}$ is $\cup_{m\geq 1} F_{q^m}$.
Then the definition of an elliptic curve follows:

Definition 4 Elliptic curve
Let K be a field and $P^2(\overline{K})$ the projective plan over K. An elliptic curve E is the set of solutions in $P^2(K)$ of a smooth Weierstrass equation.
The point at infinity is the point $\mathcal{O} = (0, 1, 0) \in P^2(\overline{K})$. It plays a special role.
For simplicity reason, it is useful to transform the Weierstrass equation with the change of variables:

$$x = \frac{X}{Z},\ y = \frac{Y}{Z}$$

then we obtain the following equation:

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

and the elliptic curve is the set of solutions of this equation in $\overline{K} \times \overline{K}$, with the point at infinity $\mathcal{O}$.

If all the coefficients a_i are in K, then the curve is said to be "over K".

3.1.2 Rational points and invariants. Normal forms

Some useful notions must be known about elliptic curves, such as rational points and isomorphism invariants:

Definition 5 Rational points
Let E be an elliptic curve over the field K. A rational point is a point of E whose coordinates are both in K, or the point at infinity. The set of rational points is written $E(K)$.

Definition 6 Isomorphism between two elliptic curves
Two elliptic curves E_1 and E_2 are isomorphic if they are isomorphic as projective varieties: there exist two morphisms over K, $\Phi : E_1 \to E_2$ and $\Psi : E_2 \to E_1$, such that $\Phi \circ \Psi$ is the identity map on E_2 and $\Psi \circ \Phi$ is the identity map on E_1.

The interesting thing for cryptographic use in this notion is that in certain cases you would prefer an isomorphic curve with more convenient coefficients to compute operations. The following theorem offers a more practical criterion of isomorphism:

Theorem 1 *Two elliptic curves E_1 and E_2 defined over K are isomorphic if and only if there exist $u, r, s, t \in K$ with $u \neq 0$ such that the change of variables*

$$(x, y) \mapsto (u^2x + r, u^3y + u^2sx + t)$$

transforms the equation of E_1 into the equation of E_2. This is an equivalence relation.

In practice, this is equivalent to say that if the respective equations of E_1 and E_2 are:

$$\begin{aligned} y^2 + a_1xy + a_3y &= x^3 + a_2x^2 + a_4x + a_6 \\ y^2 + b_1xy + b_3y &= x^3 + b_2x^2 + b_4x + b_6 \end{aligned}$$

then the coefficients satisfy the relations:

$$\begin{aligned}
ub_1 &= a_1 + 2s \\
u^2 b_2 &= a_2 - sa_1 + 3r - s^2 \\
u^3 b_3 &= a_3 + ra_1 + 2t \\
u^4 b_4 &= a_4 - sa_3 + 2ra_2 - (t+rs)a_1 + 3r^2 - 2st \\
u^6 b_6 &= a_6 + ra_4 + r^2 a_2 + r^3 - ta_3 - t^2 - rta_1
\end{aligned}$$

Two quantities are useful when you study elliptic curves: the discriminant and the j-invariant. To define them, we need to define the following quantities:

$$\begin{aligned}
d_2 &= a_1^2 + 4a_2 \\
d_4 &= 2a_4 + a_1 a_3 \\
d_6 &= a_3^2 + 4a_6 \\
d_8 &= a_1^2 a_6 + 4a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2 \\
c_4 &= d_2^2 - 24 d_4
\end{aligned}$$

Definition 7 Discriminant
$\Delta = -d_2^2 d_8 - 8d_4^3 - 27d_6^2 + 9d_2 d_4 d_6$

This quantity is characteristic of the singularity of the Weierstrass equation: the Weierstrass equation is non-singular if and only if $\Delta \neq 0$.

Definition 8 j-invariant
If $\Delta \neq 0$, we can define $j(E) = \frac{c_4^3}{\Delta}$.

The j-invariant is characteristic of the class of equivalence for isomorphism: if two curves are isomorphic, they have the same j-invariant. If K is algebraically closed, the converse is true.

The equation of the curve and the expression of the discriminant and the j-invariant can be simplified in finite fields, using some properties of their characteristic. The figure 6 shows those different expressions.

3.1.3 The group law. Addition in E(K)

The points of an elliptic curve form an Abelian group for a special group law that we will note $\oplus$.

Definition 9 Group law (geometrically)
Let P and Q be two points on an elliptic curve E. Let D be the line:

Char.	j	E	Δ	j
$\neq 2,3$		$y^2 = x^3 + a_4x + a_6$	$-16(4a_4^3 + 27a_6^2)$	$1728\frac{4a_4^3}{4a_4^3+27a_6^2}$
3	$\neq 0$	$y^2 = x^3 + a_2x^2 + a_6$	$-a_2^3a_6$	$-\frac{a_2^3}{a_6}$
3	$= 0$	$y^2 = x^3 + a_4x + a_6$	$-a_4^3$	0
2	$\neq 0$	$y^2 + xy = x^3 + a_2x^2 + a_6$	a_6	$\frac{1}{a_6}$
2	$= 0$	$y^2 + a_3y = x^3 + a_4x + a_6$	a_4^3	0

Figure 6: Expression of the Weierstrass equation depending on the field K

- *connecting P and Q if $P \neq Q$*
- *tangent to E at P if $P = Q$*

Let R be the third point of intersection of D with E. If $\mathcal{O}$ is the point at infinity, let D' the line connecting R and $\mathcal{O}$. Then the point $P \oplus Q$ is the third point of intersection of D' with E.

See figure 7 for examples.

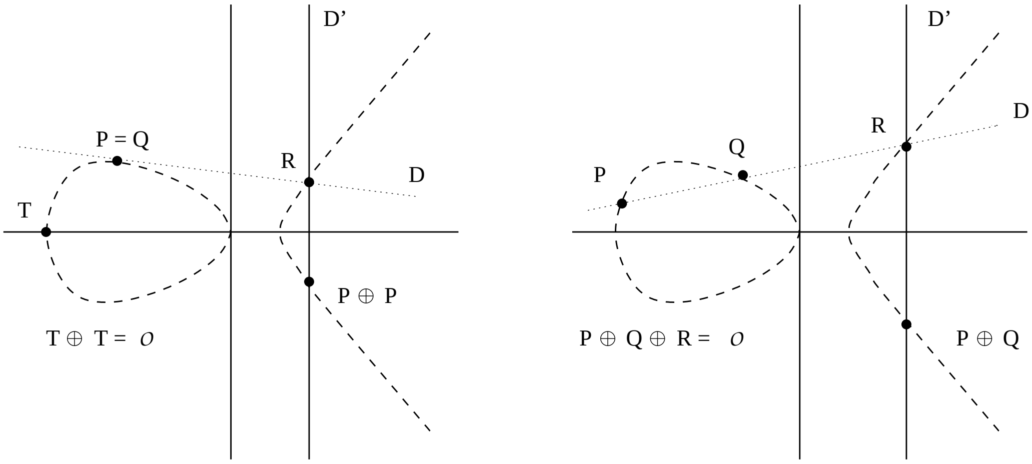


Figure 7: Geometrical definition of the sum

Theorem 2
E is an Abelian group for the law $\oplus$

Proof: All is trivial with the geometrical definition, except the associativity. See Annexe A.2.1. The addition law has the following properties:

Proposition 1 Composition law properties
For an elliptic curve given by the Weierstrass equation $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$,

1. *$\mathcal{O}$ is the zero of the law*
2. *If $P = (x_1, y_1) \neq 0$, then $-P = (x_1, -y_1 - a_1x_1 - a_3)$*

3. *If* $P = (x_1, y_1)$*,* $Q = (x_2, y_2)$ *and* $Q \neq -P$*, let*

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{if } P \neq Q \\ \\ \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}, & \text{if } P = Q \end{cases}$$

If $P \oplus Q = (x_3, y_3)$*, then*

$$\begin{aligned} x_3 &= \lambda^2 + a_1 - x_1 - x_2 \\ y_3 &= -(\lambda + a_1)x_3 - (y_1 - \lambda x_1) - a_3 \end{aligned}$$

4. *If E is defined over K, E(K) is a subgroup of E.*

3.1.4 Super-singularity

The group E(K) of an elliptic curve is a very interesting object and it is necessary to know some of its properties in order to find the best elliptic curve cryptosystems. One of the most important results is Hasse's theorem, conjectured by Artin, bounding the order of the group of an elliptic curve defined over the finite field F_q.

Theorem 3 (Hasse)
Let E be an elliptic curve defined over the field F_q*. Then:*

$$|\sharp E(K) - q - 1| \leq 2\sqrt{q}$$

An elegant proof (from [17]), is given in Annexe A.2.2.
We can also say that the group order is of the form $q + 1 - t$ with $|t| \leq 2\sqrt{q}$.

Proposition 2 *If* $q = p^m$ *then*

1. $t \not\equiv 0 \mod p$
2. *if m is odd then one of the following holds:*
   - $t = 0$
   - $t^2 = 2q$ *and* $p = 2$
   - $t^2 = 3q$ *and* $p = 3$
3. *if m is even then one of the following holds:*
   - $t^2 = 4q$
   - $t^2 = q$ *and* $p \not\equiv 1 \mod 3$

- $t = 0$ *and* $p \not\equiv 1 \mod 4$

We can define now the notion of super-singularity:

Definition 10 Super-singularity
An elliptic curve E defined over K is said to be super-singular if p divides t, where $\sharp E(F_q) = q + 1 - t$. *Otherwise E is said to be non-super-singular.*

Proposition 3 *Let E be an elliptic curve defined over* F_q *and* $\sharp E(F_q) = q + 1 - t$. *E is super-singular if and only if* $t^2 = 0, q, 2q, 3q,$ *or* $4q$.

When $p = 2$ or $p = 3$, E is super-singular if and only if $j(E) = 0$.
We will see further that it is more secure to choose non-super-singular curves.

3.1.5 Structure of the group

The structure of an Abelian finite group G is relative to its unique decomposition of the form:

$$G = \mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2} \oplus \ldots \oplus \mathbb{Z}_{n_k}.$$

where $\forall i = 1 \ldots k-1$, $n_{i+1} \mid n_i$.
G is said of rank k and of type $(n_1, \ldots, n_k)$.
It is possible to know with a good precision the type of the group E(K). The first result give a good idea of it:

Theorem 4
$E(F_q)$ is an Abelian group of rank 1 or 2. In the second case, its type is (n_1, n_2) with $n_2 \mid q - 1$.

If E is a super-singular curve, the group structure is known:

Theorem 5 Type of the group of a super-singular curve
Let $\sharp E(F_q) = q + 1 - t$ *where* $t^2 \leq 4q$.

1. *If* $t^2 = q,\ 2q,\ 3q$, *then* $E(F_q)$ *is cyclic*
2. *If* $t^2 = 4q$, *then:*
   - *if* $t = 2\sqrt{q}$, *then* $E(F_q) \cong \mathbb{Z}_{\sqrt{q}-1} \oplus \mathbb{Z}_{\sqrt{q}-1}$
   - *if* $t = -2\sqrt{q}$, *then* $E(F_q) \cong \mathbb{Z}_{\sqrt{q}+1} \oplus \mathbb{Z}_{\sqrt{q}+1}$
3. *if* $t = 0$, *then*
   - *if* $q \not\equiv 3 \mod 4$ *then* $E(F_q)$ *is cyclic*

- *if* $q \equiv 3 \mod 4$ *then* $E(F_q)$ *is cyclic or* $E(F_q) \cong \mathbb{Z}_{(q+1)/2} \oplus \mathbb{Z}_2$.

In the case of the non-super-singular curves, we use the following result:

Proposition 4
Let $\sharp E(F_q) = q + 1 - t = N$ *where* $t^2 \leq 4q$ *and* $t \not\equiv 0 \mod p$.
If a_l *and* b_l *are integers satisfying for each prime* $l \neq p$

- $a_l \geq b_l$,
- $a_l + b_l = v_l(N)$, *where* $v_l(N)$ *is the l-adic valuation of N, i.e. the greatest exponent* n *such as* $l^n \mid N$,
- $b_l \leq v_l(q-1)$,

then there exists a non-super-singular curve E defined over F_q *such that* $E(F_q)$ *has group structure:*

$$\mathbb{Z}_{p^{v_p(N)}} \oplus \bigoplus_{l \neq p} (\mathbb{Z}_{l^{a_l}} \oplus \mathbb{Z}_{l^{b_l}}).$$

3.2 The discrete logarithm problem in elliptic curves case

As described above, it is possible to formalise the discrete logarithm problem in the group of rational points E(K) of an elliptic curve E, or in a cyclic subgroup of this group.
The discrete logarithm problem becomes:
Let $P \in E(F_q)$ be a point of order n, and let $Q \in E(F_q)$ another point. The value of n is known. The problem is to determine the integer l such as $R = lP$ if such an integer does exist.
The El-Gamal cryptosystem becomes:
We define $\mathcal{K} = \{(P, l, R) \backslash R = lP\}$.
For a given key $k = (P, l, R)$ and a random number $k \in \mathbb{Z}_n$, the encryption function is:

$$e_K(x, k) = (y_1, y_2)$$

where

$$\begin{aligned} y_1 &= kP \\ y_2 &= kR + X \mod \sharp G \end{aligned}$$

and the decryption function is:

$$d_K(y_1, y_2) = y_2 - ly_1 \mod \sharp G.$$

The value l is kept secret, as the random number k, and the value n, P, R are public as well as the curve E.
Some methods exists to transform the plaintext into a point of the curve X, but the problem with the El-Gamal cryptosystem is that the message length can be multiplied by four, while it is only a factor two in F_q.
So the mostly used cryptosystem using elliptic curve is a variant found by Menezes and Vanstone. In this cryptosystem, the plaintexts and the ciphertexts are not points of the curve but pairs of elements of the field.
It works as follows:
Let E be an elliptic curve defined over F_q such that E contains a cyclic subgroup G in which the discrete logarithm problem is supposed to be intractable. The plaintexts are in $F_q^* \times F_q^*$ and the ciphertexts in $E \times F_q^* \times F_q^*$. We suppose that $q = p^m$.
We define $\mathcal{K} = \{(P, l, R) \backslash R = lP\}$ where $P \in E$.
For a given key $K = (P, l, R)$, a plaintext $X = (x_1, x_2)$ and a random number $k \in \mathbb{Z}_{|G|}$, the encryption function is:

$$e_K(X, k) = (y_0, y_1, y_2)$$

where

$$\begin{aligned} y_0 &= kP \\ (c_1, c_2) &= kR \\ y_1 &= c_1 x_1 \mod p \\ y_2 &= c_2 x_2 \mod p \end{aligned}$$

and the decryption function is:

$$d_K(y_0, y_1, y_2) = (y_1 c_1^{-1} \mod p, y_2 c_2^{-1} \mod p)$$

where $ly_0 = (c_1, c_2)$. The value l is kept secret, as the random number k, and the values n, P, R are public as well as the curve E.

3.3 Choice of the elliptic curve for secure cryptosystem

Several problems occur:

- In certain cases, the isomorphism between the group and a finite field is easy to compute. More precisely, it occurs when the curve is super-singular. In fact, it is always possible to reduce the problem to a field, but in the case of a super-singular curve, the field keep a reasonable size, while in the case of non-super-singular curves, the field is large.

- The order of the group must be chosen such as to avoid some attacks based on a factorisation like the Pollig-Hellman method and to make the group operation easier.

- The choice of the field and the choice of the representation of its elements are important in order to make the group operations easier.

3.3.1 Super-singularity: Menezes-Okamoto-Vanstone results

Before 1991, mathematicians knew only exponential algorithms to compute the discrete logarithm in the case of elliptic curves. Menezes, Okamoto and Vanstone found a way to reduce elliptic curve logarithms to logarithms in an extension of the finite field on which the curve is defined, for a certain class of curves (see [11]). More precisely, the result they proved is the following:

Theorem 6
If E is a super-singular elliptic curve defined over F_q it is possible to reduce the elliptic curve logarithm problem in E to the discrete logarithm problem in an extension field F_{q^k} in probabilistic polynomial time.

It means that the elliptic logarithm problem in this case runs in probabilistic sub-exponential time. The cryptographic implications of this result are easy to deduce: The key length in a cryptosystem using a super-singular elliptic curve must be much larger than in a cryptosystem using a non-super-singular curve to provide the same security.
In recent elliptic curve cryptosystems, the super-singular curves are avoided as much as possible.
For an idea of the proof, see A.2.3.

3.3.2 Order of the group and classical attacks

We saw with the Pollig-Hellman attack that discrete logarithms can be quite fast be computed if the order of the group has small prime factors (ie. if it is easy to factorise). So the order of the group must have large prime factors to provide a secure cryptosystem. There are algorithm that allow to count the number of point of a given elliptic curve. The Schoof algorithm (1985) gives the number of points in $E(K)$ in polynomial time depending of the size of the field. It has been improved by Atkin and Elkies. It needs some notions of modular calculus. A good sum up and synthesis of these methods can be found in [14].
Another approach is to compute a curve, with the good properties (non-super-singularity for example) and the good number of point directly. The

current trend is to use this approach.

The most effective attack on the discrete logarithm problem in a finite field is the Index Calculus method given in 2.2.2. We could be tempted by trying to adapt it to elliptic curves. But this is a impossible goal, for several reasons. A detailed survey can be found in [18]. To sum up, it is difficult to construct a convenient set S for a factor base, and furthermore it is difficult to write elements of the rational points group as "products" of a factor base.

If the order of the group is the prime p and if the curve is defined over the field F_{p^m}, the curve is said to be anomalous. In this case, it is possible to compute the discrete logarithm problem in polynomial time. It is of course the same if we consider a subgroup of $E(F_{p^m})$ of order p. A proof can be found in [16] and another in [20]. Those curves have a good property that allow to avoid square-root attacks, but unfortunately they are very vulnerable to a specific attack.

3.3.3 Operations in the group: the choice of the field

Two sorts of fields are usually chosen to be support for elliptic curve cryptosystems, because the operations can be very efficiently computed in them:

Fields of characteristic 2: F_{2^m}

In characteristic two, the Weierstrass equation of an non-singular elliptic curve can be put on the form:

$$y^2 + xy = x^3 + a_2x^2 + a_6.$$

Once a basis is chosen, the elements of such a field can be represented by 0-1 vectors of length m. The addition of two elements is then a XOR operation, A squaring is just a rotation of the vector. It has really practical interests concerning the hardware implementation. The formulae of addition and doubling of two points become, for some points $P = (x_1, y_1)$, $Q = (x_2, y_2) \neq -P$ and $R = P \oplus Q = (x_3, y_3)$:

$$-P = (x_1, x_1 + y_1)$$

$$x_3 = \begin{cases} \left(\frac{y_1+y_2}{x_1+x_2}\right)^2 + \frac{y_1+y_2}{x_1+x_2} + x_1 + x_2 + a_2 & \text{if} \quad P \neq Q \\ x_1^2 + \frac{a_6}{x_1^2} & \text{if} \quad P = Q \end{cases}$$

$$y_3 = \begin{cases} \left(\frac{y_1+y_2}{x_1+x_2}\right)(x_1+x_3)+x_3+y_1 & \text{if} \quad P \neq Q \\ x_1^2 + \left(x_1 + \frac{y_1}{x_1}\right)x_3 + x_3 & \text{if} \quad P = Q \end{cases}$$

The computation of these operations takes a short time when one uses an optimal normal basis (see A.1).

Base fields F_p, $p > 3$

In characteristic p, the Weierstrass equation of an non-singular elliptic curve can be put on the form:

$$y^2 = x^3 + ax + b.$$

The addition formulae are the following (with the same notations as before):

$$-P = (x_1, -y_1)$$

$$x_3 = \begin{cases} \left(\frac{y_2-y_1}{x_2-x_1}\right)^2 - x_1 - x_2 & \text{if} \quad P \neq Q \\ \left(\frac{3x_1^2+a}{2y_1}\right)^2 - x_1 - x_2 & \text{if} \quad P = Q \end{cases}$$

$$y_3 = \begin{cases} \left(\frac{y_2-y_1}{x_2-x_1}\right)(x_1-x_3) - y_1 & \text{if} \quad P \neq Q \\ \left(\frac{3x_1^2+a}{2y_1}\right)(x_1-x_3) - y_1 & \text{if} \quad P = Q \end{cases}$$

3.3.4 Complex multiplication

We need some definition about properties of integers:

Definition 11 Smooth integer
An integer k is said to be smooth if, given a trial division bound l_{max}, every prime divisor of k is at most l_{max}.

Definition 12 Nearly-prime integer
An integer n is said to be nearly-prime if, given two large positive integers $r_{min} \leq r_{max}$, n is of the form $n = kr$ with r prime such that $r_{min} \leq r \leq r_{max}$ and k a smooth integer.

If E is a non-super-singular elliptic curve over F_q, then if its order is m, the integer

$$Z = 4q - (q+1-m)^2$$

is positive according to Hasse's theorem (Theorem 3). It is then possible to factorise Z into a unique decomposition

$$Z = DV^2$$

such that D is a square-free positive integer.
So we have, assuming that $m = q + 1 \pm W$,

$$4q = W^2 + DV^2$$

Definition 13 Complex multiplication
In such a case, E is said to have complex multiplication by $\sqrt{-D}$, and D is called a CM-discriminant for q.

The interest of this notion is that, knowing D, we can compute the order of the group of the elliptic curve via the formula above. It is possible to construct a curve with large near-prime order with small D.
To find D that are appropriate, we need to consider two aspects:

- The order of the curve must be nearly-prime
- The construction of a curve with this order must be possible

So we can define an appropriate D as a D for which there exists an elliptic curve over F_q with complex multiplication by $\sqrt{-D}$ and a nearly-prime order. In practice, the bound r_{min} and r_{max} are chosen as follows: $r_{max} = q+1+2\sqrt{q}$, $r_{min} = q+\sqrt{q}$ (in the case a prime order is wanted). The most the two bounds are near, the most difficult it is to find a convenient curve.
If we decide to chose only prime order, then the trial bound l_{max} should be 1. The probability to find a curve with a prime order given a value of D is of course less than to find a nearly-prime one. In practice, l_{max} is small, but not equal to 1. In the case of $q = p$, there is a necessary condition for D to be a convenient CM-discriminant:

Proposition 5
Let

$$K = \left\lfloor \frac{(\sqrt{p}+1)^2}{r_{min}} \right\rfloor .$$

A square-free positive integer D can be a CM-discriminant for p only if

$$\begin{cases} \textit{If} \quad p \equiv 3 \mod 8 & \textit{, then} \quad D \equiv 2,\ 3,\ \textit{or}\ 7 \mod 8 \\ \textit{If} \quad p \equiv 5 \mod 8 & \textit{, then} \quad D \textit{ is odd} \\ \textit{If} \quad p \equiv 7 \mod 8 & \textit{, then} \quad D \equiv 3,\ 6,\ \textit{or}\ 7 \mod 8 \\ \textit{If} \quad K = 1 & \textit{, then} \quad D \equiv 3 \mod 8 \\ \textit{If} \quad K = 2 \textit{ or } 3 & \textit{, then} \quad D \not\equiv 7 \mod 8. \end{cases}$$

We can see that it is not very long to verify if D is non-appropriate.

3.3.5 Class groups

This is the key-point to compute an elliptic curve with chosen order and CM-discriminant.

Definition 14 Class group
Let S be a symmetric 2×2 matrix of the form:

$$S = \begin{pmatrix} A & B \\ B & C \end{pmatrix}.$$

We will note such a matrix $[A, B, C]$. Let $\bowtie$ be the following equivalence relation on the symmetric matrices:

$$S \bowtie S' \Leftrightarrow \exists N \textit{ such that } \det N = 1,\ S = N^{-1} S' N.$$

The class group H(D) is the equivalence class of matrices S such that $\det S = D$. *It is an Abelian group and its order h(D) is called the class number.*

We can notice that each equivalence class contains exactly one matrix S such that:

1. $gcd(A, 2B, C) = 1$
2. $|2B| \leq A \leq C$
3. If either $A = |2B|$ or $A = C$ then $B \geq 0$.

This matrix is called the reduced matrix of the class.
The class group is needed to construct the reduced class polynomial $w_D(t)$, with integer coefficients and degree $h(D)$.
It can be defined as follows: if $[A_1, B_1, C_1], \ldots, [A_{h(D)}, B_{h(D)}, C_{h(D)}]$ are the matrices of the class group,

$$w_D(t) = \prod_{j=1}^{h(D)} (t - \mathcal{C}(A_j, B_j, C_j)),$$

where $\mathcal{C}(A_j, B_j, C_j)$ is the class invariant of the matrix $[A_j, B_j, C_j]$. The class invariant is quite long to define, but it is a function of A, B, C, D, K. The irreducible factors of the polynomial $w_D(t)$, when computed modulo p, lead to the coefficients of the wanted elliptic curve.

3.3.6 Conclusion: the perfect elliptic curve

- A curve over a field where the operations are quick: F_{2^m} or F_p.
- A non-singular and non-anomalous curve
- A curve order which is prime or nearly-prime
- A curve with complex multiplication
- A curve where the messages don't take to much place to be encoded into curve points

3.4 Construction of a secure elliptic curve cryptosystem

To sum up, the main steps of the construction of a secure elliptic curve cryptosystem are the following:
If the chosen field is of the type F_p:

1. Choose a large prime p and a small appropriate CM-discriminant D
2. Choose the order of the curve as a prime or a nearly-prime, such that the curve could not be super-singular or anomalous. $m = kr$
3. Compute the class group and the polynomial $w_D(t)$
4. Factorise $w_D(t)$ modulo p
5. Compute the coefficients of the Weierstrass equation of the curve
6. Find an element P of the curve of order r.

If the chosen field is of the type F_{2^m}:

1. Choose a large m for which there exists an optimal normal basis
2. Choose an optimal normal basis in F_{2^m}
3. Find a nearly-prime order and a CM-discriminant for a possible elliptic curve on a subfield F_{2^d} of F_{2^m} such that the curve could not be super-singular or anomalous
4. Extend to F_{2^m}. $m = kr$
5. Compute the class group and the polynomial $w_D(t)$

6. Factorise the polynomial $w_D(t)$ modulo 2
7. Compute the coefficients of the Weierstrass equation of the curve
8. Find an element P of the curve of order r.

Conclusion

During this training period, the main points concerning public-key cryptography were:

- Public-key cryptography is a very large area of both mathematics and computer science. It needs complete and deep knowledge of algebraic and arithmetic tools, and the implementation considerations are very important since the security is mostly based on computer security of algorithms.

- Public-key cryptography offers a lot of possibilities that are impossible to private-key cryptography, especially signature-scheme and similar authentication protocols. This aspect is very important at a period where the electronic transactions grow in volume, on Internet for example.

- Most of the public-key cryptosystems are based on highly difficult problems such as Integer Factorisation, Discrete Logarithm and Elliptic Curve Discrete Logarithm. The challenge is to find another "hard problem" that can improve the security. So cryptography is a motor that can wake up some mathematical areas.

- Among all those cryptosystems, one is particularly interesting because it is nearly erratic: elliptic curves. Indeed, if the curve is badly chosen, the ECDL can be a trivial problem, and if it is well chosen, it can be a problem for which not even a sub-exponential time algorithm is known. Elliptic curves seem to be one of the most studied tools for public-key cryptography. The generalisation to hyper-elliptic curves is a interesting research area.

A Mathematical bases

A.1 Bases in finite fields

We will work in a finite field F_{p^m}. There are several ways to represent the elements of this field such that it is easy to do operations in the field.
To represent the elements, it is convenient to choose a basis. A basis is a set of m elements which are linearly independent over the base field $\mathbb{Z}_p$. We can distinguish:

Polynomial bases :

They are of the form: $\{1, \alpha, \alpha^2, \ldots, \alpha^{m-1}\}$ where $\alpha \in F_{p^m}$ and each element x of the field is defined using a unique polynomial $P(X) \in \mathbb{Z}_p[X]$ of degree $< m$ such that $x = P(\alpha)$ (x is a linear combination in the base field of the elements of the basis).
The operations in the field are then usual polynomial operations, modulo a certain polynomial of degree m which is characteristic of the extension $F_{p^m}/\mathbb{Z}_p$.

Normal bases :

They are of the form: $\{\beta, \beta^p, \ldots, \beta^{p^{m-1}}\}$ where $\beta \in F_{p^m}$. β is said to be a normal element of the field. To do the multiplications in the field, we need a multiplication table associated with the basis. It is defined as follows: If we suppose that $\{\alpha_0, \alpha_1, \ldots, \alpha_{m-1}\}$ is a (ordered) normal basis, the multiplication table is the linear $m \times m$ matrix with coefficients in $\mathbb{Z}_p$ such that:

$$\alpha_0 \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{m-1} \end{pmatrix} = T \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{m-1} \end{pmatrix}$$

The number of zeros in the multiplication table is called the complexity of the table and is written C_T. We have $C_T \geq 2m - 1$ and if $C_T = 2m-1$, then the basis is called an optimal normal basis. Using a certain architecture and an optimal normal basis, it is possible to compute a multiplication in m clock cycles.

It is possible to convert a normal basis into a polynomial basis and the converse in polynomial time.

A.2 On elliptic curves

As said S.Lang, "It is possible to write endlessly on elliptic curves", but we will just present here some results evocated in the report.

A.2.1 The group law

We will prove here that the law defined in 3.1.3 is a group law.
More precisely, we will prove the following theorem:

Theorem 7
The composition law has the following properties:

1. *If a line D intersects E at the points P, Q, R, then $(P \oplus Q) \oplus R = \mathcal{O}$.*
2. $\forall P \in E,\ P \oplus \mathcal{O} = P$.
3. $\forall P, Q \in E,\ P \oplus Q = Q \oplus P$.
4. $\forall P \in E$, *there exists a point of E, called $-P$ such that $P \oplus (-P) = \mathcal{O}$.*
5. $\forall P, Q, R \in E,\ (P \oplus Q) \oplus R = P \oplus (Q \oplus R)$.
6. *If E is defined over K, then $E(K)$ is a subgroup of E.*

Proof :

1. It is obvious considering the geometrical definition of the law.
2. If we take $Q = \mathcal{O}$ in the definition, we see that lines D and D' coincide. D intersects E at P, R , $\mathcal{O}$ and D' at R, $P \oplus \mathcal{O}$, $\mathcal{O}$. So $P \oplus \mathcal{O} = P$.
3. The construction given in the definition is symmetric in P and Q, so the law is symmetric too.
4. The line through P and $\mathcal{O}$ also intersects E at R, so we have: $\mathcal{O} = (P \oplus \mathcal{O}) \oplus R = P \oplus R$.
5. It is the most difficult point to prove. It can be done using an awful calculus. We will give a geometrical proof (see [6]).

   Let $\phi : E \times E$ be the function defined by: If P, $Q \in E$ then $\phi(P, Q)$ is the third point at which the line connecting P and Q intersects the curve E. We can notice that $P \oplus Q = \phi(\mathcal{O}, \phi(P, Q))$.

Let P, Q, R be three points on the curve E. We want to prove that $(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$.
Let $S = P \oplus Q$, $U = Q \oplus R$, $T_1 = \phi(S, R)$ and $T_2 = \phi(P, U)$. It is sufficient to prove that $T_1 = T_2$, since $(P \oplus Q) \oplus R = \phi(\mathcal{O}, T_1)$ and $P \oplus (Q \oplus R) = \phi(\mathcal{O}, T_2)$.
For this purpose, we need the following lemma:

Lemma 1
Let C be an irreducible cubic and C' and C" two other cubics. If $C \cap C' = \{P_1, \ldots, P_9\}$, where the P_is are distinct points on C, and if $C \cap C'' = \{P_1, \ldots, P_8, Q\}$, then $Q = P_9$.

Proof :
Let D be a line through P_9 which doesn't pass through Q. $L \cap C = \{P_9, R, S\}$. Then $(L \cup C'') \cap C = (C \cap C') \cup \{Q, R, S\}$, so there is a line D' such that $D' \cap C = \{Q, R, S\}$. But then $D' = D$ and so $P_9 = Q$.

To finish the proof, we define six lines:

D_1	such that	$D_1 \cap E = \{P, Q, S'\}$
D'_1	such that	$D'_1 \cap E = \{\mathcal{O}, S', S\}$
D_2	such that	$D_2 \cap E = \{S, R, T_1\}$
D'_2	such that	$D'_2 \cap E = \{Q, R, U'\}$
D_3	such that	$D_3 \cap E = \{\mathcal{O}, U, U'\}$
D'_3	such that	$D'_3 \cap E = \{P, U, T_2\}$

Notice that we always know two of the three points in the intersection, that automatically defines both the line and the third point.
Now we define two cubics by linear systems:

$$C' = \begin{cases} D_1 \\ D_2 \\ D_3 \end{cases}$$

and

$$C'' = \begin{cases} D'_1 \\ D'_2 \\ D'_3 \end{cases}$$

where the name of the lines represents their equations.
By applying the lemma, we have $T_1 = T_2$.

6. If P and Q have coordinates in K, then the equation of the line connecting them has coefficients in K. If further E is defined over K, then the third point of intersection will have its coordinates in K, since they are given by rational combinations of the coefficients of the line and of E.

A.2.2 Proof of Hasse's theorem

We recall here theorem 3:

Theorem 8 (Hasse)
Let E be an elliptic curve defined over the field F_q. Then:

$$|\sharp E(K) - q - 1| \leq 2\sqrt{q}$$

We need to define some notions:

Definition 15
If E is a curve defined in $\mathbb{P}^2(\overline{K})$ by a Weierstrass equation, then $E^{(q)}$ is the curve defined by the same equation in which all coefficient are to the power of q.

Definition 16
If E is a curve defined over K, let:

$$K[E] = \frac{K[X]}{\{f \in K[X] \backslash \forall P \in E,\ f(P) = 0\}}$$

and K(E) its quotient field.

Definition 17
If $\Phi : E_1 \rightarrow E_2$ is a non constant map of curves defined over K. Composition with Φ induces an injection of function fields fixing K:

$$\begin{array}{rccc} \Phi^* : & K(E_2) & \rightarrow & K(E_2) \\ & \Phi^*(f) & = & f \circ \Phi. \end{array}$$

Definition 18 Degree of a map of curves
Let $\Phi : E_1 \rightarrow E_2$ be a map of curves defined over K. If Φ is constant, then the degree of Φ is 0. Otherwise Φ is said to be finite and its degree is:

$$\deg \Phi = [K(E_1) : \Phi^* K(E_2)].$$

Definition 19 Isogeny
Let E_1, E_2 be two elliptic curves. An isogeny between E_1 and E_2 is a morphism $\Phi : E_1 \to E_2$ such that $\Phi(\mathcal{O}) = \mathcal{O}$.

The following lemmas are needed:

Lemma 2
Let K be a field of characteristic $p > 0$, $q = p^m$ and E/K a curve. Let $\Phi : E \to E^{(q)}$ be the q^{th}-power Frobenius morphism. Then

1. $\Phi^* K(E^{(q)}) = K(E)^{(q)} = \{f^q \backslash f \in K(E)\}$
2. *Φ is purely inseparable*
3. $\deg \Phi = q$

Lemma 3
Let K be a field of characteristic $p > 0$, and E/K a curve. Let $\Phi : E \to E$ be the q^{th}-power Frobenius endomorphism. For $m, n \in \mathbb{Z}$, the map $m + n\Phi : E \to E$ is separable if and only if $p \not| m$.
In particular, $1 - \Phi$ is separable.

Lemma 4
Let $\Phi : E_1 \to E_2$ a non-constant isogeny. If Φ is separable, then Φ is unramified, it means $\sharp \ker \Phi = \deg \Phi$, and $\overline{K}(E_1)$ is a Galois extension of $\Phi^* K(E_2)$.

Lemma 5
Let E_1 and E_2 be elliptic curves. The degree map

$$\deg : Hom(E_1, E_2) \to \mathbb{Z}$$

is a positive definite quadratic form.

Lemma 6
Let A be an Abelian group and $d : A \to \mathbb{Z}$ a positive definite quadratic form. Then for all $\Phi, \Psi \in A$,

$$|d(\Psi - \Phi) - d(\Phi) - d(\Psi)| \leq 2\sqrt{d(\Phi)d(\Psi)}.$$

Proof :
For $\Phi, \Psi \in A$, let $L(\Psi, \Phi) = d(\Psi - \Phi) - d(\Phi) - d(\Psi)$.
L is a bilinear form and we have for all $m, n \in \mathbb{Z}$:

$$0 \leq d(m\Psi - n\Phi) = m^2 d(\Psi) + mnL(\Psi, \Phi) + n^2 d(\Phi).$$

If we choose $m = -L(\Psi, \Phi)$ and $n = 2d(\Psi)$, we obtain

$$0 \leq d(\Psi)[4d(\Psi)d(\Phi) - L(\Psi, \Phi)^2].$$

If $\Psi = 0$ the wanted inequality is trivial, else we have $L(\Psi, \Phi)^2 \leq 4d(\Psi)d(\Phi)$, and then

$$|d(\Psi - \Phi) - d(\Phi) - d(\Psi)| \leq 2\sqrt{d(\Phi)d(\Psi)}.$$

Proof of the theorem:
We suppose that a Weierstrass equation with coefficients in $K = F_q$ defines the curve E. Let

$$\begin{array}{rccc} \Phi : & E & \rightarrow & E \\ & (x, y) & \rightarrow & (x^q, y^q) \end{array}$$

be the q^{th}-power Frobenius morphism. The Galois group $Gal_{\overline{K}/K}$ is generated by the q^{th}-power map on $\overline{K}$. Thus, we have for a point $P \in E(\overline{K})$,

$$P \in E(K) \text{ if and only if } \Phi(P) = P.$$

It means that $E(K) = \ker(1 - \Phi)$, so $\sharp E(K) = \sharp \ker(1 - \Phi)$.
By lemma 3, we know that $1 - \Phi$ is separable. Lemma 4 then gives the following equality:

$$\sharp E(K) = \deg(1 - \Phi).$$

Lemma 2 tells us that $\deg \Phi = q$ and lemma 5 that the degree map on $\mathrm{End}(E)$ is a positive definite quadratic form. So, applying the lemma 6, we obtain the wanted inequality:

$$|\sharp E(K) - q - 1| \leq 2\sqrt{q}.$$

A.2.3 Proof of the Menezes - Okamoto - Vanstone result

This is an idea of the original proof, which is given in [11].
The basis of the field F_q is known. It is possible to find a polynomial $f(x)$ of degree k (in order to compute in F_{q^k}) in probabilistic polynomial time.

Definition 20 ***n*-torsion points** *An n-torsion point P of an elliptic curve is a point satisfying $nP = \mathcal{O}$, where $n \neq 0$.*
The set of all n-torsion points over a curve E is written $E[n]$.

The notion of n-torsion point is important for the following. A few results about the set of n-torsion points:

Proposition 6
If the curve is based on the field F_q, where $q = p^m$, then:

- *If n and q are relatively prime, then* $E[n] \cong \mathbb{Z}_n \oplus \mathbb{Z}_n$
- *If* $n = p^l$ *then*
  - *if E is super-singular, then* $E[p^l] \cong \{\mathcal{O}\}$
  - *if E is non-super-singular, then* $E[p^l] \cong \mathbb{Z}_{p^l}$

We need the Weil-pairing function too:
The Weil-pairing function of order n is a function

$$e_n \ : \ E[n] \times E[n] \to \overline{F_q}$$

It has the following properties:

Proposition 7

- $\forall P \in E[n], \ e_n(P, P) = 1$
- *e_n is alternated, bilinear, non-degenerated*
- $\forall P_1, \ P_2 \in E[n]$, *if* $E[n] \subseteq E(F_{q^k})$, *then* $e_n(P_1, P_2) \in F_{q^k}$
- *If* $E[n] \subseteq E(F_q)$, *n and q are relatively prime and* $P \in E[n]$ *is a point of order n, then* $\forall P_1, \ P_2 \in E[n]$, *$P_1$ and P_2 are in the same coset of P in $E[n]$ if and only if* $e_n(P, P_1) = e_n(P, P_2)$

The last element we need is the following:

Proposition 8
Let μ_n the subgroup of the n^{th} roots of unity in F_{q^k}.

- *There exists* $Q \in E[n]$ *such that* $e_n(P, Q)$ *is a n^{th} root of unity.*
- *Let* $f \ : < P > \to \mu_n$ *be the function defined by* $f(R) = e_n(R, Q)$. *Then f is a group isomorphism.*

The figure 8 shows the different values of k depending on the values of t. We can notice that the value of k is less or equal than 6.
The complete algorithm of the reduction is given in figure 9. The point Q can be found in polynomial time. The elements α and β can be computed in probabilistic polynomial time as can be the Weil-pairing function of two given points. The discrete logarithm of β to the base α can be done in probabilistic sub-exponential time. So it is possible to compute the elliptic curve logarithm in probabilistic sub-exponential time in the case of a super-singular curve. All the details concerning the running times of the different steps are given in [11].

Class of curve	Value of t	Group structure	Value of k
I	0	cyclic	2
II	0	$\mathbb{Z}_{(q+1)/2} \oplus \mathbb{Z}_2$	2
III	$\pm\sqrt{q}$	cyclic	3
IV	$\pm\sqrt{2q}$	cyclic	4
V	$\pm\sqrt{3q}$	cyclic	6
VI	$\pm2\sqrt{q}$	$\mathbb{Z}_{\sqrt{q}\pm1} \oplus \mathbb{Z}_{\sqrt{q}\pm1}$	1

Figure 8: Classification of super-singular curves

Input: An element $P \in E(F_{q^k})$ of order n, and $R \in< P >$
Output: An integer l such that $R = lP$.
Determine the smallest integer k such that $E[n] \subseteq E(F_{q^k})$
Find $Q \in E[n]$ such that $\alpha = e_n(P, Q)$ has order n
Compute $\beta = e_n(R, Q)$
Compute l the discrete logarithm of β to the base α in $E(F_{q^k})$

Figure 9: Reduction algorithm

B To go further

B.1 Some words about Signature schemes

B.1.1 Special features

Some signature scheme have been adapted to satisfy special features. In a conventional signature scheme, the signer knows the contents of the message he has signed, the verification procedure is public and doesn't depend on the good-will of the signer. But some other goals can be performed:

Blind signature scheme : Bob can get a message signed by Alice without revealing her any information about the message. Those sort of signature schemes are used in time-stamping, anonymous access control and digital cash.

Designated confirmer signature scheme : A designated person can confirm the authenticity of a document without the signer's intervention. But without the participation of either the signer or the designated person, it is impossible to verify the authenticity of the document.

Fail-stop signature scheme : It is a signature scheme that protect from any forgery of someone's signature. If Oscar can forge Bob's signature, Bob can prove that forgery has been done: he solve a supposedly difficult problem, that Oscar cannot solve without knowing Bob's private key.

Group signature : Each member of a group can sign a document in such a manner that the verifier can see that it comes from the group, but not from which person in the group. There could be an authority in the group who can find the person who signed in case of contestation.

One-time signature : This signature scheme allows only one message to be signed with the same signature. They are based on one-way functions (it doesn't need to be a trapdoor one-way function). The principle is explained in B.1.2.

B.1.2 One-way signature schemes

A message is a 0-1 vector of length k. Each bit is signed individually, so the length of the signature is the same as the length of the message. Let X and S be two sets and $f : X \to S$ a one-way function. Let $y_{i,j} \in Y$ be chosen at random for $1 \leq i \leq k$ and $j = 0, 1$.
Let $z_{i,j} = f(y_{i,j})$ for $1 \leq i \leq k$ and $j = 0, 1$.The $y's$ are kept secret but the $z's$ are public.
The signature function is:

$$sig(x_1, \dots, x_k) = (y_{1,x_1}, \dots, y_{k,x_k})$$

and the verification procedure is:

$$ver(x_1, \dots, x_k, a_1, \dots, a_k) = \text{ true } \Leftrightarrow \forall i \in \{1 \dots k\}\ f(a_i) = z_{i,x_i}.$$

An evident drawback of this method is that the signature must be changed with a new message, and that the length of the signature is the same as the length of the message.

B.2 Hash functions: a way to more security

Definition 21 Hash function
A hash function is a transformation h with the following properties:

- *Its input can be of variable size*
- *It products a string with fixed size*

- *h is a one-way function*
- *h is as much as possible collision free*

Definition 22 Collision free
A function h is said to be weakly collision free if, given a message x, it is computationally infeasible to find a message $x' \neq x$ such that $h(x') = h(x)$.
It is said to be strongly collision free if it is computationally infeasible to find messages x and x' such that $h(x') = h(x)$.

B.2.1 Why is it better to use hash functions

Hash functions allow the message sender to sign a very long message although he uses a signature scheme with limited size messages. He just has to sign the image of the message by a hash function.
It is also useful to hide the contents of a message, in a time-stamping procedure for example. It is possible then to do operations on a message without being afraid of revealing this message.

B.2.2 Examples

MD2, MD4, MD5, SHA : Those hash functions are based on fixed size blocks manipulations, permutations and rotations. They are difficult to classify because they don't use a general scheme but a list of operations.

Block cipher : Those hash functions are based on a block cipher encryption like DES (Data encryption Standard). They are quite secure but less than the unclassified hash functions.

Modular arithmetic : Some hash functions are based on the discrete logarithm problem or the integer factorisation problem, but they don't seem to be secure enough.

B.3 Some other directions in cryptography

B.3.1 Private-key cryptography

Private-key cryptography needs a key-exchange, which can be done by a very secure channel, or by a key-exchange protocol like the Diffie-Hellman protocol:
Alice and Bob choose a group G and an element $\alpha \in G$. This operation is not secret.
Then Alice chose a random integer a , she computes α^a and send it to Bob.

Everyone can see α^a.
Bob choose also a random integer b and send α^b to Alice.
Alice computes $(\alpha^b)^a$ and Bob computes $(\alpha^a)^b$.
They have then the same key α^{ab}.
This procedure doesn't belong to private key cryptography, and before the development of public-key cryptography, it was difficult to find secure channel to exchange the key.

The encryption function and the decryption function depend on the same key, and they must be easy to apply knowing the key, but such that it is infeasible for Oscar to discover the plaintext or the key knowing the ciphertext.

The one-time pad

Definition 23 Unconditional security *A cryptosystem is said to be unconditionally secure when the eavesdropper Oscar can gain no knowledge about the plaintext except its length, even with infinite computer resources.*

The one-time pad consists of a XOR operation applied to the key and the plaintext. It is proved to be unconditionally secure. The great drawback of this system is that the key is as long as the plaintext.

The Data Encryption Standard

It is the most used private-key cryptosystem since 1977. It is very fast to implement, but the length of the key (56 bits) is not sufficient to be secure against attacks with recent computer resources. However it is still used because it is secure enough to protect from hackers with few money, it means only big societies or organisation can buy computers with special hardware that are studied to break this system.

Private-key cryptography cannot do all what public-key cryptography does. For example, the signature schemes are not possible, because Alice and Bob know the same key, so a message signed by Alice could as well have been signed by Bob.

B.3.2 Quantum cryptography

Quantum cryptography is a developing field. A quantum computer, if it can be possible to build one, would take polynomial time to do some exponential algorithm on a classical computer. So the cryptosystems based on computability difficulties wouldn't be secure anymore.
Another application of quantum mechanic in cryptography in the absolute secrecy during a data exchange: if an eavesdropped tries to see the message, he is obligatorily discovered because of the disturbance it has caused.
For general observation about quantum cryptography, see [4].

B.3.3 Geometric cryptography

It is based on a original idea of M.Burmester, R.L.Rivest and A.Shamir. The messages and the ciphertexts are represented by geometric quantities such as angles or segments. The computations are performed by ruler and compass constructions.
The impossibility of trisecting an angle in such a "computation model" gives a one-way function that is very efficient. This idea furnishes a secure identification or authentication protocol. Unfortunately, it is difficult to imagine a practical use of this idea.

B.4 Some recent works

A good way to know what is new about elliptic curves public-key cryptography is to see the improvements in implementation proposed in [7] and [15]. With some fields properties, and clever manipulations, it is possible to compute faster the operations needed in the construction and the use of the curves.

To develop the field of possibilities, some new "hard problems" are explored, such as the problem describe in [5]. This problem is to find short vectors in a given lattice:
Let $v_1, \ldots, v_n \in \mathbb{Z}^n$. Find $x_1, \ldots, x_n$ integers such that $\| \sum_{i=1}^{n} x_i v_i \|$ is minimised.
A one-way function called NF-EXP can then be constructed with the following properties:

- NF-EXP can be computed in polynomial time
- Inverting NF-EXP is at least as hard as factoring integers

- The only method currently known for inverting NF-EXP requires computing shortest vectors in lattices whose dimension is the degree of the field.

This can be the beginning of a lot of researches.

An important thing to consider in cryptography is the level of security which is chosen to say that a cryptosystem is secure. Usually, we can consider the complexity of algorithm, the way informations are stored and distributed, the probability of breaking a system using a special algorithm. But the cryptanalysists are imaginative people and a new family of factors must be considered, the observable behaviour. It means for example that, when you know an algorithm, you can deduce the secret parameters by measuring the time of computation, the volume of communications, the different emissions (electromagnetic, thermal) or the interferences with other processes running on the same computer. This idea is developed in [9].

References

[1] *Remarks on the security of the elliptic curve cryptosystem*, available on the net at
http://www.certicom.com, 1997.

[2] T.A.Berson, *Failure of the McEliece public-key cryptosystem under message-resend and related message attack,in Advances in Cryptology - CRYPTO'97*, Lecture Notes in Computer Science 1294, Springer-verlag, (p.213-220), 1997.

[3] C.Blair, *Notes on Cryptography*, available on the net at
http://www.staff.uiuc.edu/~c-blair/crn.html, 1991.

[4] G.Brassard, *Quantum Information Processing: The Good, the Bad and the Ugly, in Advances in Cryptology - CRYPTO'97*, Lecture Notes in Computer Science 1294, Springer-verlag, (p.337-341), 1997.

[5] J.Buchmann, S.Paulus, *A one-way function based on ideal arithmetic in number fields, in Advances in Cryptology - CRYPTO'97*, Lecture Notes in Computer Science 1294, Springer-verlag, (p.385-394), 1997.

[6] W.Fulton, *Algebraic curves (An introduction to algebraic geometry)*, W.A.Benjamin, INC., 1968.

[7] J.Guajardo, C.Paar, *Efficient algorithms for elliptic curve cryptosystems, in Advances in Cryptology - CRYPTO'97*, Lecture Notes in Computer Science 1294, Springer-verlag, (p.342-356) 1997.

[8] N.Koblitz, *CM-curves with good cryptographic properties, in Advances in Cryptology - CRYPTO'91*, Lecture Notes in Computer Science 576, Springer-verlag, (p.279-287) 1991.

[9] P.C.Kocher, *Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other system, in Advances in Cryptology - CRYPTO'96* , Lecture Notes in Computer Science 1109, Springer-verlag, (p.104-113) 1996.

[10] A.J.Menezes, *Elliptic curves Public key cryptosystems*, (thesis), Kluwer Academic Publishers, 1993.

[11] A.Menezes, T.Okamoto, S.Vanstone, *Reducing elliptic curve logarithms to logarithms in a finite field*, in *Proceedings of the twenty third annual ACM symposium on theory of computing*, p.80-89, ACM Press, 1991.

[12] A.J.Menezes (editor), I.F.Blake, X.Gao, R.C.Mullin, S.A.Vanstone, T.Yaghoobian, *Applications of finite fields*, Kluwer Academic Publishers, 1993.

[13] P.L.Montgomery, *Speeding the Pollard and Elliptic Curve Methods of Factorisation*, in *Mathematics of Computation*, vol. 48, p.243-264, American Mathematical society, 1987.

[14] F.Morain, *These de doctorat d'Etat,*, available on the net at ftp://ftp.inria.fr/INRIA/publication/Theses/TU-0144/, 1990.

[15] J.A.Salinas, *An improved algorithm for arithmetic on a family of elliptic curves, in Advances in Cryptology - CRYPTO'97*, Lecture Notes in Computer Science 1294, Springer-verlag, (p.357-371) 1997.

[16] I.A.Semaev, *Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p*, in *Mathematics of Computation*, vol. 67, p.353-356, American Mathematical society, 1998.

[17] J.H.Silverman, *The arithmetic of elliptic curves*, Springer-Verlag, 1986.

[18] J.H.Silverman , J.Suzuki, *Elliptic curve discrete logarithms and the index calculus*, preprint, available on the net at http://www.jacobi.math.brown.edu/~jhs/, 1998.

[19] R.D.Silverman, *The Multiple Polynomial Quadratic Sieve*, in *Mathematics of Computation*, vol. 48, p.329-339, American Mathematical society, 1987.

[20] N.P.Smart, *The discrete logarithm problem on elliptic curves of trace one*, preprint, available on the net at http://www.hpl.hp.com/techreports/97/HPL-97-128.html, 1997.

[21] D.R.Stinson, *Cryptography: Theory and practice*, CRC Press, 1995.