SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

Elliptic Curves Gone Cryptic

by

Mihai-Dinu Lazarescu

2011 - No 10

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET, 106 91 STOCKHOLM

Elliptic Curves Gone Cryptic

Mihai-Dinu Lazarescu

Självständigt arbete i matematik 30 högskolepoäng, GN

Handledare: Rikard Bögvad

2011

Foreword

Ever since the dawn of civilization human beings have exchanged information and occasionally secret information. Irrespective of the method of encryption these questions can be addressed mathematically.

Once the message is encrypted and transmitted across (usually) insecure channels it is of utmost importance that unauthorized parties cannot break the encryption. Decryption should be easy for the authorized but (ideally) impossible for the unauthorized.

You could encrypt by hashing for which you need a function easy to compute but very hard to invert. Or, you could encrypt using a secret key or, of late, public!

Public key cryptosystems started making their appearance with Diffie and Hellman's public key exchange in 1976 and the subsequent creation of the RSA public key cryptosystem by Rivest, Shamir and Adleman in 1978. The public key cryptosystems guard themselves against "burglars", whom we shall call *cryptanalysts*, through mathematical problems very hard to solve. The mathematical cryptographic model may be very simple indeed yet breaking it can be extremely hard.

These systems usually rely essentially either on group theory or lattice theory. In this paper we shall consider the group theoretical variant. The standard group in use is $\mathbb{F}_p \cong \mathbb{Z}_p$ yet lately a more exotic group has come to the fore, viz. the additive group of an elliptic curve, $\mathsf{E}(\mathbb{F}_p)$. Since the elliptic curve approach mimicks to a large extent the standard approach we shall present them both in parallel.

The first chapter is about ciphers, *symmetric* (private key) ciphers and *asymmetric* (public key) ciphers. It is intended more as a narrative background to the rest of the paper and as a methodological discussion than as a mathematical argument.

Chapter two is devoted to cryptography. Here we present some cryptosystems in use. Special focus will be placed on the elliptic curve approach.

The hard mathematical problems at the core of the public key cryptosystems are either the discrete logarithm problem or the prime factorization

problem. We shall consider only the former. In chapter three we discuss the cryptanalytical issues.

The elliptic curves and the group structure defined on them are presented in chapter four as a kind of appendix. Very briefly we shall discuss even hyperelliptic curves.

Contents

1 Ciphers

“ ’ Mine is a long and a sad tale.’

’ It *is* a long tail, certainly, but why do you call it sad? ’

’ Turn witch into fairy.’

’ Witch, winch, wench, tench, tenth, tents,

tints, tits, tills, fills, falls, fails, fairs, fairy!’ “

(from *Original Games and Puzzles* by Lewis Carroll)

1.1 Substitution ciphers

Arabella and Beau would like to exchange *billets doux* but have grown weary of Cupid's constant surveillance. They decide to shift every letter in the alphabet (standard Latin alphabet, 26 letters) 6 steps forward, so a will become g, b will become h , ... , and, finally, z will become f. This is the simplest type of cipher, the *shift* or *Caesar* cipher. The encryption could be given as $\mathsf{letter} \longmapsto \mathsf{letter}{+}\mathsf{6}$ where we have labeled the letters from a to z by numbers from $\mathsf{1}$ to $\mathsf{26}$. This is not very difficult to break, in the worst of cases you can try all possible shifts, $\mathsf{26}$ in number. An improvement may be this: write the alphabet in two rows in opposite directions and match.

$$a, b, c, ...m, n,.....,x, y, z$$

$$z, y, x, ...n, m,.....,c, b, a$$

It should be harder to break though not excessively so.

The two ciphers above are examples of *simple substitution ciphers* which may be viewd as functions

$$\{a, b, c, ..., x, y, z\} \longrightarrow \{a, b, c, ..., x, y, z\}$$

with domain = plaintext letters and range = ciphertext letters, assigning to each plaintext letter a different ciphertext letter. An arbitrary function of this kind can be viewed as a randomly chosen permutation of the $\mathsf{26}$ letters. Consequently, there are $\mathsf{26!} > \mathsf{10^{26}}$ different simple substitutions ciphers. Each such simple substitution can be presented as a table with two rows:

- upper row = plaintext letters
- lower row = ciphertext letters

and this table can be considered as the *key*. Decrypting an encrypted text without knowing the key is called *cryptanalysis*.

How hard is the task of the cryptanalyst in this case? Try brute force and check all $\mathsf{26!}$ possibilities. Say that Cupid's computer can check $\mathsf{10^{26}}$ cipher alphabets per second. The process should take

$$\frac{\mathsf{26!}}{10^6 \cdot 60 \cdot 60 \cdot 24 \cdot 365} > \frac{\mathsf{10^{26}}}{10^6 \cdot 60 \cdot 60 \cdot 24 \cdot 365} > \frac{10^{18}}{10^5} > \mathsf{10^{13}} \text{ years.}$$

Compare this to the estimated age of the universe, of the order of $\mathsf{10}^{10}$ years! Yet, you should not dispair. One should always consider the *best* of the known methods of breaking an encryption. One reasonably good method of cryptananlysis in this case would be using a frequency table that gives the frequency with which a certain letter appears in a text in a given language. (See, for instance, Hoffstein, Pipher & Silverman, pp 5-9).

1.2 Symmetric ciphers

Arabella and Beau choose a *secret key* k from the space $\mathscr{K}$ of all possible keys to encrypt message m from the space $\mathscr{M}$ of all possible messages (plaintext) and obtain ciphertext c which belongs to the space $\mathscr{C}$ of all possible cirphertexts.

Encryption becomes thus a mapping:

$$e : \mathscr{K} \times \mathscr{M} \longrightarrow \mathscr{C} \ .$$

Decryption then is the inverse operation/function:

$$d : \mathscr{K} \times \mathscr{C} \longrightarrow \mathscr{M} \ ,$$

such that $\forall \mathsf{k} \, \epsilon \, \mathcal{K} \, \forall \mathsf{m} \, \epsilon \, \mathcal{M} \;\; d(\mathsf{k}, \, e \, (\mathsf{k},\mathsf{m})) = \mathsf{m}$.

A more compact notation would be

$$e_k : \mathscr{M} \longrightarrow \mathscr{C} \text{ and } d_k : \mathscr{C} \longrightarrow \mathscr{M}$$

with property $\forall \mathsf{k} \, \epsilon \, \mathcal{K} \, \forall \mathsf{m} \, \epsilon \, \mathcal{M} \; d_k(e_k(\mathsf{m})) = \mathsf{m}$. As I said before d_k must be the inverse of e_k, $d_k = e_k^{-1}$.

The astute Cupid knows what encryption method Arabella and Beau use, i.e. Cupid knows the function e_k and *ipso facto* also function d_k. What he does ***not*** know is the key k.

A basic premise of modern cryptography is *Kerckhoff's principle* : the security of a cryptosystem should depend only on the secrecy of the key, not on the secrecy of the encryption algorithm.

The *sine qua non* conditions for a successful cipher ($\mathscr{K}$, $\mathscr{M}$, $\mathscr{C}$, e_k, d_k) are:

1. $\forall \mathsf{k} \, \epsilon \, \mathscr{K} \; \forall \mathsf{m} \, \epsilon \, \mathscr{M}$: it must be ***easy*** to compute the ciphertext $e_k(\mathsf{m})$.

2. $\forall \mathsf{k} \, \epsilon \, \mathscr{K} \; \forall \mathsf{c} \, \epsilon \, \mathscr{C}$: it must be ***easy*** to compute the plaintext $d_k(\mathsf{c})$.

3. Given ciphertexts $\{\mathsf{c}_i \epsilon \mathscr{C}\}_{i=1}^{n}$ encrypted by means of the key $\mathsf{k}\epsilon\ \mathscr{K}$ it must be ***very difficult*** to compute the corresponding $d_k(\mathsf{c})$ without knowing k.

4. *Desideratum*. Given pairs $(\mathsf{m},\ \mathsf{c})\ \epsilon\ \mathscr{M}\ \times\ \mathscr{C}$, $i = 1, 2$,.....,n, it must be ***difficult*** to decrypt any ciphertext c that is not given in the list without knowing k. This is security against a chosen plaintext attack.

Since we want to construct a mathematical model for encryption and decryption it is most convenient and natural to consider keys, plaintexts and ciphertexts as numbers and, furthermore, as binary numbers. Such an *encoding scheme*, converting text into numbers, is given by the ASCII code (***A***merican ***S***tandard ***C***ode for ***I***nformation ***I***nterchange). An encoding scheme is entirely public knowledge and everyone uses it for the same purposes!

An *encryption scheme* is used to hide information from anyone who does not possess the secret key. Using an encoding scheme we may view every plaintext or ciphertext as a sequence of binary blocks, each block consisting of eight *bits* (***b***inary dig***it***, 0 or 1). A block of eight bits is called a *byte*. A byte is often written as a decimal number between 0 and 255 or as a two-digit hexadecimal number between 00 and FF.

For simplicity we may decide to view the elements of $\mathscr{M}$ as bit strings of a fixed length B which we call the blocksize of the cipher. The encryption function then takes a message block from $\mathscr{M}$ consisting of exactly B zeros and ones and transforms it into a ciphertext block of exactly B zeros and ones in $\mathscr{C}$. If the plaintext ends with a block of fewer than B bits we fill the tail of the block with zeros. All this is public knowledge!

Since we encrypt and decrypt one block at a time it suffices to consider the process for a single plaintext block $\mathsf{m}\ \epsilon\ \mathscr{M}$. We identify the binary string m with the corresponding number in binary form thus identifying $\mathscr{M}$ with the set of integers m satisfying $0 \leq \mathsf{m} < 2^{\mathsf{B}}$ through the correspondence

$$\mathsf{m}_{\mathsf{B}-1}\mathsf{m}_{\mathsf{B}-2}\ \ldots\ldots\ldots\ldots\ \mathsf{m}_2\mathsf{m}_1\mathsf{m}_0$$

$$\updownarrow$$

$$\mathsf{m}_{\mathsf{B}-1}2^{\mathsf{B}-1}+\mathsf{m}_{\mathsf{B}-2}\ 2^{\mathsf{B}-2}+\ \ldots\ldots\ldots\ldots\ldots\ +\ \mathsf{m}_2 2^2+\mathsf{m}_1 2+\mathsf{m}_0,$$

where $\mathsf{m}_i\ \epsilon\ \{0, 1\}$, $i = 0, 1, \ldots\ldots\ldots\ldots, \mathsf{B}-1$.
We make similar identifications for $\mathscr{C}$ and $\mathscr{K}$ and thus we have:

$$\mathscr{K}\ =\ \left\{\mathsf{k}\epsilon\mathbb{Z}\ \middle|\ 0 \leq \mathsf{k} < 2^{\mathsf{B}_\mathsf{k}}\right\}$$

$$\mathscr{M} = \left\{ \mathsf{m} \epsilon \mathbb{Z} \mid 0 \leq \mathsf{m} < 2^{\mathsf{B_m}} \right\}$$

$$\mathscr{C} = \left\{ \mathsf{c} \epsilon \mathbb{Z} \mid 0 \leq \mathsf{c} < 2^{\mathsf{B_c}} \right\}.$$

It is of course not necessary to have $\mathsf{B_k} = \mathsf{B_m} = \mathsf{B_c}$ but it can be wise to let $\mathsf{B_k} = \mathsf{B_m} = \mathsf{B_c}$.

Let p be some sufficiently (!) large prime number and $\mathscr{K} = \mathscr{M} = \mathscr{C}$ $= \{1, 2, ..., \mathsf{p} - 1\} = \mathbb{F}_p^\star$, the group of units of the finite field $\mathbb{F}_p$ ($\cong \mathbb{Z}_p$), which is a multiplicative group.

Arabella and Beau choose as their (common) secret key an integer $\mathsf{k} \, \epsilon \, \mathbb{F}_p^\star$ and settle for the encryption function e_k defined by the congruence $e_k(\mathsf{m})$ $\equiv \mathsf{k} \cdot \mathsf{m} \pmod{p}$. Of course d_k will be given by $d_k(\mathsf{c}) \equiv \mathsf{k'} \cdot \mathsf{c} \pmod{p}$, where $\mathsf{k'} \equiv \mathsf{k}^{-1} \pmod{p}$.

Nota bene:

- If p is relatively small then Cupid may break the key by a brute force attack, i.e. an exhaustive search attack, since he knows the decryption algorithm (Kerckhoff's principle). He takes every $\mathsf{k} \, \epsilon \, \mathscr{K}$ and computes $d_k(\mathsf{c})$. Assuming that he can tell which text is a valid plaintext and which is invalid he will recover the message m. An exhaustive search is feasible (according to Hoffman, Pipher & Silverman) if the space has at most 2^{80} elements, so Arabella and Beau should choose $\mathsf{B}_k \geq 80$.

- It is also known that it is easier to find matching objects (collisions) in a set than it is to find a specific object in the same set. Such search methods are called *collision* or $meet - in - the - middle$ attacks. It turns out that if such methods are available Arabella and Beau should choose $\mathsf{B}_k \geq 160$. (See Hoffman, Pipher & Silverman).

Now, if Cupid tries a brute force attack on k, and $2^{159} < p < 2^{160}$, he will have a hard time trying approximately 2^{160} possibilities ($2^{160} - 2^{159}$ $= 2^{159}(2 - 1) = 2^{159}$).

What if he knows some ciphertext c?

e_k: $\mathscr{M} \longrightarrow \mathscr{C}$ is one-to-one and the cardinalities of $\mathscr{M}$ and $\mathscr{C}$ are equal and finite so e_k is also onto, for any choice of k.

Consequently, for every $\mathsf{c} \, \epsilon \, \mathscr{C}$ and any $\mathsf{k} \, \epsilon \, \mathscr{K}$ there exists an $\mathsf{m} \, \epsilon \, \mathscr{M}$ such that $e_k \, (\mathsf{m}) = \mathsf{c}$. But then, since $e_k \, (\mathsf{m}) = \mathsf{km} \pmod{p}$ we solve the congruence $\mathsf{km} \equiv \mathsf{c} \pmod{p}$ and recover the message as $\mathsf{m} \equiv \mathsf{k}^{-1}\mathsf{c} \pmod{p}$.

This shows that although it would be difficult for Cupid to recover the key k (for large p) it would not be impossible. The conclusion must be that the cryptosystem above has Properties $\mathtt{1}, \mathtt{2}$ and $\mathtt{3}$ but not Property $\mathtt{4}$.

What about the encryption function $e_k(\mathsf{m}) = \mathsf{k} \cdot \mathsf{m}$? The cipher still has Properties $\mathtt{1}$ and $\mathtt{2}$ but not Property $\mathtt{3}$ any longer because, if Cupid tries to decrypt $\mathsf{c} = \mathsf{k} \cdot \mathsf{m}$, although he still has the difficult task of factoring a large number, having acquired ciphertexts c_1, c_2 , ... , c_n , it is fairly probable that

$$\gcd(\mathsf{c}_1, \mathsf{c}_2 \text{ , ... , } \mathsf{c}_n) = \gcd(\mathsf{km}_1, \mathsf{km}_2 \text{ , ... , } \mathsf{km}_n) =$$

$$\mathsf{k}\cdot \gcd(\mathsf{m}_1, \mathsf{m}_2 \text{ , ... , } \mathsf{m}_n) \text{ .}$$

Instead of $e_k(\mathsf{m}) \equiv \mathsf{k} \cdot \mathsf{m} \pmod{p}$ one could try $e(\mathsf{m}) \equiv \mathsf{m}+\mathsf{k} \pmod{p}$ with $d_k(\mathsf{c}) \equiv \mathsf{c}-\mathsf{k} \pmod{p}$, this being the shift cipher. Another variant is the *affine* cipher, a combination of shift and multiplication. Its key is a pair $\mathsf{k} =(\mathsf{k}_1,\mathsf{k}_2)$ and

$$e_k(\mathsf{m}) \equiv \mathsf{k}_1\mathsf{m}+\mathsf{k}_2 \pmod{p} \text{ with } d_k(\mathsf{c}) \equiv \mathsf{k}_1^{-1}(\mathsf{c} - \mathsf{k}_2) \pmod{p}.$$

A generalization of the affine cipher is the *Hill* cipher where

- k_1 is a $n \times n$ matrix with integer entries mod p, hence k_1^{-1} is the inverse matrix of k_1
- m, c, and k_2 are column vectors of n integers mod p

Both the affine and the Hill ciphers lack Property $\mathtt{4}$, i.e. they are vulnerable to plaintext attacks (See Hoffman, Pipher & Silverman).

Let us consider the following operation:

xor denoted by $\oplus$, the exclusive disjunction. Given β, $\beta' \,\epsilon\, \{\mathtt{0}, \mathtt{1}\}$ we define

$$\beta \oplus \beta' = \begin{cases} \mathtt{0} & , \beta = \beta' \\ \mathtt{1} & , \beta \neq \beta' \end{cases}$$

xor is obviously addition modulo 2.

For example, $\mathtt{10110} \oplus \mathtt{11010} = [\mathtt{1} \oplus \mathtt{1}]\,[\mathtt{0} \oplus \mathtt{1}]\,[\mathtt{1} \oplus \mathtt{0}]\,[\mathtt{1} \oplus \mathtt{1}]\,[\mathtt{0} \oplus \mathtt{0}] = \mathtt{01100}$.

Arabella and Beau can construct now the following cipher:

$$e_k(\mathsf{m}) = \mathsf{k} \oplus \mathsf{m} \text{ and } d_k(\mathsf{c}) = \mathsf{k} \oplus \mathsf{c}.$$

Observe that e_k and d_k are the same function, i.e. $d_k = e_k^{-1} = e_k$.

If they wanted to use this cipher (the *Vernam one − time pad*) to exchange N bits of information they would need to know already N bits of secret information since the key is as long as the plaintext. (See Hoffman, Pipher & Silverman, pg 43 & pg 249). This makes the cipher very cumbersome and inefficient for most practical applications. It is nonetheless completely secure if the key is used only once. If the key is reused, by mistake or in want of key material, then Cupid could use the fact that

$$\mathsf{c} \oplus \mathsf{c}' = (\mathsf{k} \oplus \mathsf{m}) \oplus (\mathsf{k} \oplus \mathsf{m}') = \mathsf{m} \oplus \mathsf{m}'$$

thus getting information about m or m' although it is not quite clear how he could determine k, m or m'. And yet, dispensing so easily with the key k should be alarming!

At this stage we can ask ourselves if it is at all possible to use a single relatively short key k to send securely and efficiently arbitrary messages. Suppose we can define a function:

$$\mathsf{R}\colon \mathscr{K} \times \mathbb{Z} \longrightarrow \{0, 1\}$$

satisfying the following conditions:

1. $\forall \mathsf{k} \in \mathscr{K}\ \forall j \in \mathbb{Z}$ it is ***easy*** to compute $\mathsf{R}(\mathsf{k},j)$

2. Given an arbitrarily long sequence of integers j_1, j_2, ... , j_n and given all the values $\mathsf{R}(\mathsf{k},j_1)$, $\mathsf{R}(\mathsf{k},j_2)$, ..., $\mathsf{R}(\mathsf{k},j_\mathsf{n})$ it is ***hard*** to determine k

3. Given any list of integers j_1, j_2, ... , j_n and given all the values $\mathsf{R}(\mathsf{k},j_1)$, $\mathsf{R}(\mathsf{k},j_2)$, ..., $\mathsf{R}(\mathsf{k},j_\mathsf{n})$ it is ***hard*** to guess the value $\mathsf{R}(\mathsf{k},j)$ with better than 50% chance of success for any value of j not already in the list.

In that case we can start with a key k , compute the sequence

$$\mathsf{R}(\mathsf{k},1),\ \mathsf{R}(\mathsf{k},2),\ .....$$

and then use this sequence of bits as the key for a one-time pad. But is this sequence truly random ? R is actually a pseudorandom number generator. Do such generators exist?

We can construct candidates for R in two ways:

- Apply an *ad hoc* collection of mixing operations, efficient to execute and ***hard*** to untangle. This is the basis for most practical symmetric ciphers, including DES and AES, the two systems most widely used today.
- Construct R using a function whose efficient inversion is a well-known ***hard*** (or so believed to be) mathematical problem. Unfortunately this second approach seems to be far less efficient than any *ad hoc* constructions.

1.3 Asymmetric ciphers

In order to use a symmetric cipher Arabella and Beau must meet and agree on a secret key k. But what if they cannot meet and any communication between them is totally monitored by Cupid? Well, where there is a will there is a way.

Diffie and Hellman had the cunning insight that this is possible under certain conditions. Arabella buys a safe (the *public* key) with a narrow slot and locks it with a personal key (the *private* key). The safe is displaced in a public place. Beau comes by and drops a message through the slot (*encryption*). Later Arabella unlocks the safe with her key (*decryption*). Incidentally, anyone in the world can send encrypted messages to Arabella, not only Beau!

Mathematically, this can be formulated like this. For $\mathsf{k} \in \mathscr{K}$ it holds that the complete key consists of a pair of keys:

$$\mathsf{k} = (\mathsf{k}_{\text{priv}}, \mathsf{k}_{\text{pub}}),$$

one private and one public. For every k_{pub} there is a corresponding encryption function:

$$e_{\mathsf{k}_{\text{pub}}} \colon \mathscr{M} \longrightarrow \mathscr{C}$$

and for every k_{priv} there is a corresponding decryption function:

$$d_{\mathsf{k}_{\text{priv}}} \colon \mathscr{C} \longrightarrow \mathscr{M}$$

with the property that if $(\mathsf{k}_{\text{priv}}, \mathsf{k}_{\text{pub}}) \in \mathscr{K}$ then $\forall \mathsf{m} \in \mathscr{M}$ $d_{\mathsf{k}_{\text{priv}}}(e_{\mathsf{k}_{\text{pub}}}(\mathsf{m})) = \mathsf{m}$.

If such an asymmetric cipher is to be secure then Cupid must have a very hard time determining the decryption function $d_{\mathsf{k}_{\text{priv}}}$ even though he knows

the public key k_{pub}. Arabella can send k_{pub} to Beau any way she pleases and Beau can send back the ciphertext $e_{\mathsf{k}_{\text{pub}}}(\mathsf{m})$ without worrying about Cupid. Decryption should be easy only if you have access to the private key k_{priv} and Arabella is, hopefully, the only one with that information. That is Arabella's *trapdoor information*.

Otherwise decryption should be very hard. The difficulty can consist in solving, e.g.

1. the *discrete logarithm* problem (DLP) for a multiplicative group (the classical ElGamal cryptosystems)
2. the *discrete logarithm* problem (ECDLP) for the additive group of an elliptic curve (the elliptic curve ElGamal cryptosystems)
3. the *prime factorization* problem (the RSA cryptosystems)
4. the *short vector* problem (SVP) in a lattice (the NTRU cryptosystems)

In this paper we shall consider only the first two cases.

1.4 Digital signatures

Encryption systems secure communications over an insecure network. But there are situations where you must authenticate the source of the message or even its recipient. Arabella must sign her message to Beau.

Let us use the analogy of a bank deposit vault. It has a (narrow) slot which is the *public encryption key*. Anyone can use it to deposit an envelope (the *message*) but no one except the owner of the combination (the *private decryption key*) can open it (decrypt and read the message). So a public key cryptosystem can be viewed as a digital version of the bank deposit vault.

In past ages people used to *sign* their letters with a signet ring (the *private signing key*) with a recessed image which could be pressed into the melted wax previously dropped onto the document. So a *digital signature* may be the analogue of a signet ring.

The following are the ingredients of a *digital signature scheme*:

- a *private signing key* ($\mathsf{k}^{\mathsf{priv}}$)
- a *public signing key* ($\mathsf{k}^{\mathsf{pub}}$)

- a *signing algorithm* (sign) that takes as input a digital message m and a private key k^{priv} and returns m^{sign} for m
- a *verification algorithm* (ver) that takes as input a digital message m, a signature m^{sign} and a public key k^{pub} and returns TRUE if m^{sign} is a signature for m associated to the private key k^{priv} and FALSE, otherwise.

It is essential though that the owner of k^{priv} be able to create valid signatures at the same time as knowledge of k^{pub} does not reveal k^{priv}. There are two necessary general conditions for a secure digital signature scheme:

- Given k^{pub}, an attacker cannot feasibly determine k^{priv} or any other private key that produces the same signature as k^{priv}.
- Given k^{pub} and a list of documents D_1, D_2, ... , D_n with their signatures D_1^{sign}, D_2^{sign}, ... , D_n^{sign}, an attacker cannot feasibly determine a valid signature or any document D that is not already in the list.

You should keep in mind that every time you sign a document you reveal a new document/signature pair and this provides new information to an attacker, so the second condition says that the attacker gains nothing except the new pair. An attack that makes use of a large number of already known signatures is a *transcript attack* therefore we say that the second condition requires that a digital signature should not be vulnerable to transcript attacks. In real world applications digital signature schemes must avoid a number of very subtle, but fatal, security problems. This is not of our interest or concern here.

2 Elliptic Curve Cryptography

“ 'Twas brillig, and the slithy toves

Did gyre and gimble in the wabe;

All mimsy were the borogoves,

And the mome raths outgrabe.

.....

' It seems very pretty,' she said when she had finished it, 'but it's *rather* hard to understand!'

(You see she didn't like to confess even to herself, that she couldn't make it out at all.) “

(From *Through the Looking − Glass*, ch.1 *Looking − Glass House*, by Lewis Carroll)

2.1 The Diffie-Hellman key exchange

Choose a large prime p and a nonzero integer g mod p and make them public. It is advisable to choose g such that its order in $\mathbb{F}_p^\star$ is a large prime.

Arabella chooses a secret integer a and Beau a secret integer b. Arabella then computes the value A and Beau the value B :

$$\mathsf{A} \equiv g^a \pmod{p}$$

$$\mathsf{B} \equiv g^b \pmod{p}$$

and exchange them.

New computations give Arabella the value A' and Beau the value B' as follows:

$$\mathsf{A}' \equiv \mathsf{B}^a \pmod{p}$$

$$\mathsf{B}' \equiv \mathsf{A}^b \pmod{p}$$

$$\mathsf{A}' \equiv \mathsf{B}^a \equiv \left(g^b\right)^a \equiv g^{ab} \equiv \left(g^a\right)^b \equiv \mathsf{A}^b \equiv \mathsf{B}'$$

This common value is the ***exchanged key***. Now they can use this as the common key for a symmetric cipher. If Cupid wants it he must solve the congruence

$$g^a \equiv \mathsf{A} \pmod{p} \text{ for } a$$

or

$$g^b \equiv \mathsf{B} \pmod{p} \text{ for } b.$$

We shall call this the DLP, the discrete logarithm problem, for reasons that will become apparent in chapter three.

This key exchange is due to Whitfield Diffie and Martin Hellman who published their paper "New Directions in Cryptography" in 1976 and practically laid the foundations for what was to become the public key cryptosystems. Others seem to have invented the same key exchange system before though without making their results public for various reasons. (See Hoffstein, Pipher & Silverman). But all this was only a public exchange of a secret key. As yet no public key cryptosystem was available.

DHP

The Diffie - Hellman Problem is the problem of computing the value of $g^{ab} \pmod p$ from the known values $g^a \pmod p$ and $g^b \pmod p$. The DHP is no harder than the DLP ($\mathsf{DHP} \preccurlyeq \mathsf{DLP}$) but nobody knows the answer to the converse question.

ECDHP

Choose a particular $\mathsf{E}(\mathbb{F}_p)$ and a particular point $\mathsf{P} \in \mathsf{E}(\mathbb{F}_p)$ and make them public. Arabella chooses a secret integer n_A and Beau chooses a secret integer n_B. Arabella and Beau then compute their respective multiples of P:

$$\mathsf{Q_A} = n_\mathsf{A}\mathsf{P}$$

$$\mathsf{Q_B} = n_\mathsf{B}\mathsf{P}$$

and exchange them.
New computations give them the value

$$\mathsf{A}' = n_\mathsf{A}\mathsf{Q_B} = n_\mathsf{A}(n_\mathsf{B}\mathsf{P}) = n_\mathsf{A}n_\mathsf{B}\mathsf{P} = n_\mathsf{B}(n_\mathsf{A}\mathsf{P}) = n_\mathsf{B}\mathsf{Q_A} = \mathsf{B}'$$

This common value is their exchanged key.

Example. Let us look at the following set up:

$$\mathsf{E}\text{: } y^2 = x^3 + 171x + 853$$

$$p = 2671$$

$$\mathsf{P} = (1980, 431)$$

Arabella sends Beau the point $\mathsf{Q_A} = (2110, 543)$. Beau decides to use the secret multiplier $n_\mathsf{B} = 1943$. What point is Beau going to send back to Arabella? Well, of course, $\mathsf{Q_B} = n_\mathsf{B}\mathsf{P} = 1943\mathsf{P}$. But what is this specific point in terms of coordinates?

$$1943 = 1 + 2 + 2^2 + 2^4 + 2^7 + 2^8 + 2^9 + 2^{10}$$

or, in ternary expansion,

$$1943 = 1 + 2 + 2^2 + 2^4 - 2^7 + 2^{11}.$$

We shall either need 10 doublings + 7 additions = 17 point operations, or 11 doublings + 5 additions = 16 point operations. The difference is not enormous but often it can be quite substantial. (See chapter three).

We compute:

$$P = (1980, 431)$$

$$2P = (1950, 1697)$$

$$4P = (1894, 1829)$$

$$8P = (1160, 1268)$$

$$16P = (1116, 2037)$$

$$32P = (2125, 1001)$$

$$64P = (862, 2268)$$

$$128P = (1135, 932)$$

$$256P = (586, 2069)$$

$$512P = (2040, 1378)$$

$$1024P = (1718, 584)$$

$$2048P = (2091, 1669)$$

and

$$P + 2P = (415, 301)$$

$$3P + 4P = (2288, 2333)$$

$$7P + 16P = (1074, 754)$$

$$23P - 128P = (1704, 589)$$

$$-105P + 2048P = (1432, 667)$$

$$1943\mathsf{P} = (1432, 667)$$

Beau will send Arabella the point $(1432, 667)$

What is their secret shared value? It is

$$n_{\mathsf{A}}\mathsf{Q}_{\mathsf{B}} = n_{\mathsf{B}}\mathsf{Q}_{\mathsf{A}} = 1943\mathsf{Q}_{\mathsf{A}}$$

New computations:

$$\mathsf{Q}_{\mathsf{A}} = (2110, 543)$$

$$2\mathsf{Q}_{\mathsf{A}} = (1687, 1454)$$

$$4\mathsf{Q}_{\mathsf{A}} = (1470, 1137)$$

$$8\mathsf{Q}_{\mathsf{A}} = (1189, 577)$$

$$16\mathsf{Q}_{\mathsf{A}} = (967, 1539)$$

$$32\mathsf{Q}_{\mathsf{A}} = (2000, 1792)$$

$$64\mathsf{Q}_{\mathsf{A}} = (844, 699)$$

$$128\mathsf{Q}_{\mathsf{A}} = (1655, 1926)$$

$$256\mathsf{Q}_{\mathsf{A}} = (1775, 523)$$

$$512\mathsf{Q}_{\mathsf{A}} = (1157, 973)$$

$$1024\mathsf{Q}_{\mathsf{A}} = (1871, 1455)$$

$$2048\mathsf{Q}_{\mathsf{A}} = (1535, 1641)$$

and

$$\mathsf{Q}_{\mathsf{A}} + 2\mathsf{Q}_{\mathsf{A}} = (809, 2136)$$

$$3\mathsf{Q}_{\mathsf{A}} + 4\mathsf{Q}_{\mathsf{A}} = (928, 1620)$$

$$7\mathsf{Q}_{\mathsf{A}} + 16\mathsf{Q}_{\mathsf{A}} = (401, 2422)$$

$$23\mathsf{Q}_{\mathsf{A}} - 128\mathsf{Q}_{\mathsf{A}} = (167, 869)$$

$$-105\mathsf{Q}_{\mathsf{A}} + 2048\mathsf{Q}_{\mathsf{A}} = (2424, 911)$$

$$1943\mathsf{Q_A} = (2424, 911)$$

Arabella and Beau share the secret value $(2429, 911)$

Cupid has to solve the ECDLP $n_\mathsf{A}\mathsf{P} = \mathsf{Q_A}$ for n_A or $n_\mathsf{B}\mathsf{P} = \mathsf{Q_B}$ for n_B in order to get the key. We can formulate even here a ECDHP: compute $n_\mathsf{A}n_\mathsf{B}\mathsf{P}$ knowing the values $n_\mathsf{A}\mathsf{P}$ and $n_\mathsf{B}\mathsf{P}$.

In the example above Cupid must solve the ECDLP :

$$n_\mathsf{A}(1980, 431) = (2110, 543).$$

We still have ECDHP $\preccurlyeq$ ECDLP.

When exchanging points on an elliptic curve one need not really exchange both coordinates. It suffices to exchange only the $x-$coordinate since the $y-$coordinate may be recuperated from the equation $y^2 = x^3 + ax + b$. But, if Arabella does so and sends Beau only the $x-$coordinate of $\mathsf{Q_A}$ then he either chooses the "correct" y, thus effectively using $\mathsf{Q_A}$, or chooses the "incorrect" y, thus using $-\mathsf{Q_A}$. The following computations will give Beau $\pm n_\mathsf{B}\mathsf{Q_A} = \pm(n_\mathsf{A}n_\mathsf{B})\mathsf{P}$ and Arabella gets the same, $\pm n_\mathsf{A}\mathsf{Q_B} = \pm(n_\mathsf{B}n_\mathsf{A})\mathsf{P} = \pm(n_\mathsf{A}n_\mathsf{B})\mathsf{P}$, so both can use the $x-$coordinate as the secret key.

Example. Arabella and Beau decide to exchange a new piece of secret information using the same prime, curve and point. This time Arabella sends Beau only the $x-$coordinate of her point $\mathsf{Q_A}$, viz. $x_\mathsf{A} = 2$. On receiving this value Beau computes

$$y^2 = 2^3 + 171 \cdot 2 + 853 = 1203$$

He solves this equation mod 2671 and gets two solutions: $y_1 = 96$ and $y_2 = 2575$. So Arabella might choose as her secret point either $(2, 96)$ or $(2, 2575)$. Beau then decides to use the secret multiplier $n_\mathsf{B} = 875$ and he must send her back the $x-$coordinate of the point $\mathsf{Q_B} = n_\mathsf{B}\mathsf{P} = 875\mathsf{P}$.

Back to the computer:

$$875 = 1 + 2 + 2^3 + 2^5 + 2^6 + 2^8 + 2^9$$

We have already computed enough points so we get:

$$\mathsf{P} + 2\mathsf{P} = (415, 301)$$

$$3\mathsf{P} + 8\mathsf{P} = (1858, 644)$$

$$11\mathsf{P} + 32\mathsf{P} = (247, 1420)$$

$$43\mathsf{P} + 64\mathsf{P} = (303, 2012)$$

$$107\mathsf{P} + 256\mathsf{P} = (921, 157)$$

$$363\mathsf{P} + 512\mathsf{P} = (161, 2040)$$

$$875\mathsf{P} = (161, 2040)$$

We conclude that Beau sends back to Arabella $x_{\mathsf{B}} = 161$.

Furthermore, their secret shared value will be the $x-$coordinate of the point $\pm n_{\mathsf{A}}\mathsf{Q}_{\mathsf{B}} = \pm(n_{\mathsf{A}}n_{\mathsf{B}})\mathsf{P} = \pm\, n_{\mathsf{B}}\mathsf{Q}_{\mathsf{A}}$.

More computations in order to determine $\pm n_{\mathsf{B}}\mathsf{Q}_{\mathsf{A}} = \pm 875\mathsf{Q}_{\mathsf{A}}$ under the possibly wrong but innocuous assumption that $\mathsf{Q}_{\mathsf{A}} = (2, 96)$!

First:

$$\mathsf{Q}_{\mathsf{A}} = (2, 96)$$

$$2\mathsf{Q}_{\mathsf{A}} = (2246, 937)$$

$$4\mathsf{Q}_{\mathsf{A}} = (1077, 2113)$$

$$8\mathsf{Q}_{\mathsf{A}} = (143, 27)$$

$$16\mathsf{Q}_{\mathsf{A}} = (2469, 1258)$$

$$32\mathsf{Q}_{\mathsf{A}} = (2124, 492)$$

$$64\mathsf{Q}_{\mathsf{A}} = (1930, 2279)$$

$$128\mathsf{Q}_{\mathsf{A}} = (1684, 544)$$

$$256\mathsf{Q}_{\mathsf{A}} = (454, 2201)$$

$$512\mathsf{Q}_{\mathsf{A}} = (1306, 607)$$

Second:

$$\mathsf{Q}_{\mathsf{A}} + 2\mathsf{Q}_{\mathsf{A}} = (1150, 326)$$

$$3\mathsf{Q}_{\mathsf{A}} + 8\mathsf{Q}_{\mathsf{A}} = (1566, 1752)$$

$$11\mathsf{Q_A} + 32\mathsf{Q_A} = (915, 2120)$$

$$43\mathsf{Q_A} + 64\mathsf{Q_A} = (1124, 363)$$

$$107\mathsf{Q_A} + 256\mathsf{Q_A} = (2596, 741)$$

$$363\mathsf{Q_A} + 512\mathsf{Q_A} = (1708, 1252)$$

$$875\ \text{“}\mathsf{Q_A}\text{”} = (1708, 1252)$$

Arabella and Beau share the value 1708.

2.2 The ElGamal public key cryptosystem

The Diffie-Hellman key exchange did not as yet constitute a full-fledged public key cryptosystem. It was only a method of sharing a key through public channels but it could not permit exchange of specific information. Such a system was created by Taher ElGamal who published his paper, “A public key cryptosystem and a signature scheme based on discrtet logarithms”, in 1985 in IEEE Trans. Inform. Theory, 31 (4).

Arabella publishes a key and an algorithm. The public key is a number and the algorithm is the method for Beau to encrypt his messages using Arabella’s key. Let us look at the details.

Classical ElGamal cryptosystems

Arabella chooses a large prime p and an element $g \pmod p$ which she makes public, then she chooses a secret/private key, a number a, and computes $\mathsf{A} \equiv g^a \pmod p$. A will be the public key.

Beau wants to send Arabella the message m, an integer $2 \leq \mathsf{m} < p$. He chooses randomly a number $k \pmod p$. This will be an ephemereal key. It will be used to encrypt a single message and then it will be discarded! He computes

$$c_1 \equiv g^k \pmod p$$

$$c_2 \equiv \mathsf{m}\mathsf{A}^k \pmod p.$$

The encryption of m will be the pair (c_1, c_2) and this is sent to Arabella. She decrypts:

$$x \equiv c_1^a \pmod{p}$$

$$x^{-1}c_2 \equiv (c_1^a)^{-1}c_2 \equiv$$

$$\left(g^{ak}\right)^{-1} \cdot \left(\mathsf{m}\mathsf{A}^k\right) \equiv$$

$$\left(g^{ak}\right)^{-1} \cdot \left(\mathsf{m}g^{ak}\right) \equiv$$

$$\left(g^{ak}\right)^{-1} \cdot \mathsf{m} \cdot \left(g^{ak}\right) \equiv \mathsf{m}.$$

Example. Arabella uses the prime $p = 2137$ and the primitive root $g = 10$. She chooses $a = 73$ as her private key and computes her public key

$$\mathsf{A} \equiv g^a = 10^{73} \equiv 1405 \pmod{2137}$$

Beau wants to send her the message $\mathsf{m} = 413$, chooses as an ephemeral key k =281 and computes the two values:

$$c_1 \equiv g^k = 10^{281} \equiv 2094 \pmod{2137}$$

$$c_2 \equiv \mathsf{m}\mathsf{A}^k = 413 \cdot 1405^{281} \equiv 1602 \pmod{2137}$$

The pair $(c_1, c_2) = (2094, 266)$ is the ciphertext that Beau sends Arabella and Arabella computes:

$$x = c_1^a = 2094^{73} \equiv 445 \pmod{2137}$$

$$x^{-1} \equiv 850 \pmod{2137}$$

Finally:

$$c_2 x^{-1} \equiv 1602 \cdot 850 \equiv 413 = \mathsf{m}$$

She has got the message!

Cupid, the cryptanalyst, would have to solve the congruence $g^a \equiv \mathsf{A} \pmod{p}$ for a, a DLP.

Theorem 1. *Fix a prime p and base g for the ElGamal encryption. Suppose that Cupid has access to an oracle that decrypts ElGamal ciphertexts encrypted using arbitrary ElGamal public keys. Then he can use the oracle to solve the Diffie − Hellman problem.*

Cupid's problem is the DHP:

- given $\mathsf{A} \equiv g^a \pmod p$ and $\mathsf{B} \equiv g^b \pmod p$
- compute $g^{ab} \pmod p$.

The oracle returns the quantity $(c_1^a)^{-1} \cdot c_2 \pmod p$.

What values should one choose for c_1 and c_2?

$c_1 = \mathsf{B} \equiv g^b$ and $c_2 = 1$ would work because the oracle would return $(g^{ab})^{-1}$ and Cupid would compute the inverse of this, i.e. g^{ab}. But we exclude $c_2 = 1$, the oracle most certainly should dismiss it !

Cupid could choose an arbitrary c_2 and send the oracle the values c_2, the public key A and the ciphertext (B, c_2), in other words, he would try a chosen text attack. The oracle would return the supposed plaintext

$$\mathsf{m} \equiv (c_1^a)^{-1} \cdot c_2 \equiv (\mathsf{B}^a)^{-1} \cdot c_2 \equiv \left(g^{ab}\right)^{-1} \cdot c_2 \pmod p$$

and Cupid would be in business: $g^{ab} \equiv \mathsf{m}^{-1} \cdot c_2$.

The conclusion must be that DHP $\preccurlyeq$ ElGamal. Furthermore, the DHP could be solved without knowledge of either a or b so this is the solution to the DHP but not to the DLP.

We have shown that assuming that the DHP is hard the ElGamal cryptosystem is secure and quite specifically it is secure to chosen ciphertext attacks.

Elliptic curve ElGamal cryptosystems

We choose a prime p, an elliptic curve E and a point $\mathsf{P} \in \mathsf{E}(\mathbb{F}_p)$. All this will be made public. Then Arabella chooses her secret key n_A and reveals the public key $\mathsf{Q_A} = n_\mathsf{A}\mathsf{P}$. Beau wants to send her the message $\mathsf{M} \in \mathsf{E}(\mathbb{F}_p)$. He chooses the ephemeral key, an integer k, computes $\mathsf{C}_1 = k\mathsf{P}$ and $\mathsf{C}_2 = \mathsf{M} + k\mathsf{Q_A}$ and sends Arabella $(\mathsf{C}_1, \mathsf{C}_2)$, two points.

Arabella now decrypts:

$$\mathsf{C}_2 - n_\mathsf{A}\mathsf{C}_1 = (\mathsf{M} + k\mathsf{Q_A}) - n_\mathsf{A}(k\mathsf{P}) = \mathsf{M} + kn_\mathsf{A}\mathsf{P} - kn_\mathsf{A}\mathsf{P} = \mathsf{M}.$$

She has got the message!

All this is very well but there are a couple of practical issues/difficulties. First, there is no obvious way to attach plaintext messages to points on $\mathsf{E}(\mathbb{F}_p)$. Second, the elliptic curve ElGamal cryptosystem has $4-$to -1 message expansion whereas the standard ElGamal cryptosystem has $2-$to -1

message expansion. This is due precisely to the fact that $(\mathsf{C}_1, \mathsf{C}_2)$ is a pair of points on the elliptic curve. Hasse's Theorem (see Appendix) says that there are approximately p different points in $\mathsf{E}(\mathbb{F}_p)$, that is approximately p different plaintexts, so we might have a scarcity problem.

We could, of course, avoid the problem of such large expansion by sending only the $x-$coordinate. But, at decryption, you need whole points because if you choose the "wrong" $y-$coordinate you get $\mathsf{C}_2+n_\mathsf{A}\mathsf{C}_1$ instead of $\mathsf{C}_2-n_\mathsf{A}\mathsf{C}_1$, and these are very different points indeed ! You might circumvent the problem by sending an

$$\text{extra bit} = \begin{cases} 0 & , 0 \leq y < \frac{1}{2}p \\ 1 & , \frac{1}{2}p \leq y < p \end{cases} .$$

You might ask: why does this work? If Beau sends $x = \gamma$ then Arabella computes $\gamma^3 + a\gamma + b = \delta$ and then tries to solve the equation $y^2 = \delta$. A solution must exist because Beau sends the $x-$coordinate of a point on the elliptic curve.

Case 1. $\delta = 0$.
The solution is unique, $y = 0$, and the point is unique too, $(\gamma, 0)$.

Case 2. $\delta > 0$.
(This case is enough because we shall eventually compute modulo p). Solving we get the solution y_1 and assume (without loss of generality) that $0 < y_1 < \frac{p}{2}$. We know that $y_2 = -y_1 \equiv p - y_1$ is the other solution. Suppose that $0 < y_2 < \frac{p}{2}$ too. This is equivalent to $0 < p - y_1 < \frac{p}{2}$ which entails the inequality $y_1 > \frac{p}{2}$ contradicting the assumption.

Nota bene. $y_1 = y_2 = \frac{p}{2}$ is not possible because the equation $y^2 = \delta > 0$ has two distinct solutions.

So Beau would need two extra bits for the two points C_1 and C_2. This is called *point compression*.

Example. Arabella and Beau decide to use the prime $p = 1123$ and the elliptic curve $y^2 = x^3 + 54x + 87$. Beau sends Arabella the $x-$coordinate $x = 278$ and the bit $\beta = 0$. Arabella computes $278^3 + 54 \cdot 278 + 87 \equiv 216 \pmod{1123}$.

Now she must solve the equation $y^2 = 216$. Since $p = 1123 \equiv 3 \pmod 4$

$$y_1 = 216^{1124/4} = 216^{281} \equiv 487 \pmod{1123}$$

will be a solution. The other solution will obviously be $y_2 \equiv -487 \equiv 636 \pmod{1123}$.

$\beta = 0$ indicates that Beau sent her the point $(278, 487)$ since $487 < 561.5 = \frac{p}{2}$.

$\beta = 1$ would have given the point $(278, 636)$ as $636 > 561.5 = \frac{p}{2}$.

2.3 Digital signatures

ElGamal

The El Gamal digital signature scheme was presented in 1985.

Arabella chooses a (large) prime p and a primitive root $g \pmod{p}$ and then she chooses a secret signing exponent s and computes the verification exponent $v \equiv g^s \pmod{p}$.

(v, p, g) is Arabella's public verification key.

Suppose she has the document $1 < \mathsf{D} < p$. She chooses now a random number e, $1 < e < p$, the ephemeral key, and computes

$$\mathsf{S}_1 \equiv g^e \pmod{p}$$

$$\mathsf{S}_2 \equiv (\mathsf{D} - s\mathsf{S}_1)e^{-1} \pmod{(p-1)}$$

Caveat! e^{-1} is to be computed modulo $(p-1)$.

Arabella's digital signature on D will be the pair $(\mathsf{S}_1, \mathsf{S}_2)$.

Beau verifies:

$$v^{\mathsf{S}_1} \cdot \mathsf{S}_1^{\mathsf{S}_2} \equiv g^{s\mathsf{S}_1} \cdot g^{e\mathsf{S}_2} \equiv$$

$$g^{s\mathsf{S}_1 + e\mathsf{S}_2} \equiv g^{s\mathsf{S}_1 + e(\mathsf{D} - s\mathsf{S}_1)e^{-1}} \equiv$$

$$g^{s\mathsf{S}_1 + \mathsf{D} - s\mathsf{S}_1} \equiv g^{\mathsf{D}} \pmod{p}$$

The verification algorithm returns TRUE.

Nota bene. We know that $g^{p-1} \equiv 1 \pmod{p}$ so, in the expression $g^{\mathsf{S}_2} \pmod{p}$ we may replace S_2 by any other number congruent to $\mathsf{S}_2 \pmod{(p-1)}$.

Example. Arabella chooses a prime

$$p = 70843$$

and a primitive root

$$g \equiv 2 \pmod{70843}$$

She selects her secret signing key $s = 317$ and computes her public verification key associated to the pair $(p, g) = (70843,7)$:

$$v \equiv g^s \equiv 2^{317} \equiv 13219 \pmod{70843}$$

Suppose she wants to sign the document $\mathsf{D} = 502$. She chooses a random number $e = 427$ (the ephemeral key) in the range
$1 < e < 70843$ with inverse $e^{-1} = 65533 \pmod{70842}$. This might cause trouble, but she will simply choose e an odd number so it will be invertible modulo $(p - 1)$. Then she computes the values:

$$\mathsf{S}_1 \equiv g^e \equiv 2^{427} \equiv 63851 \ (\bmod\ 70843)$$

$$\mathsf{S}_2 \equiv (\mathsf{D} - s\mathsf{S}_1)e^{-1} \equiv (502 - 317 \cdot 63851) \cdot 65533 \equiv 12657 \ (\bmod\ 70843).$$

Her digital signature on the document D will be $\mathsf{S} = (\mathsf{S}_1,\mathsf{S}_2) = (63851, 12657)$.
Beau receives the document and verifies the signature. He computes and checks:

$$v^{\mathsf{S}_1}\mathsf{S}_1^{\mathsf{S}_2} \equiv 13219^{63851} \cdot 63851^{12657} \equiv$$

$$7373 \equiv 2^{502} \equiv g^{\mathsf{D}} (\bmod\ 11807)$$

S is the signature of his sweetheart. Bliss!

All Cupid needs to do is to solve the DLP $g^s \equiv v \pmod{p}$.
But is this the only way to break the scheme?
Given v and g^{D} Cupid must find integers x and y such that $v^x x^y \equiv g^{\mathsf{D}} \pmod{p}$. Using discrete logarithms we get

$$x\log_g v + y\log_g x \equiv \mathsf{D} \pmod{(p-1)}$$

If Cupid can solve the DLP then he can take an arbitrary value for x and solve the above equation for y. This is the only known method to do it (at present!). So Cupid must solve the DLP.

DSA (Digital Signature Algorithm)

In 1991 a modified version of the ElGamal digital signature scheme was proposed allowing shorter signatures, the DSA. This was officially published in 1994 as a national *Digital Signature Standard* (DSS). (For all this see NBS–DES. Digital Signature Standard (DSS). FIPS Publication 186-2, National Bureau of Standards, 199, as quoted by Hoffstein, Pipher & Silverman, section 7.3).

The idea is to work in a subgroup of $\mathbb{F}_p^\star$ of prime order q. Arabella chooses two primes p and q with $p \equiv 1 \pmod q$. (Usually cryptographers take $2^{1000} < p < 2^{2000}$ and $2^{160} < q < 2^{320}$). Then she chooses an element $g \in \mathbb{F}_p^\star$ of order q, e.g. $g \equiv g_1^{\frac{p-1}{q}}$ for a primitive root $g_1 \in \mathbb{F}_p^\star$. She goes on and chooses a secret exponent s and computes $v \equiv g^s \pmod p$.

(p, q, g) will be her public verification key. The document is D as before. She chooses the ephemeral key e as before in the ElGamal version but now computes:

$$\mathsf{S}_1 \equiv (g^e \pmod p) \pmod q$$

$$\mathsf{S}_2 \equiv (\mathsf{D}+s\mathsf{S}_1)e^{-1} \pmod q.$$

$\mathsf{S} = (\mathsf{S}_1, \mathsf{S}_2)$ will be Arabella's digital signature on the document D, two numbers modulo q.

Beau verifies by computing

$$\mathsf{V}_1 \equiv \mathsf{D}\mathsf{S}_2^{-1} \pmod q$$

$$\mathsf{V}_2 \equiv \mathsf{S}_1\mathsf{S}_2^{-1} \pmod q$$

and checking that

$$g^{\mathsf{V}_1} v^{\mathsf{V}_2} \equiv g^{\mathsf{D}\mathsf{S}_2^{-1}} g^{s\mathsf{S}_1\cdot\mathsf{S}_2^{-1}} \equiv g^{(\mathsf{D}+s\mathsf{S}_1)\mathsf{S}_2^{-1}} \equiv g^e \pmod p.$$

Then we have that $(g^{\mathsf{V}_1} v^{\mathsf{V}_2} \pmod p) \overset{q}{\equiv} (g^e \pmod p) \overset{q}{\equiv} \mathsf{S}_1$ and everything is as it should be.

Example. Arabella chooses two primes

$$p = 70843$$

$$q = 11807$$

$$p \equiv 1 \pmod{q}.$$

She finds then a primitive root $g_1 = 2 \in \mathbb{F}_p^\star$ and computes an element

$$g = 2^{\frac{p-1}{q}} = 64 \text{ of order } 11807 \text{ in } \mathbb{F}_p^\star .$$

Then she chooses a secret exponent $s = 317$ and computes her public verification key associated to the triple $(p, q, g) = (70843, 11807, 64)$:

$$v \equiv g^s \equiv 64^{317} \equiv 4386 \pmod{70843}$$

Suppose she wants to sign the document $\mathsf{D} = 502$. She chooses a random number $e = 427$ (the ephemeral key) in the range $1 \leq e < 11807$ with inverse $e^{-1} = 6498 \pmod{11807}$, and computes:

$$\mathsf{S}_1 \equiv (g^e = 64^{427} \overset{p}{\equiv} 70605\ ) \equiv 11570\ (\ \mathrm{mod}\ 11807)$$

$$\mathsf{S}_2 \equiv (\mathsf{D} + s\mathsf{S}_1)e^{-1} \equiv (502{+}317{\cdot}11570){\cdot}6498 \equiv 10858\ (\ \mathrm{mod}\ 11807).$$

Her digital signature on the document D will be $\mathsf{S} = (\mathsf{S}_1, \mathsf{S}_2) = (11570, 10858)$.

Beau receives the document and verifies the signature. First he computes:

$$\mathsf{V}_1 \equiv \mathsf{D}\mathsf{S}_2^{-1} \equiv 502{\cdot}10858^{-1} \equiv$$

$$502{\cdot}7552 \equiv 1057\ (\ \mathrm{mod}\ 11807)$$

$$\mathsf{V}_2 \equiv \mathsf{S}_1\mathsf{S}_2^{-1} \equiv 11570{\cdot}10858^{-1} \equiv$$

$$11570{\cdot}7552 \equiv 4840\ (\ \mathrm{mod}\ 11807)$$

and checks

$$(g^{\mathsf{V}_1} v^{\mathsf{V}_2} = 64^{1057}{\cdot}4386^{4840} \overset{p}{\equiv} 70605) \equiv 11570\ (\ \mathrm{mod}\ 11807).$$

S is her signature. Euphoria ensues!

ECDSA

The DSA works just as well in other groups, $\mathsf{E}(\mathbb{F}_p)$ in particular, so we have the elliptic curve version ECDSA:

1. A trusted party chooses a finite field $\mathbb{F}_p$, an elliptic curve $\mathsf{E}(\mathbb{F}_p)$, and a point $\mathsf{G} = (x, y) \,\epsilon\, \mathsf{E}(\mathbb{F}_p)$ of large prime order q.
2. Arabella chooses a secret signing key s, $1 < s < q - 1$. She computes $\mathsf{V} = s\mathsf{G} \,\epsilon\, \mathsf{E}(\mathbb{F}_p)$ and publishes this as the verification key.
3. She then chooses a document $d \bmod q$, an ephemeral key $e \pmod q$, computes $e\mathsf{G} \,\epsilon\, \mathsf{E}(\mathbb{F}_p)$ and
   a) $s_1 \equiv xe\mathsf{G} \pmod q$
   b) $s_2 \equiv (d + ss_1)e^{-1} \pmod q$
4. She publishes the signature (s_1, s_2).
5. Beau computes
   a) $v_1 \equiv ds_2^{-1} \pmod q$
   b) $v_2 \equiv s_1 s_2^{-1} \pmod q$
   c) $v_1\mathsf{G} + v_2\mathsf{V} \,\epsilon\, \mathsf{E}(\mathbb{F}_p)$
6. He finally verifies that $x(v_1\mathsf{G} + v_2\mathsf{V}) \equiv s_1 \pmod q$

Let us verify the last step modulo q :

$$x(v_1\mathsf{G} + v_2\mathsf{V}) \equiv$$

$$x(ds_2^{-1}\mathsf{G} + s_1 s_2^{-1} s\mathsf{G}) \equiv$$

$$x(d + s_1 s)s_2^{-1}\mathsf{G} \equiv$$

$$x\, e s_2 s_2^{-1}\mathsf{G} \equiv$$

$$x\, e\mathsf{G} \equiv s_1$$

2.4 The Massey – Omura public key cryptosystem

Arabella chooses as usual a prime p and makes it public. Then she chooses a secret key e_{A} such that $0 < e_{\mathsf{A}} < p-1$ and $\gcd(e_{\mathsf{A}},\ p-1) = 1$, thus making sure that $d_{\mathsf{A}} \equiv e_{\mathsf{A}}^{-1} \pmod{(p-1)}$ exists. e_{A} is Arabella's encryption key and d_{A} is her decryption key.

She sends Beau the message m encrypted by $\mathsf{c} \equiv \mathsf{m}^{e_{\mathsf{A}}} \pmod{(p-1)}$. Beau cannot do anything because he does not know d_{A} but he chooses himself his own encryption and decryption keys, e_{B} and d_{B}, $e_{\mathsf{B}} d_{\mathsf{B}} \equiv 1 \pmod{(p-1)}$, and sends back to Arabella the message $\mathsf{m}^{e_{\mathsf{A}} e_{\mathsf{B}}}$ which she then transforms into

$$\mathsf{m}^{e_{\mathsf{A}} e_{\mathsf{B}} d_{\mathsf{A}}} \equiv \mathsf{m}^{e_{\mathsf{B}}}$$

which she sends back to Beau who finally can decrypt it by means of his decryption key d_{B}:

$$\mathsf{m}^{e_{\mathsf{B}} d_{\mathsf{B}}} \equiv \mathsf{m}.$$

This cryptosystem relies again on the difficulty of the DLP. Even this system has its obvious elliptic curve version. We have a publicly known elliptic curve $\mathsf{E}(\mathbb{F}_p)$, p being a presumably large prime and we have computed $\#\ \mathsf{E}(\mathbb{F}_p) = \mathsf{N}$ which, of course, is public knowledge.

Arabella chooses her secret keys, e_{A} and d_{A}, and Beau his, e_{B} and d_{B}, all of these modulo N. Arabella wants to send Beau the message/point P, so she encrypts $\mathsf{c} = e_{\mathsf{A}}\mathsf{P}$ and sends this. Beau computes $e_{\mathsf{B}} e_{\mathsf{A}}\mathsf{P}$ and sends it back to Arabella who returns to him $d_{\mathsf{A}} e_{\mathsf{B}} e_{\mathsf{A}}\mathsf{P} = e_{\mathsf{B}}\mathsf{P}$ which he is now able to decrypt by $d_{\mathsf{B}} e_{\mathsf{B}}\mathsf{P} \equiv \mathsf{P}$.

If Cupid can solve the ECDLP then he is in the game.

But, apart from this, the system involves a lot of "traffic" which can jeopardize its security. Let us reconsider. Arabella sends Beau $\mathsf{m}^{e_{\mathsf{A}}}$ or $e_{\mathsf{A}}\mathsf{P}$. Cupid intercepts this message and returns himself $\mathsf{m}^{e_{\mathsf{A}} e_{\mathsf{C}}}$ or $e_{\mathsf{C}} e_{\mathsf{A}}\mathsf{P}$ to Arabella, pretending to be Beau. She now sends back, to whoever intercepts, $\mathsf{m}^{e_{\mathsf{A}} e_{\mathsf{C}} d_{\mathsf{A}}} = \mathsf{m}^{e_{\mathsf{C}}}$ or $d_{\mathsf{A}} e_{\mathsf{C}} e_{\mathsf{A}}\mathsf{P} = e_{\mathsf{C}}\mathsf{P}$ which Cupid can decrypt: $\mathsf{m}^{e_{\mathsf{C}} d_{\mathsf{C}}} \equiv \mathsf{m}$ or $d_{\mathsf{C}} e_{\mathsf{C}}\mathsf{P} = \mathsf{P}$.

Obviously there is a serious flaw in the system which must be rectified by some scheme of authentication or digital signature.

2.5 Applications of the Weil pairing

Tripartite Diffie–Hellman key exchange

Arabella and Beau want to include even Daphne in their circle of secrets. They agree all three on an elliptic curve E and a point $\mathsf{P} \epsilon \mathsf{E}(\mathbb{F}_q)[l]$ of prime order provided there exists an $l-$distortion map for P. Let $\widehat{e_l}$ be the associated modified Weil pairing. (See Appendix).

Each one of our heroes chooses a personal secret integer n_{A}, n_{B}, n_D, respectively, and computes:

- Arabella: $\mathsf{Q_A}= n_{\mathsf{A}}\mathsf{P}$
- Beau: $\mathsf{Q_B}= n_{\mathsf{B}}\mathsf{P}$
- Daphne: $\mathsf{Q_D}= n_{\mathsf{D}}\mathsf{P}$

and they all publish the respective values.

Arabella computes now $\widehat{e_l}\ (\mathsf{Q_B}, \mathsf{Q_D})^{n_{\mathsf{A}}}$, where, as we know, $\mathsf{Q_B}$ and $\mathsf{Q_D}$ are multiples of P. Arabella does not know which these multiples are but bilinearity gives:

- Arabella: $\widehat{e_l}\ (\mathsf{Q_B}, \mathsf{Q_D})^{n_{\mathsf{A}}}= \widehat{e_l}\ (n_{\mathsf{B}}\mathsf{P}, n_{\mathsf{D}}\mathsf{P})^{n_{\mathsf{A}}} = \widehat{e_l}\ (\mathsf{P}, \mathsf{P})^{n_{\mathsf{A}}n_{\mathsf{B}}n_{\mathsf{D}}}$.
- Beau: $\widehat{e_l}\ (\mathsf{Q_A}, \mathsf{Q_D})^{n_{\mathsf{B}}}= \widehat{e_l}\ (n_{\mathsf{A}}\mathsf{P}, n_{\mathsf{D}}\mathsf{P})^{n_{\mathsf{B}}} = \widehat{e_l}\ (\mathsf{P}, \mathsf{P})^{n_{\mathsf{A}}n_{\mathsf{B}}n_{\mathsf{D}}}$.
- Daphne: $\widehat{e_l}\ (\mathsf{Q_A}, \mathsf{Q_B})^{n_{\mathsf{D}}}= \widehat{e_l}\ (n_{\mathsf{A}}\mathsf{P}, n_{\mathsf{B}}\mathsf{P})^{n_{\mathsf{A}}} = \widehat{e_l}\ (\mathsf{P}, \mathsf{P})^{n_{\mathsf{A}}n_{\mathsf{B}}n_{\mathsf{D}}}$.

So they all share the same secret value $\widehat{e_l}\ (\mathsf{P}, \mathsf{P})^{n_{\mathsf{A}}n_{\mathsf{B}}n_{\mathsf{D}}}$. If Cupid can solve the ECDLP then he can break this tripartite Diffie $-$ Hellman key exchange. He will then be able to recover at least one of the integers n_{A}, n_{B} or n_D and that is enough. He can, of course, compute $\widehat{e_l}\ (\mathsf{P}, \mathsf{P})$ and $\widehat{e_l}\ (\mathsf{Q_A}, \mathsf{P}) = \widehat{e_l}\ (n_{\mathsf{A}}\mathsf{P}, \mathsf{P}) = \widehat{e_l}\ (\mathsf{P}, \mathsf{P})^{n_{\mathsf{A}}}$, thus he could recover n_{A} if he could solve the DLP in $\mathbb{F}_q$.

We draw the conclusion that tripartite Diffie $-$ Hellman key exchange is vulnerable to the classical DLP in a subgroup of $\mathbb{F}_q^\star$ of order l. According to Hoffstein, Pipher & Silverman there are subexponential algorithms for that so one should use larger fields for tripartite key exchange than for bipartite.

Id – based public key cryptosystems

Suppose Arabella wants to use her e-mail address as her identity-based public key. She needs of course some private key which she uses for decryption and that key must also be used in an essential way in encryption. Assume that there is some higher authority, say Zeus, who publishes a master public key Zeus ^{Pub} and keeps secret a private key Zeus ^{Priv}. Beau will use Zeus ^{Pub} and Arabella's id−based public key Arabella^{Pub} to send messages to her. Zeus, the master of all, creates out of Arabella^{Pub} and Zeus ^{Priv} a private key Arabella^{Priv} for Arabella who then uses it to decrypt messages from Beau. It goes without saying that the omnipotent and omniscient Zeus can keep track of all the private keys he has created and assigned, otherwise havock ensues. It is furthermore necessary and essential that not Cupid nor any other party be able to recover Zeus ^{Priv} from any number of keys that they are allotted by Zeus on request.

These ideas were initially described by Shamir in 1984 and such an id−based system was created by Boneh and Franklin in 2001. The system uses pairings on elliptic curves. I shall present the basic ingredients but abstain from any computations.

- Zeus, the master authority, selects a finite field $\mathbb{F}_q$, an elliptic curve E and a point $\mathsf{P} \,\epsilon\, \mathsf{E}(\mathbb{F}_q)[l]$ of prime order such that there is an $l-$distortion map for P with $\hat{e}_l$ the associated modified Weil pairing.
- Zeus publishes to functions

$$\mathsf{H}_1 : \{user\mathsf{ID}s\} \longrightarrow \mathsf{E}(\mathbb{F}_q)$$

$$\mathsf{H}_2 : \mathbb{F}_q^{\star} \longrightarrow \mathcal{M} = \{\text{the set of plaintexts}\}$$

- Zeus creates his master key $\mathsf{P}^{\mathsf{Zeus}} = s\mathsf{P} \,\epsilon\, \mathsf{E}(\mathbb{F}_q)$, where s is Zeus' master private key, an integer, and $\mathsf{P}^{\mathsf{Zeus}}$ becomes his master public key.
- Beau wants to send Arabella a message $\mathsf{M} \,\epsilon\, \mathcal{M}$ using her id−based public key Arabella^{Pub}. He uses this public key and the hash function H_1 to compute $\mathsf{P}^{\mathsf{Arabella}} = \mathsf{H}_1(\text{Arabella}^{Pub}) \,\epsilon\, \mathsf{E}(\mathbb{F}_q)$.
- Beau chooses a random number (ephemeral key) $\mathsf{0} \neq r \pmod{(q-\mathsf{1})}$ and computes $\mathsf{C}_1 = r\mathsf{P}$ and $\mathsf{C}_2 = \mathsf{M}$ xor $\mathsf{H}_2(\hat{e}_l(\mathsf{P}^{\mathsf{Arabella}}, \mathsf{P}^{\mathsf{Zeus}})^r)$. The ciphertext becomes $\mathsf{C} = (\mathsf{C}_1, \mathsf{C}_2)$.
- Arabella requests from Zeus her private key Arabella^{Priv} , associated to Arabella^{Pub}, and receives from Him $\mathsf{Q}^{\mathsf{Arabella}} = s\mathsf{P}^{\mathsf{Arabella}} = s\mathsf{H}_1(\text{Arabella}^{Pub}) \,\epsilon\, \mathsf{E}(\mathbb{F}_q)$.

- Arabella decrypts the message from Beau in two stages. First she computes:

$$\widehat{e}_l(\mathsf{Q}^{\mathsf{Arabella}}, \mathsf{C}_1) = \hat{e}_l(s\mathsf{P}^{\mathsf{Arabella}}, r\mathsf{P}) = \hat{e}_l(\mathsf{P}^{\mathsf{Arabella}}, \mathsf{P})^{rs} =$$

$$\hat{e}_l(\mathsf{P}^{\mathsf{Arabella}}, s\mathsf{P})^r = \hat{e}_l(\mathsf{P}^{\mathsf{Arabella}}, \mathsf{P}^{\mathsf{Zeus}})^r$$

which is the quantity that Beau used to create C_2. Then she recovers the plaintext by:

$$\mathsf{C}_2 \text{ xor } \mathsf{H}_2(\widehat{e}_l(\mathsf{Q}^{\mathsf{Arabella}}, \mathsf{C}_1)) =$$

$$(\mathsf{M} \text{ xor } \mathsf{H}_2(\hat{e}_l(\mathsf{P}^{\mathsf{Arabella}}, \mathsf{P}^{\mathsf{Zeus}})^r)) \text{ xor } \mathsf{H}_2(\hat{e}_l(\mathsf{P}^{\mathsf{Arabella}}, \mathsf{P}^{\mathsf{Zeus}})^r) = \mathsf{M},$$

since M xor L xor $\mathsf{L} = \mathsf{M}$ for any bit strings M and L.

3 *Elliptic Curve Cryptanalysis*

" I sent a message to the fish.

I told them 'This is what I wish.'

The little fishes of the sea,

They sent an answer back to me.

The little fishes' answer was

' we cannot do it, Sir, because —— ' "

(From *Through the Looking−Glass*, ch.1 *Looking − Glass House*,
by Lewis Carroll)

3.1 The discrete logarithm problem

DLP

We shall consider the finite field $\mathbb{F}_p \cong \mathbb{Z}_p$ and its multiplicative subgroup generated by a primitive element g, thus $\mathbb{F}_p^\star = < g > = \{1, g, g,^2 .....,g^{p-2}\}$.

Let $h \neq 0$ be an element of $\mathbb{F}_p^\star$. The discrete logarithm problem (DLP) is to find an exponent $n \in \mathbb{N}$ such that

$$g^n \equiv h \pmod{p} \ (\star)$$

The smallest such n is called the *discrete logarithm* of h to the *base* g and we write

$$n = log_g h$$

An older terminology was the *index* of h to the *base* g with notation $n = ind_g h$ but since our n closely resembles the logarithm of calculus one can understand the change in terminology.

If n is a solution to $(\star)$ then so is $n + k(p-1)$ because

$$g^{n+k(p-1)} = g^n \cdot (g^{p-1})^k \equiv h \cdot 1^k = h,$$

since $g^{p-1} \equiv 1$ by Fermat's Little Theorem.

I shall show that we have a group homomorphism

$$log_g \colon \mathbb{F}_p^\star \longrightarrow \mathbb{Z}_{p-1}$$

Suppose $log_g h = a$ and $log_g h = b$. This means that $g^a \equiv h \equiv g^b \pmod{p}$, so $g^{a-b} \equiv 1 \pmod{p}$, but by Fermat'Little Theorem, we know that $g^{p-1} \equiv 1 \pmod{p}$ and $p-1$ is the smallest integer with this property. Hence we have that $a - b = k(p-1)$, $k \geq 1$, or equivalently , $a \equiv b \bmod (p-1)$.

$\because$ log_g is well-defined.

Suppose

$$\begin{cases} log_g a = q & \Leftrightarrow g^q \equiv a \\ log_g b = r & \Leftrightarrow g^r \equiv b \\ log_g ab = s & \Leftrightarrow g^s \equiv ab \end{cases}$$

We have:

$$g^s \equiv ab \equiv g^q \cdot g^r = g^{q+r}$$

This entails

$$q + r \equiv s \bmod (p - 1)$$

$$log_g a + log_g b \equiv log_g ab \bmod (p - 1)$$

Furthermore

$$log_g 1 \equiv 0 \bmod (p - 1) \text{ since } g^{p-1} \equiv 1 \bmod p.$$

$\because log_g$ is a group homomorphism

$log_g a \equiv log_g b \bmod (p - 1)$ means that $a \equiv g^n \equiv b \bmod p$

$\because log_g$ is injective

$|\mathbb{F}_p^\star| = p - 1$

$\because log_g$ is surjective

$\because log_g$ is a group isomorphism.

All the usual logarithm laws are valid.We have already shown that:

$$log_g a + log_g b \equiv log_g ab \bmod (p - 1)$$

Now:

$$log_g a \equiv q \pmod{(p - 1)}$$

$$\Updownarrow$$

$$g^q \equiv a \pmod p$$

$$\Updownarrow$$

$$a^n \equiv (g^q)^n \equiv g^{nq} \pmod p.$$

The second law follows:

$$log_g a^n \equiv nq \equiv n log_g a$$

Replace b by b^{-1} in the first law and then use the second law. We get the third law:

$$log_g \frac{a}{b} = log_g ab^{-1} \equiv log_g a - log_g b \bmod (p-1)$$

How difficult is the DLP? Let us determine $log_2 13 \bmod 23$. We must solve the congruence $2^x \equiv 13 \bmod 23$.We shall simply compute by hand !

$$2^2 \equiv 4,\ 2^3 \equiv 8,\ 2^4 \equiv 16,\ 2^5 \equiv 9,\ 2^6 \equiv 18,\ 2^7 \equiv 13.$$

The answer is: $log_2 13 \equiv 7 \bmod 23$.

Let us now determine $log_{627} 608 \bmod 941$. We must solve the congruence $627^x \equiv 608 \bmod 941$.We shall definitely need a computer !

The answer will be: $log_{627} 608 \equiv 18 \bmod 941$.

It is clear that with increasing values we need more computations and the question is what is the order of computation steps needed.

The task will be to compute $g^A \pmod N$ for some large integers A and N. The brute force method would be to set

$$g_1 \equiv g \pmod N$$

and then to compute

$$\begin{cases} g_2 \equiv g g_1 & (\mathrm{mod} N) \\ g_3 \equiv g g_2 & (\mathrm{mod} N) \\ \qquad\vdots & \\ g_A \equiv g g_{A-1} & (\mathrm{mod} N) \end{cases}$$

For large A this is pure and simple nightmare! Let us look at an example. Compute $3^{319} \pmod{1000}$. Consider the binary expansion of 319:

$$319 = 1+2+2^2+2^3+2^4+2^5+2^8$$

Thus

$$3^{319} = 3^{1+2+2^2+2^3+2^4+2^5+2^8} = 3 \cdot 3^2 \cdot 3^{2^2} \cdot 3^{2^3} \cdot 3^{2^4} \cdot 3^{2^5} \cdot 3^{2^8}.$$

It is not very difficult to compute the sequence
$3, 3^2, 3^{2^2}, 3^{2^3}, 3^{2^4}, \ldots\ldots\ldots$, where each number is the square of its predecessor since $\left(3^{2^n}\right)^2 = 3^{2^n \cdot 2} = 3^{2^{n+1}}$. In our case we need 8 squarings from 3 to 3^{2^8} thus needing 8 multiplications, and then:

$$3^{319} = 3^{1+2+2^2+2^3+2^4+2^5+2^8} = 3 \cdot 3^2 \cdot 3^{2^2} \cdot 3^{2^3} \cdot 3^{2^4} \cdot 3^{2^5} \cdot 3^{2^8}$$

$$\equiv 3 \cdot 9 \cdot 81 \cdot 561 \cdot 721 \cdot 841 \cdot 521 \equiv 467 \pmod{1000}.$$

This shows that we need 6 more multiplications, a total of 14 multiplications. Since we compute mod 1000 we need only store the last 3 digits of every computation, so the storage room is not alarming!

The *Square − and − Multiply Algorithm* :

Step 1. Compute the binary expansion of A:

$$A = A_0 + A_1 \cdot 2 + A_2 \cdot 2^2 + \ldots\ldots\ldots\ldots + A_r \cdot 2^r$$

$$A_i \in \{0, 1\},\ i = 1, 2, \ldots\ldots\ldots\ldots\ldots, r,\ A_r = 1$$

Step 2. Compute $g^{2^i} \pmod{N}$, $0 \leq i \leq r$:

$$\begin{cases} a_0 \equiv g & (\mathrm{mod} N) \\ a_1 \equiv a_0^2 \equiv g^2 & (\mathrm{mod} N) \\ a_2 \equiv a_1^2 \equiv g^{2^2} & (\mathrm{mod} N) \\ \vdots & \vdots \\ a_r \equiv a_{r-1}^2 \equiv g^{2^r} & (\mathrm{mod} N) \end{cases}$$

We need r squarings.
Step 3. Compute $g^A \pmod{N}$:

$$g^A = g^{A_0 + A_1 \cdot 2 + A_2 \cdot 2^2 + \ldots\ldots\ldots\ldots + A_r \cdot 2^r} \equiv$$

$$g^{A_0} \cdot (g^2)^{A_1} \cdot \left(g^{2^2}\right)^{A_2} \cdot \ldots\ldots\ldots\ldots \cdot \left(g^{2^r}\right)^{A_r} \equiv$$

$$a_0^{A_0} \cdot a_1^{A_1} \cdot a_2^{A_2} \cdot \ldots\ldots\ldots\ldots \cdot a_r^{A_r} \pmod{N}.$$

All the a_i have already been computed in Step 2 so multiply together mod N all such $a_i^{A_i}$ where $A_i \neq 0$. This requires at most r multiplications giving a total of, at most, 2r multiplications.
$2^r \leq A \Rightarrow log_2 2^r \leq log_2 A \Rightarrow r \leq log_2 A$ (usual logarithms)
$\because 2r \leq 2 log_2 A$
This shows that the computation steps are of the order $\mathcal{O}(log_2 A)$.

ECDLP

Let us look at the discrete logarithm problem over the group $\mathsf{E}(\mathbb{F}_p)$. We shall try to solve the same problem as before only, this time , for elements/points in the additive group/elliptic curve $\mathsf{E}(\mathbb{F}_p)$. (See Appendix). In other words, given an elliptic curve over the finite field $\mathbb{F}_p$ with points P and Q on $\mathsf{E}(\mathbb{F}_p)$ find an integer $n \epsilon \mathbb{N}$ such that

$$n\mathsf{P} = \mathsf{Q}$$

and, in analogy with the DLP, we shall write $n = log_{\mathsf{P}}\mathsf{Q}$ for the solution.

Already here we run into difficulties because Q may not be a multiple of P in which case $log_{\mathsf{P}}\mathsf{Q}$ is not defined. And yet, if we want to encrypt messages we start with a public point P and a secret/private integer n and compute $\mathsf{Q}= n\mathsf{P}$ so $log_{\mathsf{P}}\mathsf{Q}$ will exist but its value will be secret, indeed $log_{\mathsf{P}}\mathsf{Q}$ $= n$ is our secret integer.

Secondly, if there exists a value n such that $n\mathsf{P} = \mathsf{Q}$, then there exist many.

We know that there exists $\mathsf{0} \neq s \epsilon \mathbb{N}$ such that $s\mathsf{P} = \mathcal{O}$ since $\mathsf{E}(\mathbb{F}_p)$ is a finite group, so every element/point of $\mathsf{E}(\mathbb{F}_p)$ has finite order. Therefore the points $\mathsf{2P}$, $\mathsf{3P}$, and so on, cannot all be distinct.

$\exists k > j \epsilon \mathbb{N}$ such that $k\mathsf{P} = j\mathsf{P}$. Then we may take $r = k - j$ and the smallest such r is called the *order* of P in $\mathsf{E}(\mathbb{F}_p)$.

The consequence of this is that $log_{\mathsf{P}}\mathsf{Q} \epsilon \mathbb{Z}_r$ so we define a map

$$log_{\mathsf{P}} : \mathsf{E}(\mathbb{F}_p) \longrightarrow \mathbb{Z}_r.$$

Suppose

$$log_{\mathsf{P}}\mathsf{Q}_1 \equiv l \Leftrightarrow \mathsf{Q}_1 = l\mathsf{P}$$

$$log_{\mathsf{P}}\mathsf{Q}_2 \equiv m \Leftrightarrow \mathsf{Q}_2 = m\mathsf{P}$$

$$log_{\mathsf{P}}(\mathsf{Q}_1 + \mathsf{Q}_2) \equiv n \Leftrightarrow \mathsf{Q}_1 + \mathsf{Q}_2 = n\mathsf{P}$$

Then

$$\mathsf{Q}_1 + \mathsf{Q}_2 = l\mathsf{P} + m\mathsf{P} = (l + m)\mathsf{P} = n\mathsf{P}, \text{ thus } n \equiv l + m \pmod{r}$$

$\because log_{\mathsf{P}}(\mathsf{Q}_1 + \mathsf{Q}_2) = log_{\mathsf{P}}\mathsf{Q}_1 + log_{\mathsf{P}}\mathsf{Q}_2$

We know that $log_{\mathsf{P}}\mathcal{O} = r \equiv 0 \pmod{r}$

$\because log_{\mathsf{P}}$ is a group homomorphism.

Let P and $\mathsf{Q}\ \epsilon\ \mathsf{E}(\mathbb{F}_p)$. Assume Q is a multiple of P and let $n_0 > 0$ be a solution to $n\mathsf{P} = \mathsf{Q}$. Let $s > 0$ be the smallest solution to $s\mathsf{P} = \mathcal{O}$.

Write $n = ms + r$, $0 \leq r < s$. We have that:

$$\mathsf{Q} = n\mathsf{P} =(ms+r)\mathsf{P} = m(s\mathsf{P}) + r\mathsf{P} = m\mathcal{O} + r\mathsf{P} = \mathcal{O} + r\mathsf{P} = r\mathsf{P}$$

So if n is a solution to our equation then so is r and the smallest such r is n_0. We conclude that every solution is of the type

$$n= ms + n_0 \ .$$

We need an efficient algorithm to compute $n\mathsf{P}$ and, since addition in $\mathsf{E}(\mathbb{F}_p)$ is not entirely trivial, we would not want to compute $2\mathsf{P}$, $3\mathsf{P}$, ... and so on.

Let us now compute $n\mathsf{P}$ mimicking the square-and-multiply algorithm.

The *Double − and − Add Algorithm*:

Step 1. We write n in binary form:

$$n = n_0 + n_1 \cdot 2 + n_2 \cdot 2^2 + \ldots\ldots\ldots\ldots\ldots\ldots + n_r \cdot 2^r,$$

$$n_i \epsilon \{0, 1\},\ 0 \leq i \leq r,\ n_r = 1.$$

Step 2. Set $\mathsf{Q}_0 = \mathsf{P}$. Compute:

$$\mathsf{Q}_1 = 2\mathsf{Q}_0 = 2\mathsf{P}$$

$$\mathsf{Q}_2 = 2\mathsf{Q}_1 = 2^2\mathsf{Q}_0 = 2^2\mathsf{P}$$

$$\mathsf{Q}_3 = 2\mathsf{Q}_2 = 2^3\mathsf{Q}_0 = 2^3\mathsf{P}$$

$$\vdots$$

$$\vdots$$

$$\mathsf{Q}_r = 2\mathsf{Q}_{r-1} = 2^r\mathsf{Q}_0 = 2^r\mathsf{P}$$

The points Q_i are $2-$power multiples of P. We are going to need r doublings to compute them.

Step 3. Compute:

$$n\mathsf{P} = n_0\mathsf{Q}_0 + n_1\mathsf{Q}_1 + \ldots\ldots\ldots + n_r\mathsf{Q}_r = n_0\mathsf{P} + n_1 2\mathsf{P} + \ldots\ldots\ldots + n_r 2^r\mathsf{P}\ .$$

This requires r additions at most. The total number of operations will be, at most, 2^r point operations in $\mathsf{E}(\mathbb{F}_p)$.

As before we get $2r \leq 2log_2 n$, so the order of the computation steps needed is $\mathscr{O}(log_2 n)$.

Example. Let the elliptic curve over the field $\mathbb{F}_{83}$ be given by $y^2 = x^3 + 23x + 13$. $\mathsf{P} = (24, 14)$ is a point on this curve. We shall compute $19\mathsf{P}$.

$$19 = 1 + 2 + 2^4$$

So we shall need 4 doublings and 2 additions. Using Mathematica I get

P	$(24, 14)$
$2\mathsf{P}$	$(30, 8)$
$4\mathsf{P}$	$(24, 69)$
$8\mathsf{P}$	$(30, 75)$
$16\mathsf{P}$	$(24, 14)$

I am not going to determine the whole addition table for $\mathsf{E}(\mathbb{F}_{83})$ but simply give the relevant results:

$$\mathsf{P} + 2\mathsf{P} = 3\mathsf{P} = (30{,}75)$$

$$(\mathsf{P} + 2\mathsf{P}) + 16\mathsf{P} = (24{,}69)$$

Thus, $19\mathsf{P} = (24{,}69)$.

Incidentally, we observe that $3\mathsf{P} = 8\mathsf{P}$ (and $16\mathsf{P} = \mathsf{P}$) so we have that $5\mathsf{P} = \mathcal{O}$. We can conclude that ord $\mathsf{P} = 5$.

Consider the binary expansion

$$947 = 1 + 2 + 2^4 + 2^5 + 2^7 + 2^8 + 2^9$$

We need 9 doublings + 6 additions = 15 point operations if we want to use the algorithm.

But

$$947 = 1 + 2 - 2^4 - 2^6 + 2^{10} \text{ too.}$$

This is called a ternary expansion of n and , considering points on $E(\mathbb{F}_p)$, we might write

$$947P = P + 2P - 2^4P - 2^6P + 2^{10}P \ .$$

Now we need 10 doublings + 4 additions = 14 point operations. Consequently, using sums and differences of 2−powers might reduce the number of operations that the algorithm requires.

Suppose n is a large number. In the "worst" of cases

$$n = 2^k - 11 + 2 + 2^2 + \ldots\ldots\ldots\ldots + 2^{k-1} = \frac{1\cdot(2^k-1)}{2-1}$$

So, in the "worst" of cases, computing nP will require, at most, k doublings + k additions = $2k$ point operations if we use the binary expansion. If we take a random number its binary expansion will have approximately the same number of 1's and 0's so we shall be needing, for most n, approximately k doublings + $\frac{1}{2}k$ additions = $\frac{3}{2}k$ point operations, but we can diminish this if we use the ternary expansion.

Theorem 2. *Let n be a positive integer and let $k = \lfloor log\ n \rfloor + 1$, which means that $2^k > n$. Then we can write*

$$n = u_0 + u_1 \cdot 2 + u_2 \cdot 2^2 + u_3 \cdot 2^3 + u_k \cdot 2^k$$

with $u_i \in \{-1, 0, 1\}$, $0 \leq i \leq k$, and at most $\frac{1}{2}k$ of the u_i nonzero.

Proof. We consider the binary expansion of n. From left to right we spot out the first occurrence of two or more consecutive nonzero coefficients n_i, e.g.

$$n_j = n_{j+1} = \ldots\ldots\ldots\ldots = n_{s+t-1} = 1,\ n_{s+t} = 0,\ t \geq 1.$$

Thus $2^s + 2^{s+1} + \ldots\ldots\ldots\ldots + 2^{s+t-1} + 0 \cdot 2^{s+t}$ will appear in the binary expansion of n.

$$2^s + 2^{s+1} + \ldots\ldots\ldots\ldots + 2^{s+t-1} + 0 \cdot 2^{s+t} =$$

$$2^s(1 + 2 + 2^2 + \ldots. + 2^{t-1}) = 2^s(2^t - 1),$$

so we can replace this sequence with $-2^s + 2^{s+t}$. We repeat the procedure until we end up with a ternary expansion of n in which no consecutive u_i are zero.

□

Whole blocks with 2 or more nonzero coefficients are replaced with only 2 terms. On average $\frac{2}{3}$ of the terms will be zero and, since the ternary expansion might go up to 2^k we shall be needing $k+1$ doublings + $\frac{1}{3}k$ additions = $\frac{4}{3}k + 1$ point operations.

3.2 A collision algorithm

DLP

Say again that we want to solve the discrete logarithm problem in G:

$$g^x = h$$

Theorem 3. *(Trivial Bound for DLP). Let G be a group and let $g \epsilon G$ be an element of order N. Then the discrete logarithm problem can be solved in $\mathcal{O}$(N) steps, where each step consists of multiplication by g.*

Proof. Make a list g, g^2, g^3, ..., g^{N-1}. If a solution exists it will appear in the list.

□

Remark. According to Hoffstein, Pipher & Silverman, if $G = \mathbb{F}_p^\star$, then computing $g^x \pmod p$ requires $\mathcal{O}((\log p)^k)$ computer operations, where k is a constant depending on the computer and the algorithm used for modular multiplication, so the total number of computer steps, or the running time, will be $\mathcal{O}(N(\log p)^k)$ but $\log p$ is negligible, thus we may say that the running time is $\mathcal{O}(N)$.

This was the brute force attack!

Theorem 4. *(Shank's Babystep − Giantstep Algorithm) Let G be a group and let $g \epsilon G$ be an element of order $N \geq 2$. The following algorithm solves the DLP $g^x = h$ in $\mathcal{O}$(logN$\sqrt{N}$) steps:*

(1) Let $n = 1 + \llcorner\sqrt{N}\lrcorner$, so $n > N$.

(2) Create two lists:

$$L1 = \{e,\ g,\ g^2,\ .....,\ g^n\}$$

$$L2 = \{h,\ hg^{-n},\ hg^{-2n},\ .....,\ hg^{-n^2}\}$$

(Multiplication by g = babystep ; multiplication by g^{-n} = giantstep.)

(3) Find a match between the two lists, say $g^i = hg^{-jn}$. Then $x = i + jn$ is a solution for the DLP.

Proof. In L2 we start with $u = g^{-n}$ and then multiply:

$$h, hu, hu^2, ..., hu^n.$$

We need $2n$ multiplications to create the two lists. If a match exists, using standard sorting and searching algorithms, we can find it in a small multiple of $\log n$ steps (all this according to Hoffstein, Pipher & Silverman) thus Step (3) above will require $\mathcal{O}(\log n)$ steps. Consequently the total running time for the algorithm is $\mathcal{O}(n\log n)$ because, for each element in L1, you have to search L2. Since $n \approx\sqrt{\mathsf{N}}$, $n\log n\approx\sqrt{\mathsf{N}}\log\sqrt{\mathsf{N}} = \frac{1}{2}\sqrt{\mathsf{N}}\log\mathsf{N}$, we have that $\mathcal{O}(n\log n) = \mathcal{O}(\sqrt{\mathsf{N}}\log\mathsf{N})$, as desired.

We still have to show, though, that L1 and L2 have a match.

Let x be the unknown solution to the DLP and write $x = nq + r$, $0 \leq r < n$. We know furthermore that $1 \leq x < \mathsf{N}$, thus $q = \frac{x-r}{n} < \frac{\mathsf{N}}{n} < n$, since $n > \sqrt{\mathsf{N}}$.

$g^x = h$ becomes $g^r = hg^{-qn}$ with $0 \leq r < n$ and $0 \leq q < n$. Thus $g^r \epsilon$ L1 and $hg^{-qn} \epsilon$ L2. Therefore, the lists always have a match.

□

The algorithm above relies on a probabilistic collision theorem that we formulate without proof.

Theorem 5. *An urn contains* N *balls, of which* n *are red and the rest are blue. You select randomly a ball from the urn, replace it in the urn, select randomly a second ball, replace it in the urn, and so on. You do this until you have looked at a total of* m *balls. The probability that you select at least one red ball is*

$$\mathcal{P}\,(\textit{at least one red}) = 1-(1 - \tfrac{\mathrm{n}}{\mathrm{N}})^m.$$

Proof. (See Theorem 4.38 in Hoffstein, Pipher & Siverman). □

To connect this theorem to our algorithm we consider the union of the two lists to be an urn containg N numbered blue balls. In the process of constructing L1 we repaint n of the balls red and return them to the urn. L2 is constructed by drawing m balls from the urn, one at a time, noting their number and colour, and then replacing them in the urn. The probability of selecting at least one red ball is equal to the probability of a match between the two lists.

There is a more general (and, possibly, more efficient) algorithm. (See Proposition 4.44 in Hoffstein, Pipher & Siverman).

Assume that the DLP $g^x = b$ has a solution

(1) Choose random exponents $y_1, y_2, ..., y_n$ between 1 and N.

(2) Create the list

$$\mathsf{L1} = \{g^{y_1}, g^{y_2}, ..., g^{y_n}\} \subseteq \{1, g, g^2, ..., g^{\mathsf{N}-1}\},\ (\ g^{\mathsf{N}} = 1).$$

(3) Choose new random exponents $z_1, z_2, ..., z_n$ between 1 and some "large enough" k .

(4) Create the list

$$\mathsf{L2} = \{bg^{z_1}, bg^{z_2}, ..., bg^{z_n}\} \subseteq \{1, g, g^2, ..., g^{\mathsf{N}-1}\}.$$

The inclusion is justified since we have assumed that $g^x = b$ has a solution. L2 is created by selecting n elements from the urn. It takes about $2n$ steps to construct the two lists. Each element in each list requires a computation of some g^i, $1 \leq i < \mathsf{N}$.

This takes approximately $2\log_2 i$ group multiplications using the square–and–multiply algorithm. Thus far we need approximately $4n\log_2 \mathsf{N}$ multiplications for the two lists. We need other $\log_2 n$ steps to check whether an element in L2 has a match in L1, thus $n\log_2 n$ comparisons altogether. The grand total becomes $4n\log_2 \mathsf{N} + n\log_2 n = n\log_2(n\mathsf{N}^4)$ steps.

According to Hoffstein, Pipher & Silverman (Proposition 4.44) $n \approx 3\sqrt{\mathsf{N}}$ gives a 99.98% chance of a match, in which case the running time will be approximately $13.5\sqrt{\mathsf{N}}\log_2(1.3\mathsf{N})$.

ECDLP

Let us look at the case $\mathsf{G} = \mathsf{E}(\mathbb{F}_p)$.

We want to solve the equation

$$\mathsf{Q} = n\mathsf{P}.$$

Choose random integers j_1, j_2, ..., j_r and k_1, k_2, ..., k_r between 1 and p and make the lists

$$\mathsf{L1} = \{j_1\mathsf{P}, j_2\mathsf{P}, ..., j_r\mathsf{P}\}$$

$$\mathsf{L2} = \{k_1\mathsf{P} + \mathsf{Q}, k_2\mathsf{P} + \mathsf{Q}, ..., k_r\mathsf{P} + \mathsf{Q}\}.$$

If you find a match (collision) you are done since

$$j_u\mathsf{P} = k_v\mathsf{P} + \mathsf{Q}$$

gives

$$\mathsf{Q} = (j_u - k_v)\mathsf{P}$$

and we have a solution $n = j_u - k_v$.

Mimicking the general case if r is somewhat larger than $\sqrt{p}$, say $r \approx 3\sqrt{p}$, then there is a very good chance of a collision. There are no general algorithms known to solve the ECDLP in fewer than $\mathcal{O}(\sqrt{p})$ steps according to our usual source.

3.3 Pollard's ρ algorithm

All these algorithms seem to require a lot of storage room for the two lists. Pollard has constructed an algorithm where practically no storage is needed!

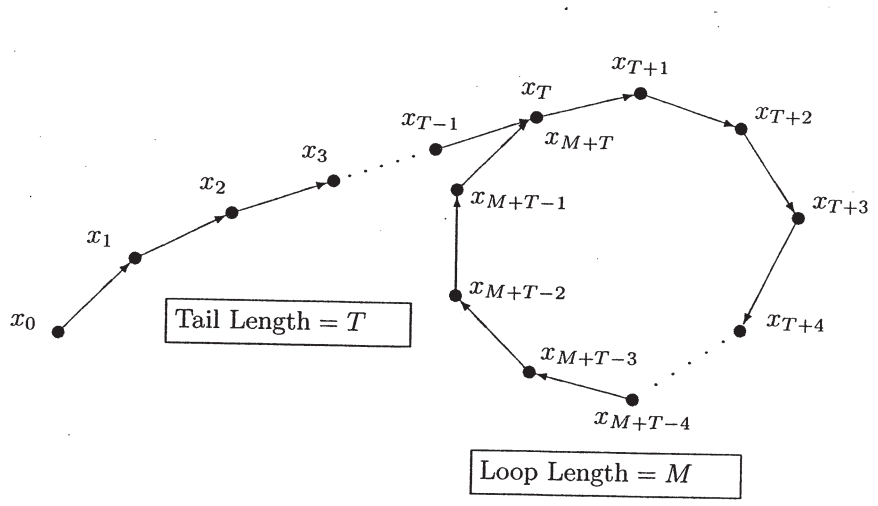
DLP

Suppose S is a finite set and $f : \mathsf{S} \longrightarrow \mathsf{S}$ an efficient mixing map. We iterate f and create a sequence of elements

$$x_0 = x,\ x_1 = f(x_0) = f(x),\ x_2 = f\ (x_1) = f^2(x_0) = f^2(x),\ \ldots\ldots\ ,$$
$$x_i = f^i(x).$$

f creates a discrete dynamical system and the set
$\mathcal{O}_f^+(x) = \{x_0, x_1, \ldots\ldots\ldots\ldots\}$ is the forward orbit of x under f.

Since S is a finite set some element of S will appear a second time in $\mathcal{O}_f^+(x)$ and from that moment onward the system will enter a loop (of length M).



Set T = the largest integer such that $x_{\mathsf{T}-1}$ appears only once in $\mathcal{O}_f^+(x)$ and M = the smallest integer such that $x_{\mathsf{T}+\mathsf{M}} = x_{\mathsf{T}}$.

Suppose $\mid \mathsf{S} \mid = \mathsf{N}$. Since $x_{\mathsf{T}+\mathsf{M}} = x_{\mathsf{T}}$ we shall obtain a collision in $\mathcal{O}(\sqrt{\mathsf{N}})$ steps (see below). But it seems as though we must store the list $x_0, x_1, ..., x_{\mathsf{T}}, x_{\mathsf{T}+1}, ..., x_{\mathsf{T}+\mathsf{M}}$.

Create another sequence $\{y_i\}_i$ such that $y_0 = x_0$ and $y_{i+1} = f^2(y_i)$, $i = 0, 1, 2, ...$, that is, at each step we apply f once to generate x_i and once again to generate y_i, hence $y_i = x_{2i}$.

How long do we have to go on before we find some i such that $x_i = x_{2i}$? In general, of course, for $j > i$, $x_j = x_i$ if and only if $i \geq \mathsf{T}$ and $j \equiv i \pmod{\mathsf{M}}$. This should be clear from the picture above: we must have passed the point x_{T} , so $i \geq \mathsf{T}$, and x_j must have passed x_i an integral number of times around the loop, so $j - i = k\mathsf{M}$, i.e. $j \equiv i \pmod{\mathsf{M}}$.

In consequence, since we want $x_{2i} = x_i$, this is possible if and only if the index $i \geq \mathsf{T}$ and $2i \equiv i \pmod{\mathsf{M}}$ meaning that $\mathsf{M} \mid i$, so we get the first $x_{2i} = x_i$ exactly when i is equal to the first multiple of M larger than T. But one of the numbers $\mathsf{T}, \mathsf{T}+1, ..., \mathsf{T}+\mathsf{M}-1$ must be divisible by M, so $x_{2i} = x_i$ for some $1 \leq i < \mathsf{T}+\mathsf{M}$.

We state, without proof, the following:

Theorem 6. *If the map f is sufficiently (!) random, then the expected value of $\mathsf{T}+\mathsf{M}$ is $\mathcal{E}(\mathsf{T}+\mathsf{M}) \approx 1.2533\sqrt{\mathsf{N}}$, a small multiple of $\sqrt{\mathsf{N}}$. Hence if N is large we are likely to find a match in $\mathcal{O}(\sqrt{\mathsf{N}})$ steps, where "step" means one evaluation of f.*

Proof. For a sketch of the proof see Theorem 4.47 (b) in Hoffstein, Pipher & Siverman. □

Let us try now to use Pollard's ρ method to solve the DLP $g^x = b$ in $\mathbb{F}_p^\star$, when g is a primitive root mod p. We want basically to find a collision between $g^i b^j$ and $g^k b^l$ for some known exponents i, j, k, l.

$g^i b^j \equiv g^k b^l \Leftrightarrow g^{i-k} \equiv b^{l-j} \pmod{p}$ and taking roots in $\mathbb{F}_p$ will do the trick. The problem is finding a function

$$f\colon \mathbb{F}_p \longrightarrow \mathbb{F}_p$$

complicated enough to give a good mixing of the elements of $\mathbb{F}_p$, yet simple enough to handle.

Pollard suggests the function

$$f(x) \begin{cases} gx & , 0 \leq x < \frac{p}{3} \\ x^2 & , \frac{p}{3} \leq x < \frac{2p}{3} \\ bx & , \frac{2p}{3} \leq x < p \end{cases}$$

x will have to be reduced mod p before evaluating $f(x)$.

We must mention that no one has proved that f above is sufficiently random but experimentally it works fairly well according to Hoffstein, Pipher & Silverman.

Suppose we start with $x_0 = 1$. At each step we either

- multiply by g , or
- multiply by b , or
- square the previous value.

What we get at each step is $x_i = g^{\alpha_i} b^{\beta_i}$. Starting, clearly with $\alpha_0 = \beta_0 = 0$, we can compute the subsequent values at each step by

$$\alpha_{i+1} = \begin{cases} \alpha_i + 1 & , 0 \leq x < \frac{p}{3} \\ 2\alpha_i & , \frac{p}{3} \leq x < \frac{2p}{3} \\ \alpha_i & , \frac{2p}{3} \leq x < p \end{cases}$$

and

$$\beta_{i+1} = \begin{cases} \beta_i & , 0 \leq x < \frac{p}{3} \\ 2\beta_i & , \frac{p}{3} \leq x < \frac{2p}{3} \\ \beta_i + 1 & , \frac{2p}{3} \leq x < p \end{cases}$$

and we reduce their values mod p since $g^{p-1} \equiv 1 \equiv b^{p-1}$ (Fermat's Little Theorem). Otherwise the values of α_i and β_i would become astronomical!

We compute in a similar fasion the other sequence given by $y_0 = 1$ and $y_{i+1} = f^2(y_i)$, thus $y_i = x_{2i} = g^{\gamma_i} b^{\delta_i}$, where the exponents γ_i and δ_i are computed by two iterations of the same recursion as for α_i and β_i.

Nota bene: the first time we use y_i and the second time we use $f(y_i)$ in order to decide which case to apply.

This way we shall eventually find a collision between the two sequences, say $y_i = x_{2i}$, meaning that we shall have found $g^{\alpha_i} b^{\beta_i} = g^{\gamma_i} b^{\delta_i}$, so letting $u \equiv \alpha_i - \gamma_i$ and $v \equiv \delta_i - \beta_i \pmod{(p-1)}$ we have $g^u \equiv b^v$ in $\mathbb{F}_p$ which is equivalent to $u \equiv v \log_g b \pmod{(p-1)}$.

If $\gcd(v,\ p-1) = 1$, then $\log_g b \equiv v^{-1} u \pmod{(p-1)}$ and we have solved the DLP. If $\gcd(v,\ p-1) = d \geq 2$, then we use the Euclidean algorithm to find an integer s such that $sv \equiv d \pmod{(p-1)}$ which then leads to

$$sv \log_g b \equiv su \pmod{(p-1)}$$

$$d\mathsf{log_g}\, b \equiv w \pmod{(p-1)}$$

We know that $d \mid p-1$, thus $d \mid w$ and so, finally, $\mathsf{log_g}\, b \equiv wd^{-1}$ is one solution. We get several, in fact

$$\mathsf{log_g}\, b \;\epsilon\; \{wd^{-1} + k(p-1)d^{-1} : k = 0, 1, 2, ..., d-1\}$$

ECDLP

The algorithms presented above can all be adapted to solve the ECDLP $n\mathsf{P} = \mathsf{Q}$.

We shall try to find a collision of points. Mimicking the standard version we want to find a collision between $i\mathsf{P} + j\mathsf{Q}$ and $k\mathsf{P} + l\mathsf{Q}$ for some known integers i, j, k, l.

$$i\mathsf{P} + j\mathsf{Q} = k\mathsf{P} + l\mathsf{Q}$$

$$(i-k)\mathsf{P} = (l-j)\mathsf{Q}$$

We need a mixing function F: $\mathsf{E}(\mathbb{F}_p) \longrightarrow \mathsf{E}(\mathbb{F}_p)$. Comparing with the standard case we realize that the mixing occurs between the three classes of a partition. Partition $\mathsf{E}(\mathbb{F}_p)$ into the sets $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$ of approximately the same size and with the *proviso* $\mathcal{O} \notin \mathcal{S}_2$. Define F by

$$F(\mathsf{R}) = \begin{cases} \mathsf{R} + \mathsf{Q} & , \mathsf{R}\epsilon\mathcal{S}_1 \\ 2\mathsf{R} & , \mathsf{R}\epsilon\mathcal{S}_2 \\ \mathsf{R} + \mathsf{Q} & , \mathsf{R}\epsilon\mathcal{S}_3 \end{cases}$$

Suppose we start with $\mathsf{R}_0 = \mathcal{O}$. At each step we either

- add Q , or
- add P , or
- double the previous value.

What we get at each step is $\mathsf{R}_i = \alpha_i\mathsf{P} + \beta_i\mathsf{Q}$. Starting, clearly with $\alpha_0 = \beta_0 = 0$, we can compute the subsequent values at each step by

$$\alpha_{i+1} = \begin{cases} \alpha_i + 1 & , \mathsf{R}\epsilon\mathcal{S}_1 \\ 2\alpha_i & , \mathsf{R}\epsilon\mathcal{S}_2 \\ \alpha_i & , \mathsf{R}\epsilon\mathcal{S}_3 \end{cases}$$

and

$$\beta_{i+1}= \begin{cases} \beta_i & ,\mathsf{R}\epsilon\mathcal{S}_1 \\ 2\beta_i & ,\mathsf{R}\epsilon\mathcal{S}_2 \\ \beta_i+1 & ,\mathsf{R}\epsilon\mathcal{S}_3 \end{cases}$$

We shall create, of course even here, two lists: $\{\mathsf{R}_i\}$ and $\{\mathsf{R}_{2i}\}$. For all i we shall have $log_{\mathsf{P}}\mathsf{R} = log_{\mathsf{P}}(\alpha_i\mathsf{P}) + log_{\mathsf{P}}(\beta_i\mathsf{Q}) = \alpha_i + \beta_i log_{\mathsf{P}}\mathsf{Q} = \alpha_i + n\beta_i$ for the sequence $\{\mathsf{R}_i\}$ whereas for the sequence $\{\mathsf{R}_{2i}\}$ we get $log_{\mathsf{P}}\mathsf{R} = \alpha_{2i} + n\beta_{2i}$ which entails that

$$\alpha_{2i} + n\beta_{2i} = \alpha_i + n\beta_i$$

and the result is:

$$n = \frac{\alpha_i-\alpha_{2i}}{\beta_{2i}-\beta_i}.$$

We have solved the ECDLP.

3.4 The Pohlig – Silver – Hellman algorithm

DLP

We shall present another algorithm for the DLP $g^x \equiv h \pmod p$.

$x = \mathsf{log}_{\mathsf{g}}h$ and as such it is determined modulo $p-1$. If the order of g is N then the solutions to $g^x = h$ in G are determined modulo N so the prime factorization of N must be relevant.

Theorem 7. *(Pohlig −Silver − Hellman) Let $g\epsilon$ G be an element of order N and suppose N factors into*

$$\mathsf{N} = q_1^{e_1}\ q_2^{e_2}.....q_t^{e_t},$$

q_i being distinct primes.

Then the DLP $g^x \equiv h$ (mod p) can be solved in the following way:

Step 1. For each $1 \leq i \leq t$ let $g_i = g^{\mathsf{N}/q_i^{e_i}}$ and $h_i = h^{\mathsf{N}/q_i^{e_i}}$. (Notice that g_i has order $= q_i^{e_i}$, prime power.)

Solve the DLP $g_i^y \equiv h_i$ and let $y = y_i$ be a solution.

Step2. Use the Chinese Remainder Theorem to solve

$$\begin{cases} x \equiv y_1 & (mod q_1^{e_1}) \\ x \equiv y_2 & (mod q_2^{e_2}) \\ \qquad\vdots & \\ x \equiv y_t & (mod q_t^{e_t}) \end{cases}$$

The solution to this system of congruences is the solution to our DLP $g^x \equiv h \ (mod\ p)$.

Proof. Let x be a solution to the system of congruences above. Then for each i we have

$$x = y_i + q_i^{e_i} \cdot z_i \text{ , for some } z_i, \text{ so}$$

$$(g^x)^{\mathsf{N}/q_i^{e_i}} = g^{(y_i + q_i^{e_i} \cdot z_i)^{\mathsf{N}/q_i^{e_i}}} = g^{(\mathsf{N}/q_i^{e_i})y_i} \cdot g^{\mathsf{N}z_i} = g_i^{y_i} = h_i = h^{\mathsf{N}/q_i^{e_i}} \text{ , } (g^{\mathsf{N}} = \mathtt{1}).$$

This means, in terms of the discrete logarithm to the base g, that

$$(\star)\ \tfrac{\mathsf{N}}{q_i^{e_i}} x \equiv \tfrac{\mathsf{N}}{q_i^{e_i}} \mathsf{log}_g h \ (\text{mod } \mathsf{N}).$$

Observe now that $\frac{\mathsf{N}}{q_1^{e_1}}$, $\frac{\mathsf{N}}{q_{2i}^{e_2}}$, ..., $\frac{\mathsf{N}}{q_t^{e_t}}$ have no nontrivial common factors so their greatest common divisor is $\mathtt{1}$. By repeated application of the Euclidean algorithm we get

$$(\dagger)\ c_1 \tfrac{\mathsf{N}}{q_1^{e_1}} + c_2 \tfrac{\mathsf{N}}{q_{2i}^{e_2}} + \ldots\ldots\ldots\ldots + c_t \tfrac{\mathsf{N}}{q_t^{e_t}} = \mathtt{1}.$$

Multiply $(\star)$ by c_i and sum over $i = \mathtt{1}, \mathtt{2}, \ldots\ldots, \mathtt{t}$, in order to get

$$\Sigma_{i=1}^t c_i \tfrac{\mathsf{N}}{q_i^{e_i}} x = \Sigma_{i=1}^t c_i \tfrac{\mathsf{N}}{q_i^{e_i}} \mathsf{log}_g h \ (\text{mod } \mathsf{N}).$$

By $(\dagger)$ we have $x = \mathsf{log}_g h$, thus x is a solution to $g^x \equiv h \ (\text{mod } p)$.

□

The lesson to draw is that the Pohlig − Silver − Hellman Algorithm reduces the DLP for elements of arbitrary order to the DLP for elements of prime power order and this means that the DLP in the group G is as hard as the DLP in the subgroup of G with highest prime power order ! Furthermore, this can be refined to reduce the whole problem to the DLP for elements of prime order.

Theorem 8. *Let* G *be a group. Suppose* q *is a prime and let* $g \epsilon$ G *be an element of order* q^e, $e \geq \mathtt{1}$. *Then we can reduce the* DLP $g^x \equiv h$ *in* G *for an element of prime power order to a* DLP *for an element of prime order.*

Proof. Write the unknown exponent x as

$$x = x_0 + x_1 q + x_2 q^2 + \ldots\ldots\ldots\ldots\ldots + x_{e-1} q^{e-1} \ , \ 0 \leq x_i < q$$

and, then, determine successively x_0, x_1, x_2, ... like this:

$$h^{q^{e-1}} \equiv (g^x)^{q^{e-1}} = g^{\left(x_0 + x_1 q + x_2^2 + \ldots\ldots\ldots\ldots\ldots + x_{e-1} q^{e-1}\right)^{q^{e-1}}} =$$

$$g^{x_0 q^{e-1}} \cdot \left(g^{q^e}\right)^{(x_1 + x_2 q + \ldots\ldots\ldots\ldots\ldots + x_{e-1} q^{e-2})} \equiv \left(g^{q^{e-1}}\right)^{x_0}$$

Since $g^{q^e} = 1$, the equation $\left(g^{q^{e-1}}\right)^{x_0} \equiv h^{q^{e-1}}$ which we must solve is a DLP whose base $g^{q^{e-1}}$ is an element of prime order q. After determining x_0 we start afresh:

$$h^{q^{e-2}} \equiv (g^x)^{q^{e-2}} = g^{\left(x_0 + x_1 q + x_2^2 + \ldots\ldots\ldots\ldots\ldots + x_{e-1} q^{e-1}\right)^{q^{e-2}}} =$$

$$g^{x_0 q^{e-2}} \cdot g^{x_1 q^{e-1}} \cdot \left(g^{q^e}\right)^{(x_2 + x_3 q + \ldots\ldots\ldots\ldots\ldots + x_{e-1} q^{e-3})} \equiv \left(g^{q^{e-2}}\right)^{x_0} \cdot \left(g^{q^{e-1}}\right)^{x_1}.$$

At this level we must solve the DLP

$$\left(g^{q^{e-1}}\right)^{x_1} \equiv (h g^{-x_0})^{q^{e-2}}$$

for x_1 , with base $g^{q^{e-1}}$, an element of prime order q.
At the following level we have the DLP

$$\left(g^{q^{e-1}}\right)^{x_2} \equiv (h g^{-x_0 - x_1 q})^{q^{e-3}}$$

for x_2 , with base $g^{q^{e-1}}$, an element of prime order q.
In general form we must solve the DLP

$$\left(g^{q^{e-1}}\right)^{x_i} \left(g^{q^{e-1}}\right)^{x_2} \equiv \left(h g^{-x_0 - x_1 q - \ldots\ldots\ldots - x_{i-1} q^{i-1}}\right)^{q^{e-i+1}}$$

for x_i , with base $g^{q^{e-1}}$, an element of prime order q.
Finally, after solving a number of DLP′s whose bases are elements of prime order, we get the exponent

$$x = x_0 + x_1 q + x_2 q^2 + \ldots\ldots\ldots\ldots\ldots + x_{e-1} q^{e-1} \ , \ 0 \leq x_i < q$$

which solves the original DLP.

□

ECDLP

The Pohlig − Silver − Hellman Algorithm can be adapted for solving ECDLP, i. e. $n\mathsf{P} = \mathsf{Q}$. Assume that ord $\mathsf{P} = m = \Pi_{i=1}^{t} q_i^{e_i}$, q_i distinct primes.

Step 1. Let $t_i = m/q_i^{e_i}$. Set $\mathsf{Q}' = t_i\mathsf{Q}$. We have

$$\mathsf{Q}' = t_i\mathsf{Q} = t_i(n\mathsf{P}) = n\ (t_i\mathsf{P}) = n\mathsf{P}'.$$

(These equalities make it clear that both Q' and P' belong to a subgroup of the group generated by P of order $q_i^{e_i}$, a prime order.) Now all we have to do is to solve, for all i, the equations

$$\mathsf{Q}' = n\ \mathsf{P}' \ (\text{mod } q_i^{e_i})$$

Step 2. Use the Chinese Remainder Theorem to recover n (mod m).

A further refinement now would be to reduce the problem to solving equations modulo a prime.

Assume that ord $\mathsf{P} = q^e$, q a prime, i.e. $q^e\mathsf{P} = \mathcal{O}$. Expand n as

$$n = n_0 + n_1 q + n_2 q^2 + \ldots\ldots\ldots\ldots + n_{e-1} q^{e-1}$$

$$q^{e-1}\mathsf{Q} = q^{e-1}(n\mathsf{P}) =$$

$$q^{e-1}(n_0 + n_1 q + n_2 q^2 + \ldots\ldots\ldots\ldots + n_{e-1} q^{e-1})\mathsf{P} =$$

$$q_0^{e-1} n_0 \mathsf{P} + (\ n_1 + n_2 q + \ldots\ldots\ldots\ldots + n_{e-1} q^{e-2}) q^e \mathsf{P} =$$

$$q_0^{e-1} n_0 \mathsf{P} + \mathcal{O} =$$

$$n_0\ (q_0^{e-1}\mathsf{P})$$

We must thus solve the equation

$$n_0\ (q_0^{e-1}\mathsf{P}) = q^{e-1}\mathsf{Q}$$

whose base, $q_0^{e-1}\mathsf{P}$, is an element of order q, a prime, as desired.

The rest follows as before and in the end we reassemble

$$n = n_0 + n_1 q + n_2 q^2 + \ldots\ldots\ldots\ldots + n_{e-1} q^{e-1},$$

the solution to the initial problem. *Divide et impera* !

3.5 The MOV algorithm

(In what follows consult the Appendix for the definition of the relevant concepts.)

Definition.

Let E be an elliptic curve over $\mathbb{F}_p$ and let $m \geq 1$ with $p \nmid m$. (In cryptography one usually chooses m to be a large prime.) The *embedding degree* of E with respect to m is the smallest value of k such that

$$\mathsf{E}(\mathbb{F}_{p^k})[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}.$$

We state without proof:

Theorem 9. *Let E be an elliptic curve over $\mathbb{F}_p$ and let $l \neq p$ be a prime. Assume that $\mathsf{E}(\mathbb{F}_p)$ contains a point of order l. Then the embedding degree of E with respect to l is*

- 1*, not possible if* $l > \sqrt{p} + 1$
- l*, if* $p \equiv 1 \pmod{l}$
- *the smallest* $k \geq 2$ *such that* $p^k \equiv 1 \pmod{l}$*, if* $\neg(p \equiv 1 \pmod{l})$*. (This is the case that most often happens in practice!)*

Proof. For the proof of (iii) Hoffstein, Pipher & Silverman refer to L.C. Washington, Elliptic Curves: Number Theory and Cryptography, 2003. □

MOV

The MOV Algorithm (**M**enezes, ***O***kamoto, ***V***anstone) goes like this:

1. Compute $\mathsf{N} = \#\mathsf{E}(\mathbb{F}_{p^k})$ which is feasible if k is not too large. *Nota bene*: $l \mid \mathsf{N}$ because $\mathsf{E}(\mathbb{F}_p)$ has a point of order l by assumption.
2. Choose a random point $\mathsf{T} \in \mathsf{E}(\mathbb{F}_{p^k})$ such that $\mathsf{T} \notin \mathsf{E}(\mathbb{F}_p)$.
3. Compute $\mathsf{T}' = (\mathsf{N}/l)\mathsf{T}$. If $\mathsf{T}' = \mathcal{O}$ discard it and go back to Step 2. If not, ord $\mathsf{T}' = l$, so go to Step 4.
4. Compute

$$\alpha = e_l(\mathsf{P}, \mathsf{T}')\epsilon\mathbb{F}^{\star}_{p^k} \text{ and } \beta = e_l(\mathsf{Q}, \mathsf{T}')\epsilon\mathbb{F}^{\star}_{p^k}$$

5. Solve the DLP $\alpha^n = \beta$ for n with α and β in $\mathbb{F}^{\star}_{p^k}$.
6. Then it holds too that $n\mathsf{P} = \mathsf{Q}$ which is the ECDLP in $\mathsf{E}(\mathbb{F}_p)$.

Remark 1. The point T' is generally independent of P so $\{\mathsf{P}, \mathsf{T}'\}$ forms a basis for the $2-$dimensional vector space $\mathsf{E}[l] \cong \mathbb{Z}/l\mathbb{Z} \times \mathbb{Z}/l\mathbb{Z}$. Therefore $e_l(\mathsf{P}, \mathsf{T}')$ is a nontrivial lth root of unity in $\mathbb{F}^{\star}_{p^k}$ by the nondegeneracy of the Weil pairing, i.e. $e_l(\mathsf{P}, \mathsf{T}')^r \equiv 1$ if and only if $l \mid r$. Suppose $\mathsf{Q} = j\mathsf{P}$. We want to determine j modulo l. The MOV algorithm finds an integer n such that $e_l(\mathsf{Q}, \mathsf{T}') = e_l(\mathsf{P}, \mathsf{T}')^n$. By linearity we have

$$e_l(\mathsf{P}, \mathsf{T}')^n = e_l(\mathsf{Q}, \mathsf{T}') = e_l(j\mathsf{P}, \mathsf{T}') = e_l(\mathsf{P}, \mathsf{T}')^j$$

so $n \equiv j \pmod{l}$, which means that n solves the ECDLP for P and Q.

Remark 2. What if k is large, say $k > (\ln p)^2$. Then the algorithm should solve a DLP in $\mathbb{F}_{p^k}$ with $k > 4000$ if we take $p \approx 2^{160}$. A random elliptic curve over $\mathbb{F}_p$ almost always has embedding degree much larger than $(\ln p)^2$and so the MOV algorithm would not do. Yet it shows that it can break elliptic curve cryptography for a certain class of curves whose embedding degree is small. These are the curves satisfying $\#\mathsf{E}(\mathbb{F}_p) = p + 1$, the *supersingular* curves, usually curves of the form $y^2 = x^3 + ax$ in char $(\mathbb{F}_p) = p \equiv -1 \pmod 4$ or $y^2 = x^3 + b$ in char $(\mathbb{F}_p) = p \equiv -1 \pmod 3$. They have quite often embedding degree $k = 2$ or, in any case, $k \leq 6$. (See Koblitz, A Course in Number Theory and Cryptography, VI.2)

Then there are even the *anomalous* curves for which $\#\mathsf{E}(\mathbb{F}_p) = p$. What can we do about these curves? Well, we can avoid them!

3.6 Lenstra's algorithm

Shortly after the publication of Diffie and Hellman's seminal paper in 1976 Rivest, Shamir and Adleman published their paper " A method for obtaining digital signatures and public-key cryptosystems" in *Comm. ACM*, 21(2). This resulted in the first public key cryptosystem built upon the group $\mathbb{Z}_{\mathsf{N}}$, $\mathsf{N} = pq$, p and q distinct primes. It relies on the notorious difficulty of prime factorization and it is called RSA. The system is presented e.g. in Hoffman, Pipher & Silverman and I am not going to discuss it since it is not related or relatable in any way to elliptic curve cryptography. Yet one can use elliptic curve cryptanalysis against it.

Pollard divised his $p-1$ method for prime factorization and Lenstra mimicked that method to construct a prime factorization algorithm using the addition law for $\mathsf{E}(\mathbb{F}_p)$ instead of the multiplication law modulo N.

(See Lenstra, Factoring integers with elliptic curves,
in *Annals of Mathematics* (2), 126 (3), pp 649-673, 1987).

We start with an equation

$$\mathsf{E} : y^2 = x^3 + ax + b$$

Suppose $\mathsf{P}= (u,\ v)$ is a point on E (mod N) . This entails that $v^2 \equiv u^3 + au + b$ (mod N).

Then we compute $2\mathsf{P}$, $3\mathsf{P}$, ...

Nota bene. During our computations we might need the reciprocal of non-units. $\mathbb{Z}_{\mathsf{N}}$ is not a field !

Mimicking Pollard's $p-1$ method Lenstra replaces multiplication modulo N with addition modulo N and computes

$$2!\mathsf{P},\ 3!\mathsf{P},\ \ldots\ldots\ldots\ (\text{mod } \mathsf{N})$$

If we have already computed $\mathsf{Q} = (n-1)!\mathsf{P}$ then it is easy to compute $n!\mathsf{P} = n\mathsf{Q}$. (The algorithm uses factorials since N is supposedly large so this can speed up things a bit.)

What can happen?

- we are able to compute $n!\mathsf{P}$ and after a preset number of iterations we get nothing and start again.
- during the computation we may need the reciprocal of a number $d = k\mathsf{N}$ which is not good at all but not very likely to happen since we would be working modulo N all along. In any case we start again.
- we may need the reciprocal of a number d such that $1 < \gcd(d,\ \mathsf{N}) < \mathsf{N}$. We fail again to compute $n!\mathsf{P}$ but $\gcd(d,\ \mathsf{N}) = d > 1$ would then be a non-trivial factor so we are done ! *Lesson*: failure may be benefficient.

Algorithm

Input. Integer N to be factored.

1. Choose random values a, u and v modulo N.

2. Set $\mathsf{P} = (u, v)$ and $b \equiv v^2 - u^3 - a \cdot u$ modulo N. (This solves the problem of finding a point on the elliptic curve; we start with a point and find a curve that will do instead !) Let E be the elliptic curve $y^2 = x^3 + ax + b$.

3. Loop $j = 2, 3, 4, ...$ up to a specified bound.

4. Compute $\mathsf{Q} \equiv j\mathsf{P} \pmod{\mathsf{N}}$ and set $\mathsf{P} = \mathsf{Q}$.

5. If the computation in Step 4 fails, then we have found a $d > 1$ such that $d|\mathsf{N}$.

6. If $d < \mathsf{N}$ then **success**, return d.

7. If $d = \mathsf{N}$, go to Step 1 and choose a new point and a new curve.

8. Increment j and loop again at Step 2.

Remark. There exist very powerful sieve factorization methods: the average running time of the quadratic sieve to factor a composite number N is approximately $\mathscr{O}(e^{\sqrt{log\mathsf{N} \cdot \log(\log\mathsf{N})}})$ steps thus depending on the integer N.

Using the elliptic curve factorization algorithm the running time will depend on the smallest factor of N, say p, viz. $\mathscr{O}(e^{\sqrt{2log\mathsf{p} \cdot \log(\log\mathsf{p})}})$ steps. But a sieve method step is much faster than an elliptic curve step!

If $\mathsf{N} = pq$ and $p \approx q$ the running times of these two methods are approximately the same. Nonetheless, the elliptic curve method should not be dismissed since it can find moderately large factors of extremely large numbers in possibly shorter time since its running time depends on $p < \mathsf{N}$ and not on N. (For all this see Hoffstein, Pipher & Silverman)

4 *Appendix: Elliptic Curves*

“ O mathématiques sévères, je ne vous ai pas oubliées, depuis que vos savantes leçons, plus douces que le miel, filtrèrent dans mon coeur, comme une onde rafraîchissante.

[.....]

Arithmétique! algèbre! géométri! trinité grandiose! triangle lumineux! Celui qui ne vous a pas connues est un insensé! “

(Le Comte de Lautréamont, *Les Chants de Maldoror* , *Chant II*)

4.1 Elliptic curves over $\mathbb{R}$

Consider the equation :

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

This is called a Weierstraβ equation. For $(x, y)\ \epsilon\ \mathbb{R}^2$ the set of solutions to this equation will determine a geometric locus which we shall call a *cubic* curve. The presence of the mixed term a_1xy suggests that some substitution might rotate the axes and thus eliminate it. In fact, over any field k such that $char(\mathsf{k}) \neq 2$ the substitution

$$y = \tfrac{1}{2}(y' - a_1x' - a_3)$$

will do the trick.

$$\tfrac{1}{4}(y - a_1x - a_3)^2+\tfrac{1}{2}\ (a_1x + a_3)(y - a_1x - a_3) =$$

$$(\tfrac{1}{4}y-\tfrac{1}{4}a_1x-\tfrac{1}{4}a_3+\tfrac{1}{2}a_1x+\tfrac{1}{2}a_3)(y - a_1x - a_3) =$$

$$(\tfrac{1}{4}y+\tfrac{1}{4}a_1x+\tfrac{1}{4}a_3)(y - a_1x - a_3) =$$

$$\tfrac{1}{4}(y+(a_1x + a_3))(y-(a_1x + a_3)) =$$

$$\tfrac{1}{4}(y^2-(a_1x + a_3)^2) =$$

$$\tfrac{1}{4}(y^2\ -a_1^2x^2-2a_1a_2x-a_3^2) =$$

$$\tfrac{1}{4}y^2-\tfrac{1}{4}a_1^2x^2-\tfrac{1}{2}a_1a_3x-\tfrac{1}{4}a_3^2.$$

So we have :

$$\tfrac{1}{4}y^2-\tfrac{1}{4}a_1^2x^2-\tfrac{1}{2}a_1a_3x-\tfrac{1}{4}a_3^2 = x^3 + a_2x^2 + a_4x + a_6$$

which is equivalent to

$$y^2 = 4x^3+\ 4a_2x^2+a_1^2x^2+4a_4x+2a_1a_3x+a_3^2+4a_6$$

thus obtaining

$$y^2= 4x^3+(a_1^2+4a_2)x^2+2(a_1a_3+2a_4)x+(a_3^2+4a_6).$$

Setting

$$\begin{cases} b_2 = a_1^2 + 4a_2 \\ b_4 = a_1 a_3 + 2a_4 \\ b_6 = a_3^2 + 4a_6 \end{cases}$$

we can write

$$y^2 = 4x^3 + b_2x^2 + 2b_4x + b_6.$$

Scaling y by means of $y = 2y'$ and renaming coefficients we get in the end

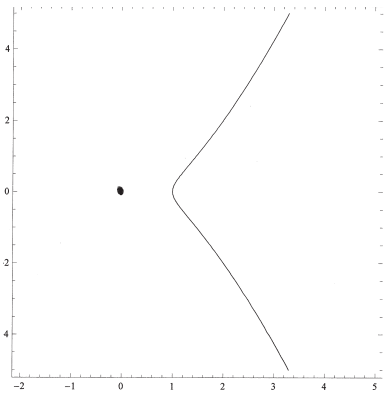
$$y^2 = x^3 + b_2x^2 + 2b_4x + b_6$$

as the general Weierstraβ equation, or , equivalently,

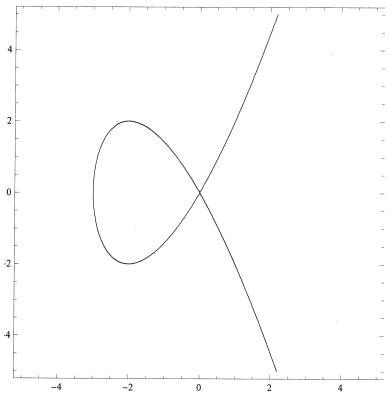
$$y^2 = f(x),$$

where, naturally, $f(x) = x^3 + b_2x^2 + 2b_4x + b_6$. (It should, by the way, be obvious that these curves are symmetric about the $x-$axis, intersecting it at the zeros of $f(x)$.)

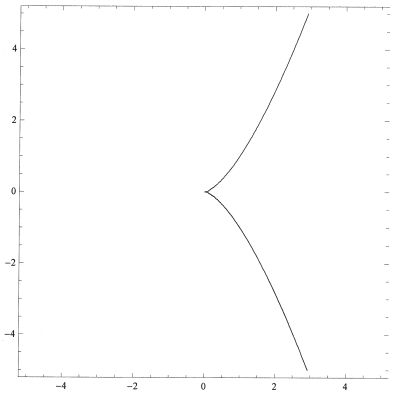
Studying a little more closely $f(x)$ with the methods of elementary calculus, if $\mathsf{k} = \mathbb{R}$, we realize that we have the following types of cubics:



$y^2 = x^2(x - \alpha)$ ($\alpha > 0$) (an *isolated* point at $x = 0$)

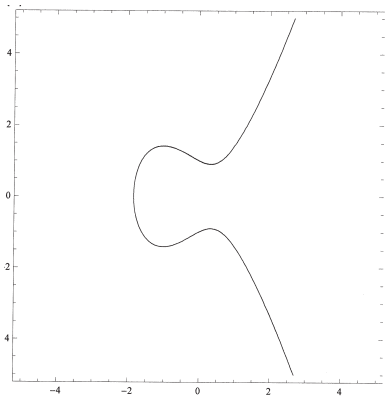


$y^2 = x^2(x - \alpha)$ ($\alpha < 0$) (a *node* at $x = 0$)

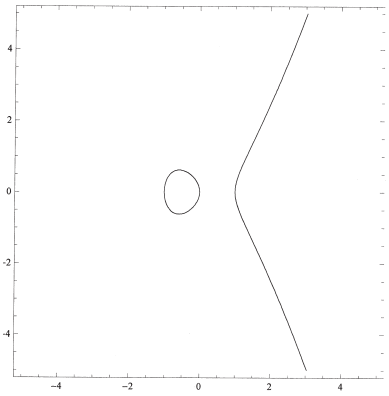


$y^2 = x^3$ (a *cusp* at $x = 0$)

The origin is a *singularity* in all of the above cases.



$y^2 = (x - \alpha)(x^2 + px + q)$ (α unique real root over $\mathbb{R}$)



$y^2 = (x-\alpha)(x-\beta)(x-\gamma)$ (three distinct real roots)

We can do even more if $char(\mathsf{k}) \neq 2$ or 3. Make the substitutions

$$\begin{cases} x = (x' - 3b_2)/36 \\ y = y'/108 \end{cases}$$

(*Nota bene*. $108 = 2^2 \cdot 3^2 \overset{2,3}{\equiv} 0$ and $36 = 2^2 \cdot 3^2 \overset{2,3}{\equiv} 0$).
We get:

$$\tfrac{y^2}{108^2} = 4\left(\tfrac{x-3b_2}{36}\right)^3 + b_2\left(\tfrac{x-3b_2}{36}\right)^2 + 2b_4\ \tfrac{x-3b_2}{36} + b_6$$

$$y^2 = \tfrac{108^2 \cdot 4}{36^3}(x^3 - 9b_2x^2 + 27b_2^2x - 27b_2^3) + \tfrac{108^2 \cdot b_2}{36^2}(x^2 - 6b_2x + 9b_2^2) +$$

$$\tfrac{108^2 \cdot 2b_4}{36}(x - 3b_2) + 108^2 b_6$$

$$y^2 = x^3 - 9b_2x^2 + 27b_2^2x - 27b_2^3 + 9b_2x^2 - 54b_2^2x + 81b_2^3 + 648b_4x - 1944b_2b_4 + 11664b_6$$

$$y^2 = x^3 - 27(b_2^2 - 24b_4)x - 54(-b_2^3 + 36b_2b_4 - 216b_6)$$

Setting

$$\begin{cases} c_4 = b_2^2 - 24b_4 \\ c_6 = -b_2^3 + 36b_2b_4 - 216b_6 \end{cases}$$

we can write

$$y^2 = x^3 - 27c_4x - 54c_6,$$

or, with an obvious change of notation,

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

This is the simplest, yet quite general, Weierstraβ equation giving the whole family of curves just as the initial equation did, as long as $char(\mathsf{k}) \neq 2$ or 3.

Since we shall work first over $\mathbb{R}$, and $char(\mathbb{R}) = 0$, everything is fine for now.

Furthermore we shall define the *discriminant* function

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

(This definition is highly arbitrary and hardly illuminating at this point but we shall leave it at that.)

If we start with a cubic and write the equation as $\mathsf{F}(x, y) = 0$ we know from the *implicit function theorem* that the curve in question will lack a tangent at points where $\nabla\mathsf{F} = 0$. Such points are called *singular*; the rest of them are *regular*. In other words, we can always find the set of singular points on a cubic by solving the system

$$\begin{cases} \mathsf{F}(x, y) = 0 \\ \nabla\mathsf{F}(x, y) = 0 \end{cases}$$

Let us now concentrate on the curves

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

Writing

$$\mathsf{F}(x, y) = y^2 - x^3 - ax - b$$

we get

$$\nabla\mathsf{F}(x, y) = (-3x^2 - a,\ 2y),$$

so $\nabla\mathsf{F}(x, y) = 0$ if and only if

$$\begin{cases} 3x^2 + a = f'(x) = 0 \\ 2y = 0 \iff y = 0 \end{cases}$$

Consequently, the singular points $(\alpha, 0)$ must satisfy the system

$$\begin{cases} f(\alpha) = 0 \\ f'(\alpha) = 0 \end{cases}$$

and this would mean that $x = \alpha$ is a double root of the polynomial function $f(x) = x^3 + ax + b$.

Therefore the following must hold:

$$x^3 + ax + b = (x - \alpha)^2 \, (x - \beta) = \ldots\ldots\ldots\ldots\ldots\ldots =$$

$$x^3 - (2\alpha + \beta)x^2 + (\alpha^2 + 2\alpha\beta)x - \alpha^2\beta,$$

which is possible if and only if

$$\begin{cases} 2\alpha + \beta = 0 \iff \beta = -2\alpha \\ \alpha^2 + 2\alpha\beta = a \\ -\alpha^2\beta = b \end{cases}$$

But the system

$$\begin{cases} \alpha^2 + 2\alpha\beta = a \\ \beta = -2\alpha \end{cases}$$

entails

$$a = -3\alpha^2$$

while the system

$$\begin{cases} -\alpha^2\beta = b \\ \beta = -2\alpha \end{cases}$$

entails

$$b = 2\alpha^3.$$

Using the above values for a and b the discriminant becomes

$\Delta = -16(4a^3 + 27b^2) = -16[4(-3\alpha^2)^3 + 27(2\alpha^3)^2] = -16[-4 \cdot 27\alpha^6 + 4 \cdot 27\alpha^6] = 0.$

So if the curve $y^2 = x^3 + ax + b$ is singular then the polynomial

$$f(x) = x^3 + ax + b$$

has a double root and we have just shown that this entails that $\Delta = 0$.

But, in fact, we might use a result in polynomial algebra which states generally that two polynomials

$$f = a_m x^m + a_{m-1}x^{m-1} + \ldots\ldots\ldots\ldots a_1 x + a_0 \ , \ a_m \neq 0$$

and

$$g = b_n x^n + b_{n-1}x^{n-1} + \ldots\ldots\ldots\ldots\ldots + b_1 x + b_0 \ , \ b_n \neq 0$$

have a non-constant common factor if and only if their *resultant*

$$\mathcal{R}(f, g) = 0.$$

The resultant is defined as the determinant of the matrix

$$\begin{bmatrix}
a_m & & & & b_n & & & \\
a_{m-1} & a_m & & & b_{n-1} & b_n & & \\
a_{m-2} & a_{m-1} & \ddots & & b_{n-2} & b_{n-1} & \ddots & \\
\vdots & & \ddots & a_m & \vdots & & \ddots & b_n \\
 & & & a_{m-1} & & \vdots & & b_{n-1} \\
a_0 & \vdots & & & b_0 & & & \\
 & a_0 & & \vdots & & b_0 & & \vdots \\
 & & \ddots & & & & \ddots & \\
 & & & a_0 & & & & b_0
\end{bmatrix}.$$

(n columns + m columns)

In our case we have

$$\mathcal{R}(f,f') = \det \begin{bmatrix}
1 & & 3 & & \\
0 & 1 & 0 & 3 & \\
a & 0 & a & 0 & 3 \\
b & a & & a & 0 \\
 & b & & & a
\end{bmatrix} = 4a^3 + 27b^2.$$

We realize that $\Delta = -16\mathcal{R}(f,f')$ and, consequently, $\Delta = 0$ if and only if $\mathcal{R}(f,f') = 0$ which is possible if and only if f and f' have a common non-constant factor. In our case this means that f has a repeated root.

Definition

An *elliptic curve* is the set of solutions to a Weierstraβ equation

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

together with an extra point $\mathcal{O}$, and such that $\Delta = -16(4a^3 + 27b^2) \neq 0$.

Remark: $\Delta = 0 \iff 4a^3 + 27b^2$. We shall not discuss the factor -16.

Question: What is this extra point $\mathcal{O}$? Going back to our last two examples, which are the only elliptic curves among all the five curves depicted there, we realize intuitively that any straight line L intersecting the elliptic curve E in points P and Q must intersect E in a third point R, at least for most pairs P and Q.

- What if $\mathsf{P} = \mathsf{Q}$? Then L is tangent to E at P and, if L is not vertical, it will meet E in another point which we shall consider the third point and say that P is a double point
- What if $\mathsf{P} = (a, b)$ and $\mathsf{Q} = (a, -b)$? Then L will be a vertical line. This can be the case even if $\mathsf{P} = \mathsf{Q}$. Let the third point be $\mathcal{O}$. We shall call it the *point at infinity*.

All this was a preparation for defining an operation $\oplus$ on the set of points of E, called *addition*, as follows:

Let P and Q be two points on E, not necessarily distinct. Let L be the line through P and Q if $\mathsf{P} \neq \mathsf{Q}$,or tangent to E at P, if $\mathsf{P} = \mathsf{Q}$. Then $\mathsf{L} \cap \mathsf{E}$ consists of three points counting multiplicities, say $\mathsf{L} \cap \mathsf{E} = \{\mathsf{P}, \mathsf{Q}, \mathsf{R}\}$. Writing $\mathsf{R} = (a, b)$ define the point $\ominus\mathsf{R} = (a, -b)$, the reflection of R across the x-axis. Surely $\ominus\mathsf{R} \in \mathsf{E}$. Define $\mathsf{P} \oplus \mathsf{Q} = \ominus\mathsf{R}$.

Theorem 10. *Let E be an elliptic curve (over $\mathbb{R}$). Then the addition law described above has the following properties:*

(a) (closure)

$$\forall\ \mathsf{P}, \mathsf{Q} \in \mathsf{E} \quad \mathsf{P} \oplus \mathsf{Q} \in \mathsf{E}$$

(b) (additive identity)

$$\exists \mathcal{O} \in \mathsf{E}\ \forall \mathsf{P} \in \mathsf{E} \quad \mathsf{P} \oplus \mathcal{O} = \mathcal{O} \oplus \mathsf{P} = \mathsf{P}$$

(c) (additive inverse)

$$\forall \mathsf{P} \in \mathsf{E}\ \exists\ \ominus\mathsf{P} \in \mathsf{E} \quad \mathsf{P} \oplus (\ominus\mathsf{P}) = (\ominus\mathsf{P}) \oplus \mathsf{P} = \mathcal{O}$$

(d) (commutativity)

$$\forall\ \mathsf{P}, \mathsf{Q}\ \epsilon\ \mathsf{E} \quad \mathsf{P} \oplus \mathsf{Q} = \mathsf{Q} \oplus \mathsf{P}$$

(e) (associativity)

$$\forall\ \mathsf{P}, \mathsf{Q}, \mathsf{R}\ \epsilon\ \mathsf{E} \quad (\mathsf{P} \oplus \mathsf{Q}) \oplus \mathsf{R} = \mathsf{P} \oplus (\mathsf{Q} \oplus \mathsf{R})$$

After the proof of this theorem we shall be able to assert safely that

$$< \mathsf{E}(\mathbb{R}), \oplus > \textit{ is an abelian additive group.}$$

Proof. Let $\mathsf{P} = (x_1, y_1)$ and $\mathsf{Q} = (x_2, y_2)$ in what follows, assuming $\mathsf{P} \neq \mathcal{O}$, $\mathsf{Q} \neq \mathcal{O}$. The case $\mathsf{P} = \mathcal{O}$ or $\mathsf{Q} = \mathcal{O}$ will be covered by (b) below.

(a) (*closure*)

If $x_1 = x_2$ and $y_1 = -y_2$, then $\mathsf{P} \oplus \mathsf{Q} = \mathcal{O}\ \epsilon\ \mathsf{E}$, by definition of $\oplus$. Otherwise, define

$$k = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & , \mathsf{P} \neq \mathsf{Q} \\ \frac{3x_1^2 + a}{2y_1} & , \mathsf{P} = \mathsf{Q} \end{cases}$$

and let

$$\begin{cases} x_3 = k^2 - x_1 - x_2 \\ y_3 = k(x_1 - x_3) - y_1 \end{cases}$$

Nota bene:

- The coordinates x_3 and y_3 are rational functions of the coordinates of P and Q .
- If $\mathsf{Q} = \ominus\mathsf{P}$, i.e. $\mathsf{Q} = (x, -y)$ then $k = \infty$ and L is vertical just as expected. This is the case even if $\mathsf{Q} = \ominus\mathsf{P} = \mathsf{P}$ ($= (x, 0)$).

We claim that $\mathsf{P} \oplus \mathsf{Q} = (x_3, y_3)$.

Set the equation of L as $y = kx + m$ and substitute this in the equation determining E. We get:

$$(kx + m)^2 = x^3 + ax + b.$$

This becomes, after expanding,

$$x^3 - k^2x^2 + (a - 2km)x + (b - k^2) = 0.$$

But this cubic has, as two of its roots, x_1 and x_2, and we shall call the third x_3.

Thus

$$x^3-k^2x^2+(a-2km)x+ (b-k^2) = (x - x_1)(x - x_2)(x - x_3) =$$

$$x^3-(x_1 + x_2 + x_3)x^2 + x_1x_2x + (x_1x_3+x_2x_3-x_1x_2x_3),$$

which is possible only if

$$k^2 = x_1+x_2+x_3$$

meaning that

$$x_3= k^2-x_1 - x_2$$

and, finally, we obtain :

$$-y_3 = kx_3 + m = kx_3+(y_1-kx_1) = k(x_3-x_1) + y_1$$

so

$$y_3 = k(x_1-x_3)-y_1$$

as desired.

We can be sure that $\mathsf{P} \oplus \mathsf{Q} = (x_3,\, y_3)\ \epsilon\ \mathsf{E}$.

Remark. We have actually given in this proof an algorithm for addition and after proving the whole theorem we shall write $+$ instead of $\oplus$ and $-$ instead of $\ominus$.

(b) (*additive identity*)

This is clear since $\mathcal{O}$ lies on every vertical line intersecting E. (I should like to mention here that we could have chosen $\mathcal{O}$ as an arbitrary point on E taking then reflections through $\mathcal{O}$).

(c) (*additive inverse*)

Set $\ominus\mathsf{P} = (x_1,\, -y_1)$, the reflection of P, and we are done.

(d) (*commutatitvity*)

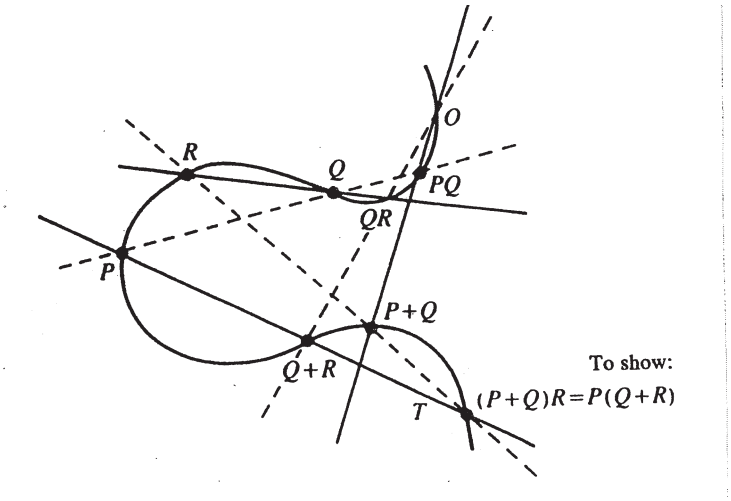
This is easy. The line through P and Q is identical with the line through Q and P so the third intersection point R is the same.

(e) (*associativity*)

Associativity is hardest to prove. An elegant proof of this should definitely be geometrical but it is far from trivial. One could try an algebraic

proof using the addition algorithm described in (a) above in some package as Mathematica, for instance. It is pure and simple horror !

I shall nonetheless try a proof sketch following Husemöller, 2000. Consider the following figure where we denote by PQ the third intersection point on the chord through P and Q , and likewise for the other pairs of points and have actually chosen $\mathcal{O}$ to be some visible point on E :



We have nine points: $\mathcal{O}$, P, Q, R, PQ, P + Q, QR, Q + R and, finally, the intersection T of the line joining P to Q + R with the line joining R to P + Q. The union of the three dotted lines and the union of the three solid lines are degenerate cubics each. By construction they pass through our nine points. On the other hand our curve E passes through the first eight points. A theorem of elementary algebraic geometry guarantees that it passes through the ninth as well, i.e. through T in our case. (See Ekedahl, 2003, or Fulton, 1989). But then (P + Q)R = P(Q + R) so we have (P + Q) + R = P + (Q + R), as desired.

We have only considered the case of three distinct points. The case P = Q could be treated as a limiting position in our figure. We shall not look into it.

□

4.2 Elliptic curves over finite fields

If we want to use elliptic curves in cryptography then we have to identify messages with points on elliptic curves. A message is a finite object so we shall only need finitely many points. Intuitively it is quite reasonable to

content ourselves with finite fields so now we shall switch from $\mathbb{R}$ to $\mathbb{F}_p$, p being a prime, of course.

Considering the fact that the number of points on $\mathsf{E}(\mathbb{F}_2)$ can at most be $\mathtt{5}$, namely $(\mathtt{0},\mathtt{0})$, $(\mathtt{0},\mathtt{1})$, $(\mathtt{1},\mathtt{0})$, $(\mathtt{1},\mathtt{1})$ and $\mathcal{O}$, it is obvious that we cannot do much cryptography with this; too few points, too few messages, if any! Analogously, $p = \mathtt{3}$ will not bring us salvation.

There is no reason to consider these cases yet, so we shall concentrate at present on $\mathbb{F}_p$, $p > \mathtt{3}$, and, consequently, consider only the Weierstraβ equation

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

Everything that we have said about elliptic curves over $\mathbb{R}$ carries over to curves over $\mathbb{F}_p$, $p > \mathtt{3}$.

Definition.

An elliptic curve over $\mathbb{F}_p$, $p > \mathtt{3}$, is the geometric locus determined by the Weierstraβ equation $y^2 = x^3 + ax + b$ with coefficients a, $b \,\epsilon\, \mathbb{F}_p$ such that $\Delta = -\mathtt{16}(\mathtt{4}a^3+\mathtt{27}b^2) \neq \mathtt{0}$ plus the *point at infinity*, $\mathcal{O}$.

We shall denote this locus

$$\mathsf{E}(\mathbb{F}_p) = \{(x, y) : x, y\epsilon\mathbb{F}_p \wedge y^2 = x^3 + ax + b\} \cup\{\mathcal{O}\}.$$

We have been considering up to now the Weierstraβ equation $y^2 = x^3+ax+b$ which we obtained from the initial equation by a first substitution $y = \frac{1}{2}(y' - a_1x' - a_3)$.

One can show that the substitutions that we have used are special cases of the pair of substitutions

$$\begin{cases} x = u^2x' + r \\ y = u^3y' + su^2x' + t \end{cases}$$

where $r, s, t, u \,\epsilon\, \mathsf{k}$.

Indeed, we can even envisage the situation where r, s, t belong to some (commutative) ring R (with unity) and u is a unit in R. (See Ekedahl or Silverman). These transformations form a group under composition and the orbits under this group are the equivalence classes of the curves.

Recalling the coefficients b_i defined before we can add a new one to the list, viz.

$$b_8 = a_1^2a_6 - a_1a_3a_4 + 4a_2a_6 + a_2a_3^2 - a_4^2$$

and define the *discriminant* as

$$\Delta = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6 \ .$$

If $char(\mathsf{k}) \neq 2,3$ then, considering the class $y^2 = x^3 + ax + b$, we have $a_1 = a_2 = a_3 = 0$, hence $b_2 = 0$, $b_4 = 2a_4 = 2a$ and $b_6 = 4a_6 = 4b$, so

$$\Delta = -8b_4^3 - 27b_6^2 = -8\cdot(2a_4)^3 - 27\cdot(4a_6)^2 = -16(4a^3 + 27b^2)$$

which is the discriminant we defined initially.

If $char(\mathsf{k}) = 2$ then an elliptic curve is the set of solutions to one of the following two equations:

$$y^2 + cy = x^3 + ax + b \qquad \text{or} \qquad y^2 + xy = x^3 + ax^2 + b \ .$$

(See Koblitz, 1998, A Course in Number Theory and Cryptography, pg 168)

We shall not pursue this further but instead shall reconsider the most general Weierstraβ equation:

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

Let us assure ourselves that we can define addition just as before.

Theorem 11. *Theorem 10 holds for* $\mathsf{E}(\mathbb{F}_p)$.

Proof. Set $\mathsf{P} = (x, y)$ and $-\mathsf{P} = (x, y')$. If $y' = y$ then $\mathsf{P} = -\mathsf{P}$ and $\mathsf{P} - \mathsf{P} = \mathcal{O}$, so assume that $y' \neq y$. Since P, $-\mathsf{P}$ ϵ E the following must hold:

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

Hence:

$$(y' + y + a_1x + a_3)(y' - y) = 0$$

and therefore

$$y' = -y - a_1x - a_3.$$

We have just shown that in the general case the reflection rule should be $(x, y) \longmapsto (x, -y - a_1x - a_3)$

and, therefore, $-\mathsf{P} = (x, -y - a_1x - a_3)$.

Let $\mathsf{P} = (x_1, y_1)$ and $\mathsf{Q} = (x_2, y_2)$ be points on E. Let L be the line through these point. The equation for L can, of course, be written as

$$y = kx + m$$

and we get:

$$(kx + m)^2 + a_1x(kx + m) + a_3(kx + m) =$$

$$x^3 + a_2x^2 + a_4x + a_6$$

Expanding and simplifying we get

$$x^3 + (a_2 - a_1k - k^2)x^2 + (a_4 + a_3k + a_1m + 2km)x + (a_6 + a_3m + m^2) = 0.$$

This, of course, is a monic cubic equation in x, and, as such, can be factorized in the form $(x - x_1)(x - x_2)(x - x_3) = 0$. Setting the two expressions equal we have

$$x^3 + (a_2 - a_1k - k^2)x^2 + (a_4 + a_3k + a_1m + 2km)x + (a_6 + a_3m + m^2) =$$

$$x^3 - (x_1 + x_2 + x_3)x^2 + x_1x_2x + (x_1x_3 + x_2x_3 - x_1x_2x_3) \ (\dagger)$$

with x_3 and y_3' are the coordinates of R , the third point of intersection of L with E. This is possible only if

$$x_3 = k^2 + a_1k - a_2 - x_1 - x_2.$$

As for y_3 , we know that $y_3 = -y_3' - a_1x_3 - a_3$, but $y_3' = kx_3 + m$, and so we get:

$$y_3 = -(k + a_1)x_3 - m - a_3 \ .$$

Of course we must define k.

If $x_1 \neq x_2$ we define

$$\begin{cases} k = \frac{y_2 - y_1}{x_2 - x_1} \\ m = y_1 - kx = (y_1x_2 - y_2x_1)/(x_2 - x_1) \end{cases}$$

If $x_1 = x_2$ we know from elementary calculus that, given the curve $\mathsf{F}(x, y) = 0$, the slope of the tangent at that point is $k = \frac{\mathsf{F}_x}{-\mathsf{F}_y}$. In our case we have

$\mathsf{F}(x, y) = x^3 + a_2x^2 + a_4x + a_6 - y^2 - a_1xy - a_3y$.

This yields, after some computations and simplifications,

$$\begin{cases} k = (3x_1^2 + 2a_2x_1 + a_4 - a_1y_1)/(2y_1 + a_1x_1 + a_3) \\ m = (-x_1^3 + a_4x_1 + 2a_6 - a_3y_1)/(2y_1 + a_1x_1 + a_3) \end{cases}$$

We conclude:

$$\mathsf{P} + \mathsf{Q} = (k^2 + a_1k - a_2 - x_1 - x_2,\ -(k + a_1)x_3 - m - a_3\ ).$$

If you thought that the addition algorithm for $char(\mathsf{k}) \neq 2,\ 3$ was quite cumbersome, what about this ?

Could something go wrong if $\mathsf{P} = \mathsf{Q}$?

I would like to formulate the problem like this: Given an elliptic curve $\mathsf{E}(\mathsf{k})$ and a straight line L , non-collinear with the point at infinity $\mathcal{O}$, such that $\mathsf{L} \cap \mathsf{E} \neq \emptyset$, what is the cardinality of $\mathsf{L} \cap \mathsf{E}$? There arise three theoretical situations:

- $| \mathsf{L} \cap \mathsf{E} | = 3$

  $\mathsf{L} \cap \mathsf{E} = \{\mathsf{P}, \mathsf{Q}, \mathsf{R}\}$

  We have three distinct points and no problems.

- $| \mathsf{L} \cap \mathsf{E} | = 2$

  $\mathsf{L} \cap \mathsf{E} = \{\mathsf{P}, \mathsf{R}\}$

  We have a double point, because our cubic equation must have three roots counting multiplicities.

- $| \mathsf{L} \cap \mathsf{E} | = 1$

  This means that the cubic equation in question has a root $x_1 \in \mathsf{k}$ and two roots $x_2,\ x_3 \notin \mathsf{k}$ and we might have a problem! Let us look into it.

We know that x_1 is a root of the equation $(x - x_1)(x - x_2)(x - x_3) = 0$ Comparing coefficients in the expression (†) above we realize that it must hold that $x_1x_2 = a_4 + a_3k + a_1m + 2km \in \mathbb{F}_p^\star$. Since $x_1 \in \mathbb{F}_p^\star$, this entails that $x_2 = x_1^{-1} \cdot x_1x_2 \in \mathbb{F}_p^\star$, and then surely $x_3 \in \mathbb{F}_p^\star$ too. So our equation has either three distinct roots in $\mathbb{F}_p^\star$, which cannot be the case since we have chosen $\mathsf{P} = \mathsf{Q}$, or it has one double root and that is $x_1 = x_2$, the coordinate of P, while x_3 is the x-coordinate of R such that $2\mathsf{P} + \mathsf{R} = \mathcal{O}$. So, $| \mathsf{L} \cap \mathsf{E} | = 1$ is simply impossible and the answer to our former question is that nothing can go wrong if $\mathsf{P} = \mathsf{Q}$!

□

Nota bene. Although the associative law can be proved as previously using the addition algorithm it would require, besides tedious calculations, considering many special cases. One solution would be to use Bézout's Theorem but this requires a more advanced set up. Hoffstein, Pipher and Silverman refer to more elegant and advanced proofs presented in S. Lang, J.H.Silverman or J.H.Silverman & J. Tate.

Definition.

We call a an nth *power residue* mod m if the congruence $x^n \equiv a \pmod{m}$ is solvable.

Theorem 12. *If m possesses primitive roots and* gcd *$(a,\ m) = 1$, then a is an nth power residue mod m iff $a^{\varphi(m)/d} \equiv 1$, where $d =$* gcd *$(n,\ \varphi(m))$.*

Proof. Let g be a primitive root mod m and $a = g^b$, $x = g^y$. Then the congruence $x^n \equiv a \pmod{m}$ is equivalent to $g^{ny} \equiv g^b \pmod{m}$, which in turn is equivalent to $ny \equiv b \bmod (\varphi(m))$. The latter congruence is solvable iff $d \mid b$. *Nota bene*: if there is a solution then there are exactly d solution. In particular, if $n = 2$ then the congruence $x^2 \equiv a \pmod{m}$ has at most two solutions.

If $d \mid b$, then $a^{\varphi(m)/d} \equiv g^{b\varphi(m)/d} \equiv 1 \pmod{m}$. Conversely, if $a^{\varphi(m)/d} \equiv 1 \pmod{m}$, then $g^{b\varphi(m)/d} \equiv 1 \pmod{m}$, which implies that $\varphi(m) \mid b\varphi(m)/d$ or $d \mid b$. The result follows.

□

Denoting the cardinality of the set of points on $\mathsf{E}(\mathbb{F}_p)$ by $\#\ \mathsf{E}(\mathbb{F}_p)$ we realize that it must be a finite quantity since we have p possibilities for the x-coordinate and, since $y^2 = x^3 + ax + b$, we get, by Theorem 2, at most two possibilities for the y-coordinate once x is chosen. The conclusion must be that

$$\#\ \mathsf{E}(\mathbb{F}_p) \leq \quad 2p + 1, \text{ the } 1 \text{ because of the point } \mathcal{O}.$$

This upper bound is much larger than the true value of $\#\ \mathsf{E}(\mathbb{F}_p)$.

Pick a value for x and compute $x^3 + ax + b$. This quantity may be a quadratic residue modulo p in which case we get two square roots, meaning two distinct values for y and, thus, two points on $\mathsf{E}(\mathbb{F}_p)$. But how often does this happen? Since there are as many residues as nonresidues according to a result in number theory, 50% of the time.

Another possibility is that $x_0^3 + ax_0 + b = 0$ and so we get only the point $(\mathtt{x}_0,0)$ but this rarely happens, according to Hoffstein, Pipher & Silverman. Altogether this might indicate that we should expect

$$\#\ \mathsf{E}(\mathbb{F}_p) \approx 0.5{\cdot}2p + 1 = p + 1.$$

Theorem 13. *(Hasse) Let E be an elliptic curve over $\mathbb{F}_p$. Then*
$\#\ \mathsf{E}(\mathbb{F}_p) = p + 1 - \mathsf{t}_p$, with t_p satisfying $\mid \mathsf{t}_p \mid \leq 2\sqrt{p}$.
The quantity t_p is called the trace of Frobenius for $\mathsf{E}(\mathbb{F}_p)$.

Proof. For a proof of this I refer to Silverman,1992. □

Hasse's theorem can be proved even for $\mathsf{E}(\mathbb{F}_{p^k})$ stating then that

$$\#\ \mathsf{E}(\mathbb{F}_{p^k}) = p^k + 1 - \mathsf{t}_{p^k}$$

with t_{p^k} satisfying $\mid \mathsf{t}_{p^k} \mid \leq 2p^{\frac{k}{2}}$. Hasse's theorem gives a bound for the quantity $\#\ \mathsf{E}(\mathbb{F}_p)$ but it is not as such constructive.

Example. We shall consider the eliptic curve

$$\mathsf{E}(\mathbb{F}_{11}) : y^2 = x^3 + x + 1.$$

We remark that $\triangle = -16(4\mathsf{a}^3 + 27\mathsf{b}^2) = -16(4 \cdot 1^3 + 27 \cdot 1^2) \equiv 6 \cdot 9 \equiv 10$ so E is a nonsingular cubic and therefore elliptic.

We have just seen that $\#\ \mathsf{E}(\mathbb{F}_{11}) \leq 2 \cdot 11 + 1 = 23$. We could compute by hand but I have used a Mathematica package.

The result is

$$\mathsf{E}(\mathbb{F}_{11}) = \{\mathcal{O},\ (0,1),\ (0,10),\ (1,5),\ (1,6),\ (2,0),\ (3,3),\ (3,8),\ (4,5),\ (4,6),\ (6,5),\ (6,6),(8,2),\ (8,9)\},$$

so $\#\ \mathsf{E}(\mathbb{F}_{11}) = 14$ and we can compute $\mid \mathsf{t}_{11} \mid = \mid 11 + 1 - 14 \mid = \mid -2 \mid = 2 \leq 2\sqrt{11}$.

Let us compute (0,1) + (3,3) and (1,5) + (1,5) by hand.

First, (0,1) + (3,3):

$$k = \frac{y_2 - y_1}{x_2 - x_1} = \frac{3-1}{3-0} = \frac{2}{3} \equiv 2 \cdot 3^{-1} \equiv 2 \cdot 4 = 8$$

$$x_3 = k^2 - x_1 - x_2 = 8^2 - 0 - 3 \equiv 6$$

$$y_3 = k(x_1 - x_3) - y_1 = 8 \cdot (0 - 6) - 1 = 6$$

The result is (0,1) + (3,3) = (6,6).

Second, (1,5) + (1,5):

$$k = \frac{3\mathsf{x}_1^2 + 1}{2\mathsf{y}_1} = \frac{3 \cdot 1^2 + 1}{2 \cdot 5} = \frac{4}{10} \equiv 4 \cdot 10^{-1} \equiv 4 \cdot 10 \equiv 7$$

$$x_3 = k^2 - 2x_1 = 7^2 - 2 \cdot 1 \equiv 3$$

$$y_3 = k(x_1 - x_3) - y_1 = 7 \cdot (1 - 3) - 5 = 3$$

The result is (1,5) + (1,5) = (3,3).

Using again Mathematica I have determined the complete addition table:

$\oplus$	$\mathcal{O}$	$(0,1)$	$(0,10)$	$(1,5)$	$(1,6)$	$(2,0)$	$(3,3)$
$\mathcal{O}$	$\mathcal{O}$	$(0,1)$	$(0,10)$	$(1,5)$	$(1,6)$	$(2,0)$	$(3,3)$
$(0,1)$	$(0,1)$	$(3,3)$	$\mathcal{O}$	$(4,5)$	$(2,0)$	$(1,5)$	$(6,6)$
$(0,10)$	$(0,10)$	$\mathcal{O}$	$(3,8)$	$(2,0)$	$(4,6)$	$(1,6)$	$(0,1)$
$(1,5)$	$(1,5)$	$(4,5)$	$(2,0)$	$(3,3)$	$\mathcal{O}$	$(0,1)$	$(8,2)$
$(1,6)$	$(1,6)$	$(2,0)$	$(4,6)$	$\mathcal{O}$	$(3,8)$	$(0,10)$	$(1,5)$
$(2,0)$	$(2,0)$	$(1,5)$	$(1,6)$	$(0,1)$	$(0,10)$	$\mathcal{O}$	$(4,5)$
$(3,3)$	$(3,3)$	$(6,6)$	$(0,1)$	$(8,2)$	$(1,5)$	$(4,5)$	$(6,5)$
$(3,8)$	$(3,8)$	$(0,10)$	$(6,5)$	$(1,6)$	$(8,9)$	$(4,6)$	$\mathcal{O}$
$(4,5)$	$(4,5)$	$(8,2)$	$(1,5)$	$(6,6)$	$(0,1)$	$(3,3)$	$(8,9)$
$(4,6)$	$(4,6)$	$(1,6)$	$(8,9)$	$(0,10)$	$(6,5)$	$(3,8)$	$(2,0)$
$(6,5)$	$(6,5)$	$(3,8)$	$(6,6)$	$(4,6)$	$(8,2)$	$(8,9)$	$(0,10)$
$(6,6)$	$(6,6)$	$(6,5)$	$(3,3)$	$(8,9)$	$(4,5)$	$(8,2)$	$(3,8)$
$(8,2)$	$(8,2)$	$(8,9)$	$(4,5)$	$(6,5)$	$(3,3)$	$(6,6)$	$(4,6)$
$(8,9)$	$(8,9)$	$(4,6)$	$(8,2)$	$(3,8)$	$(6,6)$	$(6,5)$	$(1,6)$

$\oplus$	$(3,8)$	$(4,5)$	$(4,6)$	$(6,5)$	$(6,6)$	$(8,2)$	$(8,9)$
$\mathcal{O}$	$(3,8)$	$(4,5)$	$(4,6)$	$(6,5)$	$(6,6)$	$(8,2)$	$(8,9)$
$(0,1)$	$(0,10)$	$(8,2)$	$(1,6)$	$(3,8)$	$(6,5)$	$(8,9)$	$(4,6)$
$(0,10)$	$(6,5)$	$(1,5)$	$(8,9)$	$(6,6)$	$(3,3)$	$(4,5)$	$(8,2)$
$(1,5)$	$(1,6)$	$(6,6)$	$(0,10)$	$(4,6)$	$(8,9)$	$(6,5)$	$(3,8)$
$(1,6)$	$(8,9)$	$(0,1)$	$(6,5)$	$(8,2)$	$(4,5)$	$(3,3)$	$(6,6)$
$(2,0)$	$(4,6)$	$(3,3)$	$(3,8)$	$(8,9)$	$(8,2)$	$(6,6)$	$(6,5)$
$(3,3)$	$\mathcal{O}$	$(8,9)$	$(2,0)$	$(0,10)$	$(3,8)$	$(4,6)$	$(1,6)$
$(3,8)$	$(6,6)$	$(2,0)$	$(8,2)$	$(3,3)$	$(0,1)$	$(1,5)$	$(4,5)$
$(4,5)$	$(2,0)$	$(6,5)$	$\mathcal{O}$	$(1,6)$	$(4,6)$	$(3,8)$	$(0,10)$
$(4,6)$	$(8,2)$	$\mathcal{O}$	$(6,6)$	$(4,5)$	$(1,5)$	$(0,1)$	$(3,3)$
$(6,5)$	$(3,3)$	$(1,6)$	$(4,5)$	$(0,1)$	$\mathcal{O}$	$(2,0)$	$(1,5)$
$(6,6)$	$(0,1)$	$(4,6)$	$(1,5)$	$\mathcal{O}$	$(0,10)$	$(1,6)$	$(2,0)$
$(8,2)$	$(1,5)$	$(3,8)$	$(0,1)$	$(2,0)$	$(1,6)$	$(0,10)$	$\mathcal{O}$
$(8,9)$	$(4,5)$	$(0,10)$	$(3,3)$	$(1,5)$	$(2,0)$	$\mathcal{O}$	$(0,1)$

I am not going to repeat the procedure for some $\mathsf{E}(\mathbb{F}_{p^k})$ too but it should be obvious what we have to do !

What if $p = 2, 3$?

Computers work binarily so computations tend to become smoother modulo 2 and therefore the case $p = 2$ may be interesting. On the other hand, $p = 3$ does not present this advantage so we shall not consider it!

I have already mentioned that $\#\, \mathsf{E}(\mathbb{F}_2) \leq 5$ so, since we need $\#\, \mathsf{E}$ to be "slightly" larger than that, we shall consider elliptic curves $\mathsf{E}(\mathbb{F}_{2^k})$, $k > 1$.

To reassume: one advantage of working with $\mathsf{E}(\mathbb{F}_{2^k})$ is ease of computations modulo 2, particularly congenial for computers. Choosing k composite $\mathbb{F}_{2^j}$ is a subfield of $\mathbb{F}_{2^k}$ for all $j \mid k$. These subfields can sometimes be used to speed up computations. This is another advantage but, of course, these subfields can cause security problems too. There is though a third advantage, namely using an elliptic curve with coefficients in $\mathbb{F}_2$ but points with coordinates in $\mathbb{F}_{2^k}$, a so called Koblitz curve.

Definition

The $(p - power)$ *Frobenius map*:

$$\tau\colon \mathbb{F}_{p^k} \longrightarrow \mathbb{F}_{p^k}$$

$$\alpha \longrightarrow \alpha^p.$$

$\tau(\alpha\beta) = (\alpha\beta)^p = \alpha^p\beta^p = \tau(\alpha)\tau(\beta)$. ($\tau(1) = 1$, trivial.) This shows at once that the Frobenius map preserves multiplication. The binomial theorem says that

$$(\alpha + \beta)^n = \sum_i \binom{n}{i} \alpha^i \beta^{n-i},\ n \geq 1.$$

$\binom{n}{i} = \frac{n!}{i!(n-i)!}$ and, if $n = p$, a prime, then $p \mid p!$, yet, for $1 < i < p$, $p \nmid i!(p - i)!$ because these products are all strictly less than p. Consequently $p \mid \binom{p}{i}$, for $1 < i < p$ and, hence, we are bound to have:

$$(\alpha + \beta)^p = \alpha^p + \beta^p.$$

Of course, we shall only need the case $p = 2$ which is immediate since $(\alpha + \beta)^2 = \alpha^2 + 2\alpha\beta + \beta^2 \overset{2}{\equiv} \alpha^2 + \beta^2$.

Actually, by induction, we get even $(\alpha + \beta)^{p^k} = \alpha^{p^k} + \beta^{p^k}$. In any case, we have

$$\tau(\alpha + \beta) = (\alpha + \beta)^p = \alpha^p + \beta^p = \tau(\alpha) + \tau(\beta),$$

so τ preserves addition too. ($\tau(\mathsf{0}) = \mathsf{0}$, trivial.)

$\therefore \tau$ is a ring homomorphism.

Returning to the case $p = \mathsf{2}$, $\tau(\alpha) = \alpha^2$, but $\alpha \in \mathbb{F}_2 = \{0, 1\}$. This means that for every $\alpha \in \mathbb{F}_2$ it holds that $\tau(\alpha) = \alpha$, so $\mathbb{F}_2$ is invariant under τ.

Let E be an elliptic curve given by a generalized Weierstraβ equation with coefficients in $\mathbb{F}_2$ but points $\mathsf{P} = (x, y) \in \mathbb{F}_{2^k} \times \mathbb{F}_{2^k}$. (We shall write $\mathsf{P} \in \mathsf{E}(\mathbb{F}_{2^k})$). We define a Frobenius map on E by $\tau(\mathsf{P}) = (\tau(x),\tau(y))$.

Claim 1.

$\tau(\mathsf{P}) \in \mathsf{E}(\mathbb{F}_{2^k})$

Proof. Given $\mathsf{P} = (x, y) \in \mathsf{E}(\mathbb{F}_{2^k})$ and

$$\mathsf{E}\text{: } y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 = \mathsf{0}$$

we have:

$$\tau(y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6\ ) = \tau(\mathsf{0})$$

$$\tau(y)^2 + \tau(a_1)\tau(x)\tau(y) + \tau(a_3)\tau(y) - \tau\,(x)^3 - \tau(a_2)\tau(x)^2 - \tau(a_4)\tau(x) - \tau(a_6) = \mathsf{0}$$

The Weierstraβ equation has coefficients in $\mathbb{F}_2$ and $\mathbb{F}_2$ is τ-invariant, so:

$$\tau(y)^2 + a_1\tau(x)\tau(y) + a_3\tau(y) - \tau\,(x)^3 - a_2\tau(x)^2 - a_4\tau(x) - a_6 = \mathsf{0}$$

But this means that $\tau(\mathsf{P}) = (\tau(x), \tau(y))$ satisfies our Weierstraβ equation

$\therefore \tau(\mathsf{P}) \in \mathsf{E}(\mathbb{F}_{2^k})$ as claimed.

□

Claim 2.

$\tau(\mathsf{P} + \mathsf{Q}) = \tau(\mathsf{P}) + \tau(\mathsf{Q})$.

Proof. Recall first that τ is a ring homomorphism. (In fact it is a field homomorphism.) If u is a unit we have

$\tau(u^{-1})\cdot\tau(u) = \tau(u^{-1}u) = \tau(\mathsf{1}) = \mathsf{1}$.

$\therefore (\tau(u))^{-1} = \tau(u^{-1})\cdot$

$$\mathsf{P} + \mathsf{Q} = (k^2 + a_1k - a_2 - x_1 - x_2,\ -(k + a_1)x_3 - m - a_3\ )$$

with k and m rational expressions of elements in $\mathbb{F}_{2^k}$.

$$\tau(\mathsf{P}+\mathsf{Q}) = (\tau(k^2+ a_1k -a_2-x_1-x_2) \ , \tau(-(k+a_1)x_3-m- a_3)) =$$

$$(\tau(k)^2+ a_1\tau(k)\ a_2 - \tau(x_1) -\tau(x_2), -(\tau(k)+a_1)\tau(x_3) -\tau(m) -a_3)$$

$$\tau(\mathsf{P}) + \tau(\mathsf{Q}) = (\hat{k}^2 + a_1\hat{k} -a_2 -\tau(x_1) -\tau(x_2), -(\hat{k} + a_1)\tau(x_3) -\hat{m} -a_3)$$

$$\hat{k} = \tfrac{\tau(y_2)-\tau(y_1)}{\tau(x_2)-\tau(x_1)} = \tfrac{\tau(y_2-y_1)}{\tau(x_2-x_1)} = \tau(y_2-y_1)(\tau(x_2-x_1))^{-1}$$

$$\tau(y_2-y_1)\tau\ ((x_2-x_1)^{-1}) = \tau\Big(\tfrac{y_2-y_1}{x_2-x_1}\Big) = \tau(k)$$

or

$$\hat{k} = \tfrac{3\tau(x_1)^2+2a_2\tau(x_1)+a_4-a_1\tau(y_1)}{2\tau(y_1)+a_1\tau(x_1)+a_3} = \tfrac{\tau(3x_1^2+2a_2x_1+a_4-a_1y_1)}{\tau(2y_1+a_1x_1+a_3} = ... = \tau(k)$$

Analogously one can show that $\hat{m} = \tau(m)$. Hence we get:

$$\tau(\mathsf{P}) + \tau(\mathsf{Q}) =$$

$$(\tau(k)^2 + a_1\tau(k) -a_2 -\tau(x_1) -\tau(x_2), -(\tau(k) + a_1)\tau(x_3) -\tau(m) -a_3) =$$

$$\tau(\mathsf{P} + \mathsf{Q})$$

as claimed. □

I formulate the following theorem just for the record.

Theorem 14. *Let* E *be an elliptic curve over* $\mathbb{F}_p$ *and let* $\mathsf{t}=p+1-\#\mathsf{E}(\mathbb{F}_p)$.
(a) Let α *and* β *be the complex roots of the polynomial* $z^2-tz + p$.
Then $|\alpha| = |\beta| = \sqrt{p}$, *and, for every* $k \geq 1$, *we have*

$$\#\mathsf{E}(\mathbb{F}_p) = p^k + 1 - \alpha^k - \beta^k.$$

(b) Let $\tau : \mathsf{E}(\mathbb{F}_{p^k}) \longrightarrow \mathsf{E}(\mathbb{F}_{p^k})$ *be the Frobenius map.*
Then for every point $\mathsf{Q} \in \mathsf{E}(\mathbb{F}_{p^k})$ *we have* $\tau^2(\mathsf{Q}) - \mathsf{t}\cdot\tau(\mathsf{Q}) + p\cdot\mathsf{Q} = 0$.

Proof. For a proof see J. H: Siverman, 1992 , The Arithmetic of Elliptic Curves, (V.2, Prop. 2.3, pp 134-136) □

We shall see later that in trying to solve the ECDLP we shall need to compute $n\mathsf{P}$. Using the Frobenius map we shall save time!

Definition.

A *Koblitz curve* is an elliptic curve defined over $\mathbb{F}_2$ by an equation of the form

$$\mathsf{E}_a\colon\ y^2 + xy = x^3 + ax^2 + 1,\ a \in \{0, 1\}.$$

The discriminant of E_a is $\Delta = 1$.

Let us consider the example

$$\mathsf{E}_0\colon\ y^2 + xy = x^3 + 1.$$

We check by brute force that

$$\mathsf{E}_0(\mathbb{F}_2) = \{(0,1), (1,0), (1,1), \mathcal{O}\}\ ,$$

so $\#\mathsf{E}_0(\mathbb{F}_2) = 4$ and we get $\mathsf{t}_2 = 2 + 1 - 4 = -1$.

Applying the previous theorem we solve the equation $z^2 + z + 2 = 0$ and get the roots

$$\begin{cases} \alpha = \frac{-1+\sqrt{-7}}{2} \\ \beta = \frac{-1-\sqrt{-7}}{2} \end{cases}$$

so $\#\mathsf{E}_0(\mathbb{F}_{2^k}) = 2^k + 1 - \left(\frac{-1+\sqrt{-7}}{2}\right)^k - \left(\frac{-1-\sqrt{-7}}{2}\right)^k$

The same theorem says that the Frobenius map satisfies the equation $\tau^2 + \tau + 2 = 0$ when it acts on points of $\mathsf{E}(\mathbb{F}_{2^k})$ and this means that $\tau^2(\mathsf{P}) + \tau(\mathsf{P}) + 2\mathsf{P} = \mathcal{O}$ for all $\mathsf{P} \in \mathsf{E}(\mathbb{F}_{2^k})$.

Write now an arbitrary integer n as a sum of powers of τ under the assumption $\tau^2 + \tau + 2 = 0$, e.g.

$$n = v_0 + v_1\tau + v_2\tau^2 + \ldots\ldots + v_m\tau^m,\ v_i \in \{-1, 0, 1\}.$$

Then we can compute

$$n\mathsf{P} = (v_0 + v_1\tau + v_2\tau^2 + \ldots\ldots + v_m\tau^m)\mathsf{P} =$$

$$v_0\mathsf{P} + v_1\tau(\mathsf{P}) + v_2\tau^2(\mathsf{P}) \ldots\ldots\ldots + v_m\tau^m(\mathsf{P}).$$

All this will take less time because it is easier to compute $\tau^j(\mathsf{P})$ than $2^j\mathsf{P}$ since

$$\tau^j(\mathsf{P}) = \tau^j((x,y)) = (\tau^j(x), \tau^j(y)) = (x^{2^j}, y^{2^j}),$$

so we need only multiply in $\mathbb{F}_{2^k}$ not add in $\mathsf{E}(\mathbb{F}_{2^k})$.

4.3 Torsion. Rational functions. Divisors

Torsion points.

Let $m \geq 1$ be an integer. A point $\mathsf{P} \in \mathsf{E}$ satisfying $m\mathsf{P} = \mathcal{O}$, m being the least such integer, is called a point of order m in the group E . We write $\mathsf{E}[m] = \{\mathsf{P} \in \mathsf{E}\colon [m]\,\mathsf{P} = \mathcal{O}\}$. These points are called points of finite order or *torsion* points. Suppose that P and Q belong to $\mathsf{E}[m]$. Then:

$$m(\mathsf{P} + \mathsf{Q}) = m\mathsf{P} + m\mathsf{Q} = \mathcal{O} \qquad \because \mathsf{P} + \mathsf{Q} \in \mathsf{E}[m]$$

$$m(-\mathsf{P}) = -m\mathsf{P} = \mathcal{O} \qquad \because -\mathsf{P} \in \mathsf{E}[m]$$

The conclusion is that $\mathsf{E}[m]$ is a subgroup of E.

Theorem 15. *Let $m \geq 1$ be an integer.*

(a) Let E be an elliptic curve over $\mathbb{Q}$, $\mathbb{R}$ or $\mathbb{C}$. Then $\mathsf{E}[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$.

(b) Let E be an elliptic curve over $\mathbb{F}_p$ and assume that p does not divide m. Then there exists a value k such that

$\mathsf{E}(\mathbb{F}_{\mathsf{p}^{\mathsf{jk}}})[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$ for all integers $j \geq 1$.

Proof. For a proof I refer to Silverman, 1992, (Cor. III 6.4) □

If m is a prime and k is a field such that $\mathsf{E}(\mathsf{k})[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$ then $\mathsf{E}[m]$ can be viewed as a 2–dimensional vector space over the field $\mathbb{Z}/m\mathbb{Z}$ $\cong \mathbb{Z}_m$. And even in case m is not a prime it can be proved that we still can find a basis $= \{\mathsf{P}_1, \mathsf{P}_2\}$ in the sense that every point $\mathsf{P} \in \mathsf{E}[m]$ can be represented as a linear combination $\mathsf{P} = a\mathsf{P}_1 + b\mathsf{P}_2$ for unique coefficients $a,\ b \in \mathbb{Z}_m$.

Nota bene. If m is very large it can be difficult to find a and b. If $\mathsf{P} = a\mathsf{P}_1$ then finding a amounts to solving the ECDLP for P and P_1.

Rational functions.

Let k be a field. A *rational function* in x_1, x_2, ..., x_m with coefficients in k is a quotient $\frac{f}{g}$ of two polynomials $f,\ g \in \mathsf{k}[x_1, x_2, ..., x_m]$ where g is not the zero polynomial. Furthermore, two rational functions $\frac{f}{g}$ and $\frac{h}{l}$ are equal provided that $lf = gh$ in $\mathsf{k}[x_1, x_2, ..., x_m]$. Finally, the set of all rational functions in x_1, x_2, ..., x_m with coefficients in k is denoted $\mathsf{k}(x_1, x_2, ..., x_m)$. For simplicity we shall consider first the case $\mathbb{C}(x)$.

Allowing complex numbers we can factorize numerator and denominator and write

$$f(x) = \frac{a(x-\alpha_1)^{e_1}(x-\alpha_2)^{e_2}.....(x-\alpha_r)^{e_r}}{b(x-\beta_1)^{d_1}(x-\beta_2)^{d_2}.....(x-\beta_s)^{d_s}}$$

and we can assume that α_j , $1 \leq j \leq r$, and β_i , $1 \leq i \leq s$, are distinct numbers because otherwise we might cancel certain factors. The numbers α_j are called the *zeros* of f whereas the numbers β_i are called the *poles* of f and the exponents e_j and d_i , respectively, are the associated *multiplicities*.

Divisors.

The formal sum

$$\text{div } f = e_1[\alpha_1] + ... + e_r[\alpha_{1r}] - d_1[\beta_1] - ..... - d_s[\beta_s]$$

is called the *divisor* of f.

We can extend all these definitions to functions of two variables so if E is the elliptic function given by $y^2 = x^3 + ax + b$ we can consider a rational function $f(x, y)$ on E. Even in this case we can speak about the zeros and poles of f and assigning the proper multiplicities to these zeros and poles we can define

$$\text{div } f = \sum_{\mathsf{P}\epsilon\mathsf{E}} n_{\mathsf{P}}\,[\mathsf{P}],$$

the divisor of f on E. The coefficients n_{P} are integers and only finitely many of them are nonzero so the sum is finite. If P is a zero of multiplicity n then $n_{\mathsf{P}} = n$, if P is a pole of multiplicity n then $n_{\mathsf{P}} = -n$, and if P is neither then $n_{\mathsf{P}} = 0$.

Definition

The *degree* of a divisor is the sum of its coefficients

$$\text{deg } \mathsf{D} = \text{deg } (\sum_{\mathsf{P}\epsilon\mathsf{E}} n_{\mathsf{P}}\,[\mathsf{P}]) = \sum_{\mathsf{P}\epsilon\mathsf{E}} n_{\mathsf{P}}.$$

Definition

The *sum* of a divisor is simply

$$\text{sum } \mathsf{D} = \text{sum } (\sum_{\mathsf{P}\epsilon\mathsf{E}} n_{\mathsf{P}}\,[\mathsf{P}]) = \sum_{\mathsf{P}\epsilon\mathsf{E}} n_{\mathsf{P}}\mathsf{P}$$

Theorem 16. *Let* E *be an elliptic curve.*

(a) Let f *and* g *be rational functions on* E*. If div* f = *div* g*, then there exists a nonzero constant* c *such that* $f = cg$*.*

(b) Let $\mathsf{D} = \sum_{\mathsf{P}\epsilon\mathsf{E}} n_{\mathsf{P}}[\mathsf{P}]$ *be a divisor on* E *. Then* D *is the divisor of a rational function if and only if*

deg $\mathsf{D} = 0$ *and sum* $\mathsf{D} = \mathcal{O}$*. In particular, if a rational function on* E *has no zeros or no poles, then it is constant.*

Proof. For a proof of this see Silverman, 1992, (Prop. II 3.1 and III 3.4) □

4.4 The Weil pairing

Let P and $\mathsf{Q}\ \epsilon\ \mathsf{E}[m]$ and let f_{P} and f_{Q} be rational functions on E satisfying

$$\text{div } f_{\mathsf{P}} = m\,[\mathsf{P}] - m[\mathcal{O}] \text{ and div } f_{\mathsf{Q}} = m\,[\mathsf{Q}] - m[\mathcal{O}].$$

The *Weil pairing* of P and Q is the quantity

$$e_m(\mathsf{P},\mathsf{Q})\ \frac{f_{\mathsf{P}}(\mathsf{Q}+\mathsf{S})}{f_{\mathsf{P}}(\mathsf{S})} \Big/ \frac{f_{\mathsf{Q}}(\mathsf{P}-\mathsf{S})}{f_{\mathsf{Q}}(-\mathsf{S})}\ ,$$

where $\mathsf{S}\ \epsilon\ \mathsf{E}$ is any point satisfying $\mathsf{S} \notin \{\mathcal{O}, \mathsf{P}, -\mathsf{Q}, \mathsf{P}-\mathsf{Q}\}$.

Claim.

e_m is well defined.

Proof. Let $f'_{\mathsf{P}}$and f'_{Q} be a different choice of rational functions such that

$$\text{div } f'_{\mathsf{P}} = m\,[\mathsf{P}] - m[\mathcal{O}] \text{ and div } f'_{\mathsf{Q}} = m\,[\mathsf{Q}] - m[\mathcal{O}].$$

It is immediate that

$$\text{div } f'_{\mathsf{P}} = \text{div } f_{\mathsf{P}} \text{ and div } f'_{\mathsf{Q}} = \text{div } f_{\mathsf{Q}}.$$

Consequently, by Theorem 7, $f'_{\mathsf{P}} = a f_{\mathsf{P}}$ and $f'_{\mathsf{Q}} = b f_{\mathsf{Q}}$. Hence we have that :

$$e'_m(\mathsf{P},\mathsf{Q}) = \frac{f'_{\mathsf{P}}(\mathsf{Q}+\mathsf{S})}{f'_{\mathsf{P}}(\mathsf{S})} \Big/ \frac{f'_{\mathsf{Q}}(\mathsf{P}-\mathsf{S})}{f'_{\mathsf{Q}}(-\mathsf{S})} = \frac{a f_{\mathsf{P}}(\mathsf{Q}+\mathsf{S})}{a f_{\mathsf{P}}(\mathsf{S})} \Big/ \frac{b f_{\mathsf{Q}}(\mathsf{P}-\mathsf{S})}{b f_{\mathsf{Q}}(-\mathsf{S})} = \frac{f_{\mathsf{P}}(\mathsf{Q}+\mathsf{S})}{f_{\mathsf{P}}(\mathsf{S})} \Big/ \frac{f_{\mathsf{Q}}(\mathsf{P}-\mathsf{S})}{f_{\mathsf{Q}}(-\mathsf{S})} = e_m(\mathsf{P},\mathsf{Q})$$

$\because e_m$ is independent of the choice of rational functions. □

We would not want e_m to depend on the choice of point $\mathsf{S}\ \epsilon\ \mathsf{E}$. Fix some points P and Q, call them $\mathcal{P}$ and $\mathcal{Q}$. Consider the function

$$\mathsf{F}(\mathsf{S}) = \frac{f_{\mathcal{P}}(\mathcal{Q}+\mathsf{S})}{f_{\mathcal{P}}(\mathsf{S})} / \frac{f_{\mathcal{Q}}(\mathcal{P}-\mathsf{S})}{f_{\mathcal{Q}}(-\mathsf{S})} \ .$$

Let us concentrate on $\frac{f_{\mathcal{P}}(\mathcal{Q}+\mathsf{S})}{f_{\mathcal{P}}(\mathsf{S})}$.

It is quite obvious that $f_{\mathcal{P}}(\mathcal{Q}+\mathsf{S}) = (\ f_{\mathcal{P}} \circ \tau_{\mathcal{Q}})(\mathsf{S})$, where $\tau_{\mathcal{Q}}$: $\mathsf{E} \longrightarrow \mathsf{E}$ is the (bijective !) translation given by $\mathsf{S} \longmapsto \mathsf{S} + \mathcal{Q}$.

Trying to determine div $(\ f_{\mathcal{P}} \circ \tau_{\mathcal{Q}})$ we can consider $(\ f_{\mathcal{P}} \circ \tau_{\mathcal{Q}})(\mathsf{S})$ and determine n_{S} or consider $f_{\mathcal{P}}(\tau_{\mathcal{Q}}(\mathsf{S}))$ and determine $n_{\tau_{\mathcal{Q}}(\mathsf{S})}$.

The following must hold:

$$\text{div } (\ f_{\mathcal{P}} \circ \tau_{\mathcal{Q}}) = \sum_{\mathsf{S} \epsilon \mathsf{E}} n_{\mathsf{S}} (\ f_{\mathcal{P}} \circ \tau_{\mathcal{Q}}) \ [\mathsf{S}]$$

$$\sum_{\mathsf{S} \epsilon \mathsf{E}} n_{\mathsf{S}} (\ f_{\mathcal{P}}) \ [\tau_{\mathcal{Q}}(\mathsf{S})] = \sum_{\tau_{\mathcal{Q}}(\mathsf{S}) \epsilon \mathsf{E}} n_{\tau_{\mathcal{Q}}(\mathsf{S})} (\ f_{\mathcal{P}}) \ [\tau_{\mathcal{Q}}(\mathsf{S})] = \text{div } (f_{\mathcal{P}}).$$

$\because f_{\mathcal{P}} \circ \tau_{\mathcal{Q}}) = c f_{\mathcal{P}}$.

Hence:

$$\frac{f_{\mathcal{P}}(\mathcal{Q}+\mathsf{S})}{f_{\mathcal{P}}(\mathsf{S})} = \frac{(f_{\mathcal{P}} \circ \tau_{\mathcal{Q}})(\mathsf{S})}{f_{\mathcal{P}}(\mathsf{S})} = \frac{c f_{\mathcal{P}}(\mathsf{S})}{f_{\mathcal{P}}(\mathsf{S})} = c \text{ , a constant.}$$

Analogously we can show that $\frac{f_{\mathcal{Q}}(\mathcal{P}-\mathsf{S})}{f_{\mathcal{Q}}(-\mathsf{S})}$ is constant.

$\because \mathsf{F}(\mathsf{S})$ is a constant function and thus e_m does not depend on the point S (as long as $\mathsf{S} \notin \{\mathcal{O}, \mathsf{P}, -\mathsf{Q}, \mathsf{P}-\mathsf{Q}\}$).

Theorem 17. *(a) The Weil pairing is bilinear, which means that*

$$e_m(\mathsf{P}_1 + \mathsf{P}_2, \mathsf{Q}) = e_m(\mathsf{P}_1, \mathsf{Q}) \cdot e_m(\mathsf{P}_2, \mathsf{Q})$$

$$e_m(\mathsf{P}, \mathsf{Q}_1 + \mathsf{Q}_2) = e_m(\mathsf{P}, \mathsf{Q}_1) \cdot e_m(\mathsf{P}, \mathsf{Q}_2)$$

for all $\mathsf{P}, \mathsf{P}_1, \mathsf{P}_2, \mathsf{Q}, \mathsf{Q}_1, \mathsf{Q}_2 \ \epsilon \ \mathsf{E}[m]$.

A consequence of this is that $e_m(2\mathsf{P}, \mathsf{Q}) = e_m(\mathsf{P}+\mathsf{P}, \mathsf{Q}) = e_m(\mathsf{P}, \mathsf{Q}) \cdot e_m(\mathsf{P}, \mathsf{Q}) = e_m(\mathsf{P}, \mathsf{Q})^2$ *, and then, more generally, and by induction,* $e_m(r\mathsf{P}, s\mathsf{Q}) = e_m(\mathsf{P}, \mathsf{Q})^{rs}$.

(b) The Weil pairing is alternating, which means that

$$e_m(\mathsf{P}, \mathsf{P}) = 1$$

A consequence of this and of bilnearity is that

$$1 = e_m(\mathsf{P} + \mathsf{Q}, \mathsf{P} + \mathsf{Q}) =$$

$$e_m(\mathsf{P}, \mathsf{P}) \cdot e_m(\mathsf{P}, \mathsf{Q}) \cdot e_m(Q, \mathsf{P}) \cdot e_m(\mathsf{Q}, \mathsf{Q}) =$$

$$1 \cdot e_m(P, Q) \cdot e_m(Q, P) \cdot 1$$

$\because e_m(Q, P) = e_m(P, Q)^{-1}$.

Another consequence. Consider the representations $P = a_P P_1 + b_P P_2$ *and* $Q = a_Q P_1 + b_Q P_2$*, given the basis* $\{P_1, P_2\}$ *of* $E[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$.

$$e_m(P, Q) = e_m(a_P P_1 + b_P P_2\ , a_Q P_1 + b_Q P_2) =$$

$$e_m(a_P P_1,\ a_Q P_1) e_m(a_P P_1,\ b_Q P_2) e_m(b_P P_2,\ a_Q P_1) e_m(b_P P_2,\ b_Q P_2) =$$

$$e_m(P_1,\ P_1)^{a_P a_Q} \cdot e_m(P_1,\ P_2)^{a_P b_Q} \cdot e_m(P_2,\ P_1)^{b_P a_Q} \cdot e_m(P_2,\ P_2)^{a_P b_Q} =$$

$$1^{a_P a_Q} \cdot e_m(P_1,\ P_2)^{a_P b_Q} \cdot e_m(P_1,\ P_2)^{-b_P a_Q} \cdot 1^{a_P b_Q} =$$

$$e_m(P_1,\ P_2)^{a_P b_Q - a_Q b_P} =$$

$$e_m(P_1,\ P_2)^{det \begin{bmatrix} a_P & a_Q \\ b_P & b_Q \end{bmatrix}}.$$

(c) The Weil pairing is nondegenerate, which means that

$$e_m(P, Q) = 1$$

for all points Q *if and only if* $P = \mathcal{O}$. *A consequence of this and of bilinearity is that*

$$e_m(P, Q)^m = e_m(mP, Q) = e_m(\mathcal{O}, Q) = 1.$$

Finally we conclude that $e_m(P, Q)$ *is an* m^{th} *root of unity.*

Proof. For a proof see Silverman, 1992. □

In cryptography we shall want to evaluate the Weil pairing at
$P_1 = aP$ and $P_2 = bP$, so $e_m(P_1, P_2) = e_m(aP, bP) = e_m(P, P)^{ab} = 1$, and this is not very helpful.

The solution is to choose an elliptic curve that has some *nice* map

$$\varphi\colon E \longrightarrow E$$

attached to it, such that P and $\varphi(P)$ are linearly independent in $E[m]$, and then we can compute

$$e_m(P_1, \varphi(P_2) = e_m(aP, \varphi(bP)) =$$

$$e_m(aP, b\varphi(P)) = e_m(P, \varphi(P)^{ab}.$$

4.5 Distortion maps

Let $l \geq 3$ be a prime and let E be an elliptic curve with $\mathsf{P} \in \mathsf{E}[l]$ a point of order l. Let φ: $\mathsf{E} \longrightarrow \mathsf{E}$ be a map. We say that φ is an l – distortion map for P if it has the following properties:

(i) $\varphi(n\mathsf{P}) = n\varphi(P)$ for all $n \geq 1$.

(ii) The number $e_l(\mathsf{P}, \varphi(\mathsf{P}))$ is a primitive lth root of unity, meaning that $e_l(\mathsf{P}, \varphi(\mathsf{P}))^r = 1$ if and only if $r = kl$.

Theorem 18. *Let E be an elliptic curve, let $l \geq 3$ be a prime, and view $\mathsf{E}[l] \cong \mathbb{Z}_l \times \mathbb{Z}_l$ as a 2–dimensional vector space over the field $\mathbb{Z}_l$. Let $\mathsf{P}, \mathsf{Q} \in \mathsf{E}[l]$. Then the following are equivalent:*

(a) P and Q form a basis for the vector space $\mathsf{E}[l]$.

(b) $\mathsf{P} \neq \mathcal{O}$ and Q is not a multiple of P.

(c) $e_l(\mathsf{P}, \mathsf{Q})$ is a primitive lth root of unity.

(d) $e_l(\mathsf{P}, \mathsf{Q}) \neq 1$.

Proof. (a) $\Rightarrow$(b) is obvious.

Suppose (b) holds and assume (a) is false, so $u\mathsf{P} + v\mathsf{Q} = \mathcal{O}$, where u and $v \in \mathbb{Z}_l$ are not both zero.

If $v = 0$ then $\mathsf{P} = \mathcal{O}$ so (b) turns out to be false too. Contradiction!

If $v \neq 0$ then v has an inverse in $\mathbb{Z}_l$, so $\mathsf{Q} = -v^{-1}u\mathsf{P}$ and this shows Q to be a multiple of P which, again, contradicts (b).

Thus (a) must hold.

$\because$(b) $\Rightarrow$(a)

$\because$(a) $\Leftrightarrow$(b)

Set $\zeta = e_l(\mathsf{P}, \mathsf{Q})$. We know that $\zeta^l = 1$. Let $r \geq 1$ be the order of ζ, i. e. $\zeta^r = 1$.

Using the Euclidean algorithm we have $sr+tl = \gcd(r, l)$, $s, t \in \mathbb{Z}$. Hence:

$\zeta^{\gcd(r,l)} = \zeta^{sr+tl} = (\zeta^r)^s\left(\zeta^l\right)^t = 1$. Consequently $r = \gcd(r, l)$ by the minimality of r. This implies that $r \mid l$. But l is prime so either $r = 1$ and so $\zeta = 1$ or $r = l$. This argument shows that

(c) $\Leftrightarrow$(d)

Suppose (a) holds. Given that $\{\mathsf{P}, \mathsf{Q}\}$ is a basis for $\mathsf{E}[l]$ we then know that $\mathsf{P} \neq \mathcal{O}$.

By the nondegeneracy of the Weil pairing there must exist a point R in $\mathsf{E}[l]$ such that $e_l(\mathsf{P}, \mathsf{R}) \neq 1$. We can write $\mathsf{R} = u\mathsf{P} + v\mathsf{Q}$, $u, v \in \mathbb{Z}_l$. By the bilinearity and alternating properties of the Weil pairing we get:

$$1 \neq e_l(\mathsf{P},\mathsf{R}) = e_l(\mathsf{P},u\mathsf{P}+v\mathsf{Q}) = e_l(\mathsf{P},\mathsf{P})^u \cdot e_l(\mathsf{P},\mathsf{Q})^v 1^u \cdot e_l(\mathsf{P},\mathsf{Q})^v = e_l(\mathsf{P},\mathsf{Q})^v.$$

$\because$ (d) is true
$\because$ (a)$\Rightarrow$(d)

Suppose (d) holds and assume that (b) is false. The latter assumption means that either $\mathsf{P} = \mathcal{O}$ or $\mathsf{Q} = u\mathsf{P}$, for some $u \in \mathbb{Z}_l$.

If $\mathsf{P} = \mathcal{O}$ then $e_l(\mathsf{P},\mathsf{Q}) = e_l(\mathcal{O},\mathsf{Q}) = 1$. If $\mathsf{Q} = u\mathsf{P}$ then $e_l(\mathsf{P},\mathsf{Q}) = e_l(\mathsf{P},u\mathsf{P}) = e_l(\mathsf{P},\mathsf{P})^u = 1$.

Thus (d) is false. Contradiction!
$\because$(d) $\Rightarrow$(b)
We have shown that (a) $\Rightarrow$(d) $\Rightarrow$(b) $\Rightarrow$ (a)
$\because$ (a) $\Leftrightarrow$(d)
$\because$(a) $\Leftrightarrow$(b) $\Leftrightarrow$ (c) $\Leftrightarrow$(d)

□

A modified Weil pairing

Let E be an elliptic curve, let $\mathsf{P} \in \mathsf{E}[l]$, and let φ be an l – distortion map for P. The *modified Weil pairing* $\hat{e}_l$ on $\mathsf{E}[l]$ (relative to φ) is defined by

$$\hat{e}_l(\mathsf{P},\mathsf{P}') = e_l(\mathsf{P},\varphi(\mathsf{P}')).$$

This is important because in cryptographic applications we must evaluate the modified Weil pairing at multiples of P.

Theorem 19. *Let E be an elliptic curve, let $\mathsf{P} \in \mathsf{E}[l]$, let φ be an l – distortion map for P and let $\hat{e}_l$ be the modified Weil pairing on $\mathsf{E}[l]$ (relative to φ). Let Q and Q' be multiples of P. Then*

$$\hat{e}_l(\mathsf{Q},\mathsf{Q}') = 1 \text{ if and only if } \mathsf{Q} = \mathcal{O} \text{ or } \mathsf{Q}' = \mathcal{O}.$$

Proof. Let $\mathsf{Q} = s\mathsf{P}$ and $\mathsf{Q}' = t\mathsf{P}$.

$$\hat{e}_l(\mathsf{Q},\mathsf{Q}') = \hat{e}_l(s\mathsf{P},t\mathsf{P}) = e_l(s\mathsf{P},\varphi(t\mathsf{P})) = e_l(s\mathsf{P},t\varphi(\mathsf{P})) = e_l(\mathsf{P},\varphi(\mathsf{P}))^{st}\ .$$

$e_l(\mathsf{P},\varphi(\mathsf{P}))$ is a primitive lth root of unity, so

$$\hat{e}_l(\mathsf{Q},\mathsf{Q}') = 1 \iff l \mid st \iff l \mid s \text{ or } l \mid t \iff \mathsf{Q} = \mathcal{O} \text{ or } \mathsf{Q}' = \mathcal{O}\ .$$

□

Let us consider the elliptic curve E: $y^2 = x^3 + 1$ over the field $\mathbb{F}_{691}$. We shall try to define a distortion map by

$$\varphi(x, y) = (\omega x, y)$$

with ω a primitive cube root of unity, i.e. $\omega^3 = 1$. Note that this means that $\omega^2 = \omega^{-1}$. Furthermore we set $\varphi(\mathcal{O}) = \mathcal{O}$.

Let $\mathsf{P} = (x, y)\ \epsilon\ \mathsf{E}$. Obviously, $\varphi(\mathsf{P}) = (\omega x, y)$.

$$(\omega x)^3 + \mathsf{1} = \omega^3 x^3 + \mathsf{1} = x^3 + \mathsf{1} = y^2,$$

and this shows that $\varphi(\mathsf{P})\ \epsilon\ \mathsf{E}$.

We shall denote the $x-$coordinate of P by $X(\mathsf{P})$ and the $y-$coordinate of P by $Y(\mathsf{P})$.

Assume P and Q are two distinct points on E. Then we have:

$$X(\varphi(\mathsf{P}) + \varphi\ (\mathsf{Q})) =$$

$$X((\omega x_1, y_1) + (\omega x_2, y_2)) =$$

$$k^2 - \omega x_1 - \omega x_2 = (\tfrac{y_2 - y_1}{\omega x_2 - \omega x_1})^2 - \omega x_1 - \omega x_2 =$$

$$\tfrac{1}{\omega^2}\ (\tfrac{y_2 - y_1}{\omega x_2 - \omega x_1})^2 - \omega x_1 - \omega x_2 = \omega\ (\tfrac{y_2 - y_1}{\omega x_2 - \omega x_1})^2 - \omega x_1 - \omega x_2 =$$

$$\omega((\tfrac{y_2 - y_1}{\omega x_2 - \omega x_1})^2 - x_1 - x_2) = \omega X(\mathsf{P} + \mathsf{Q})$$

$$\because X(\varphi(\mathsf{P} + \mathsf{Q})) = X(\varphi(\mathsf{P}) + \varphi\ (\mathsf{Q}))$$

and

$$Y(\varphi(\mathsf{P}) + \varphi\ (\mathsf{Q})) =$$

$$Y((\omega x_1, y_1) + (\omega x_2, y_2)) =$$

$$k(\omega x_1 - \omega X(\mathsf{P} + \mathsf{Q}))\ - y_1 =$$

$$\tfrac{y_2 - y_1}{\omega x_2 - \omega x_1}(\omega x_1 - \omega X(\mathsf{P} + \mathsf{Q}))\ - y_1 =$$

$$\tfrac{1}{\omega}\omega\tfrac{y_2 - y_1}{x_2 - x_1}(x_1 - X(\mathsf{P} + \mathsf{Q}))\ - y_1 =$$

$$Y(\mathsf{P} + \mathsf{Q})$$

$$\because Y(\varphi(\mathsf{P} + \mathsf{Q})) = Y(\mathsf{P} + \mathsf{Q})$$

We conclude that

$$\varphi(\mathsf{P}) + \varphi\ (\mathsf{Q}) = \varphi(\mathsf{P} + \mathsf{Q})$$

Assume $\mathsf{P} = \mathsf{Q}$ on E. Then we have:

$$X(2\varphi(\mathsf{P})) = X(2(\omega x, y)) =$$

$$k^2 - 2\omega x = (\tfrac{3(\omega x)^2}{2y}) - 2\omega x =$$

$$\omega^2(\tfrac{3x^2}{2y})^2 - 2\omega x = \omega(k^2 - 2x) =$$

$$\omega X(2\mathsf{P}) = X(\varphi(2\mathsf{P}))$$

and

$$Y(2\varphi(\mathsf{P})) = Y(2(\omega x, y)) =$$

$$k(\omega x - k^2 + 2\omega x) - y =$$

$$Y(\varphi(2\mathsf{P}))$$

We conclude that

$$\varphi(2\mathsf{P}) = 2\varphi(\mathsf{P})$$

and this can be generalized, by induction, to

$$\varphi(\mathtt{n}\mathsf{P}) = \mathtt{n}\varphi(\mathsf{P}).$$

Now, with an appropriate point $\mathsf{P} \in \mathsf{E}(\mathbb{F}_{691})$ and by means of the Miller algorithm we might show that $\hat{e}_l(\mathsf{P}, \mathsf{P}) = e_l(\mathsf{P}, \varphi(\mathsf{P}))^r = \mathtt{1}$ if and only if $r = kl$, where $l = \text{ord } \mathsf{P}$ in $\mathsf{E}(\mathbb{F}_{691})$. This would prove φ to be an $l-$distortion map for P. For all this I refer to Hoffstein, Pipher & Silverman, section 5.8.4.

4.6 Hyperelliptic curves

A hyperelliptic curve of genus g is the set of solutions to an equation of the form

$$\mathsf{H} : y^2 = x^{2g+1} + a_1 x^{2g} + \ldots\ldots\ldots + a_{2g} x + a_{2g+1},$$

where $f(x) = x^{2g+1} + a_1 x^{2g} + \ldots\ldots\ldots + a_{2g} x + a_{2g+1}$ has distinct roots.

To this set we add a point $\mathcal{O}$ at infinity. Obviously an elliptic curve has genus $g = 1$, according to this definition.

In general we cannot define addition of points on H as we could in the case of an elliptic curve E. But we shall define a *divisor* on H to be a formal sum of points

$$n_1[\mathsf{P}_1] + n_2[\mathsf{P}_2] + \ldots\ldots\ldots + n_r[\mathsf{P}_r]$$

with $\mathsf{P}_i \in \mathsf{H}$, $n_i \in \mathbb{Z}$ for $i = 1, 2, \ldots\ldots, r$.

Clearly a divisor is a finite sum of points each with its own multiplicity. If $f(x, y)$ is a rational function on H then we define div $f = \mathsf{D}$ by listing the zeros and poles of f on H with their respective multiplicities. The *degree* of a divisor is defined as

$$\deg \mathsf{D} = \deg\,(\, n_1[\mathsf{P}_1] + n_2[\mathsf{P}_2] + \ldots\ldots\ldots + n_r[\mathsf{P}_r]) = n_1 + n_2 + \ldots\ldots\ldots + n_r.$$

We define then DivH to be the set of divisors on H.

We can clearly add and subtract divisors by adding and subtracting the multiplicities at each point. Further, we denote by $\mathsf{Div_0H}$ the set of divisors of degree zero.

Two divisors D_1 and D_2 are said to be linearly equivalent if $\mathsf{D}_1 - \mathsf{D}_2 =$ divisor of a function. The divisor of a function always has degree zero. We define $\mathsf{Jac_oH}$ to be the set of divisors of degree zero where we identify linearly equivalent divisors, i.e. $\mathsf{Jac_oH} \cong \mathsf{Div_0H}/\sim$.

$\mathsf{Jac_oH}$ together with the addition law obtained by adding the multiplicities of points is called the *Jacobian variety* of H. Thus $\mathsf{Jac_oH}$ can be described as the set of solutions to a system of polynomial equations and the addition law may be described using polynomials as well. If we take solutions with coordinates in $\mathbb{F}_p$ we obtain a group analogous to $\mathsf{E}(\mathbb{F}_p)$.

Setting $\mathsf{J} := \mathsf{Jac_oH}$ and writing $\mathsf{J}(\mathbb{F}_p)$ for the points in $\mathsf{Jac_oH}$ with coefficients in $\mathbb{F}_p$ we can formulate the discrete logarithm problem as follows: given points P and Q in $\mathsf{J}(\mathbb{F}_p)$ find an integer n such that $\mathsf{Q} = n\mathsf{P}$. Mimicking then the elliptic case we can construct cryptosystems based on the

Hyperelliptic Curve Discrete Logarithm Problem (HCDLP), i.e. a hyperelliptic Diffie – Hellman key exchange and a hyperelliptic ElGamal public key cryptosystem.

Why would we want to do that? Because there are more points on $\mathsf{J}(\mathbb{F}_p)$ than there are on $\mathsf{E}(\mathbb{F}_p)$ for one thing.

An analogue of Hasse's theorem due to Weil says that $\#\mathsf{J}(\mathbb{F}_p) = p^g + \mathcal{O}(p^{g-\frac{1}{2}})$, so a hyperelliptic curve of genus 2 can offer approximately p^2 points in $\mathsf{J}(\mathbb{F}_p)$.

Cryptanalysis could now be conducted with collision algorithms such as Pollard's ρ algorithm yet this is not the best known method. An index calculus algorithm seems to be better according to Hoffstein, Pipher & Silverman (8.10)

Since $\#\mathsf{J}(\mathbb{F}_p) \approx p^g$ this means, according to the same trio, that solving the HCDLP would require $\mathcal{O}(p^{\frac{g}{2}})$ steps so using curves with $g > 1$ would achieve levels of security equivalent to those offered by elliptic curves but with a lesser p. There are of course both advantages and disadvantages with regard to computations and security just as in the elliptic case.

I shall not pursue the matter further. I simply wanted to mention the hyperelliptic curves and the existence of a hyperelliptic cryptography.

Bibliography

[1] T. Ekedahl 2000. *One Semester of Elliptic Curves*. European Mathematical Society.

[2] T. Ekedahl 2003. *Elementär algebraisk geometri*. Matematiska Institutionen. Stockholms Universitet.

[3] J. Hoffstein, J. Pipher & J. H. Silverman 1983. *An Introduction to Mathematical Cryptography*. Springer Verlag.

[4] D. Husemöller 2002. *Elliptic Curves*. Springer Verlag.

[5] N. Koblitz 1994. *A Course in Number Theory and Cryptography*. Springer Verlag.

[6] J. H. Silverman 1992 *The Arithmetic of Elliptic Curves*. Springer Verlag.