Security risk associated with multiple users sharing the same elliptic curve

Sergi Blanch-Torné[1], Ramiro Moreno Chiral[2], Francesc Sebé Feixa[2]

[1] Escola Politècnica Superior, Universitat de Lleida. Spain.
sblanch@alumnes.udl.es
[2] Departament de Matemàtica. Universitat de Lleida. Spain.
{ramiro,fsebe}@matematica.udl.es

August 11, 2013
github.Papers: 2013-08-11 (revision eb72e90)

Abstract. [3] FIXME: "*this probably is too long*" One of the main good features of the elliptic curves is the possibility to change the group where the cryptography is based, without a penalty by increase the length of this group. Using a different elliptic curve, the cyclic subgroup is completely different and the cryptanalysis that someone can have in progress, over this curve is useless over another curve over the same finite field size. The complexity to port an attack from one curve to another can be even bigger than restart the attack over the second one from scratch.
Also as is said in section 5.1 of [1], the security in elliptic curves doesn't rely on the secrecy of the domain parameters, the risk is when multiple users shares the same elliptic curve parameters. The current standardization [2] flow goes in the way to have one curve per length and three possible lengths. That means more than multiple users with the same domain parameters, almost all the users will share the same domain parameters.
It is a request for this standard to find a way to add variability between the used elliptic curves, and in this paper two ways are be explored. Both ways likes to get an *auditable* algorithm to the final user to get its particular elliptic curve. The difference between this two approaches is if the algorithm starts from scratch or starts from one cryptographically good curve.

Keywords: Cryptography, Elliptic Curves, Isogeny

1 Introduction

The introduction of the elliptic curves in the standards are following its path, and it is having a good health. Even that the [3] does not include yet the elliptic curve cryptosystem in the same level than the finite fields, the ECDSA is

[3] Partially supported by grants MTM2010-21580-C02-01 (Spanish Ministerio de Ciencia e Innovación), 2009SGR-442 (Generalitat de Catalunya).

already standardized ([4] and [5]) and many implementations are available, the encryption has been released in June 2012 after a long process [2] and at least there is one implementation available in the *GnuPG* since 2011.

But this standard is restricting the number of curves to three (NIST curves from [4]: P-256, P-384, P-521). In the best case, this standard allows the usage of curves with an assigned OID from the IANA, for example the german ecc-brainpool standardized [6] or the certicom's ones [7]. Even if the list can be very extended with this OIDs, there are over hundreds of thousands good elliptic curves excluded with this method.

This OID method can be an option for keys where the bandwidth is limited, but there must be the possibility to use any good curve, because is one of the best advantages of the elliptic curve cryptography.

In the abstract was mention the X9.62-1998[1] , where in section 5.1 is mention that is a security thread to share the same curve between many users. In annex H, more specifically in H.2 (point 5) and H.1.3, is explained (but not proof or reference a proof, only mention the *Pollard-λ* attack) that the time to break one key pair (when they have the same curve and the cyclic group generator, the case of the standard rfc6637 [2]) is reduced by $\sqrt{k}$, where is the number of solving instances:

#Compromised keys	Time to break next
2	$(\sqrt{2}-1)t \approx 0.41t$
3	$(\sqrt{3}-\sqrt{2})t \approx 0.32t$
4	$(\sqrt{4}-\sqrt{3})t \approx 0.27t$
$\vdots$	$\vdots$
k	$(\sqrt{k}-\sqrt{k-1})t$

Back to the rfc6637 [2] and apart from this issue of the curve limitation, the refereed standard for elliptic curve encryption have many other good aspects. This standard propose an schema that is similar to the ones proposed in [8] with the collaboration of Mikael Mylnikov, developed independently to the standard proposed, but we arrive to the same conclusions. The schema of ElGamal cannot be translated from finite fields to elliptic curves without modification.

Before to enter in a detailed proposal of how to make cryptographically good elliptic curves (sections 2 and 3 of this paper) some concepts must be introduced.

First of all, and before to enter in elliptic curve concepts, there is a more general definitions about algebra that are presupposed to the reader about fields $\mathbb{K}$, or algebraic rings $(R,+,\cdot)$ or $\mathbb{Z}/[\mathbb{Z}\mathrm{n}]$, or more particularly finite fields $\mathbb{F}_q$, that is the only one that will be specifically introduced:

Definition 1. *A finite field of q order, $\mathbb{F}_q$ factorizes necessarily like $q = p^r$ where p is a characteristic prime of the field and $r \in \mathbb{Z}_{>0}$ the extension degree.*

In case of a small p, the extension is big like happens with the primes 2 or 3. This way vectorial spaces are defined like the finite fields of characteristic 2 denoted as $\mathbb{F}_{2^m}$. In another case, when p is big, the extension is small (usually $r = 1$ and prime finite fields are defined denoted as $\mathbb{F}_p$.

1.1 What is an elliptic curve?

Definition 2. *An Elliptic curve is* nonsingular *curve of genus* 1 *over a field* $\mathbb{K}$*, with at least one* K*-rational point denoted* $E/\mathbb{K}$*, given by the Weierstraß Normal Form (WNF) equation (in an Affine plane* $\mathcal{A}_2$*):*

$$E/\mathbb{K}: \ y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6, \ a_i \in \mathbb{K} \tag{1}$$

When it is defined over a prime finite field, it is call the Weierstraß Reduced Form (WRF) equation that can be written in affine notation as:

$$y^2 = x^3 + ax + b \tag{2}$$

An elliptic curve can be defined over any field, like real numbers $\mathbb{R}$ where representations over coordinate axis are more human-readable, but in cryptology usually are defined over finite fields.

TODO: "*Place a figure with the same curve represented over the reals and over a very small finite field*"

A singular point is when both partial derivatives vanish, in $\mathbb{R}$ representation means that the curve never has beaks neither nodes. Analytically can be differentiate by the *discriminant* (represented with Δ) that must *not* be 0 in the *nonsingular* curves. The singularities, under a geometrical representation of a curve over the real numbers $\mathbb{R}$, they are drawn as intersections, isolated points and cusp.

Definition 3. *The discriminant, denoted with* Δ*, is a polynomial function of the coefficients of the equation which gives information about the nature of the equations roots.*

FIXME: "*Improve this discriminant definition*"

About the *discriminant* of an elliptic curve defined over the prime finite field, when this curve has the representation of the WRF 2, the coefficients are a and b as $\Delta = -16(4a^3 + 27b^2) \neq 0$.

Another important parameter of an elliptic curve is the cardinality, who can be defined as:

Definition 4. *The* cardinality *of an elliptic curves* E *over* $\mathbb{F}_q$*is defined as the number of* $\mathbb{F}_q$*-rational points. Where a* $\mathbb{F}_q$*-rational points is a pair* $(x, y) \in \mathbb{F}_q^2$ *such that follows the elliptic curve equation.*

FIXME: "*this definition here is a bit forced, but where is this better?*"

A curve over a general finite field $\mathbb{F}$ is composed by the points of the curve with the notation $E(\mathbb{F})$. The points of an elliptic curve over a prime finite field are all ones that resolves the equation 2 union the point at infinity (the neutral element of the group, denoted $\mathcal{O}$) who does not have representation over the affine plane and other coordinate representations are better like the projectives $\mathcal{P}_2$ or Jacobians $\mathcal{J}_2$.

$$E(\mathbb{F}) = \left\{(x, y) \in \mathbb{F} \mid y^2 = x^3 + ax + b\right\} \bigcup \mathcal{O}_E \tag{3}$$

Definition 5. *There are an application between the points in the afine plane $\mathcal{A}_2(\mathbb{F})$ and the projective plane $\mathcal{P}_2(\mathbb{F})$ and this correspondence is reciprocal:*

$$\begin{array}{rl} f : \mathcal{A}_2(\mathbb{F}) \to & \mathcal{P}_2(\mathbb{F}) \\ (x,y) \mapsto & [x:y:1] \end{array} \tag{4}$$

$$\begin{array}{rcl} f^{-1}: \quad \mathcal{P}_2(\mathbb{F}) & \to & \mathcal{A}_2(\mathbb{F}) \\ [X:Y:Z] \mapsto & \left\{ \begin{array}{l} \left(\frac{X}{Z},\frac{Y}{Z}\right) \; \mathit{if}\; Z \neq 0 \\ \quad \neg\exists \quad\; \mathit{if}\; Z = 0 \end{array} \right. \end{array} \tag{5}$$

There exist a equivalence relation between points in the projective plane following:

$$\begin{array}{c} (x,y,z) \sim (x',y',z') \Rightarrow \exists\lambda \in \mathbb{F}^* \\ x = \lambda x', y = \lambda y', z = \lambda z' | x,y,z \in \mathbb{F} \end{array} \tag{6}$$

In the projective plane, three coordinates are used to represent a two-dimensional point, knowing $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$ the Weierstraß reduced normal form 2 will be $\frac{Y^2}{Z^2} = \frac{X^3}{Z^3} + a\frac{X}{Z} + b$, simplified:

$$Y^2Z = X^3 + aXZ^2 + bZ^3 \tag{7}$$

Over a Jacobian plane, also three coordinate are used to represent the points, knowing $x = \frac{X}{Z^2}$ and $y = \frac{Y}{Z^3}$ the Weierstraß reduced normal form 2 is:

$$Y^2Z = X^3 + aXZ^4 + bZ^6 \tag{8}$$

As is described in [9] the projective coordinates improves the operation of the point addition, but the jacobian coordinates gives an advantage over the doubling point operation.

TODO: "*the lost transforming from one representation to the other is worst than use only one? Are they equally good or bad because the distribution of 0's and 1's are similar?*"

An the set of points of an elliptic curve over a prime finite field, like the equation 3 but described in projective coordinates is:

$$E(\mathbb{F}_p) = \left\{ Y^2Z = X^3 + aXZ^2 + bZ^3 \bigcup \mathcal{O}_E \right\} \tag{9}$$

Simplifying the definition 4, the cardinality of an elliptic curve defined over $\mathbb{F}_q$is the number of elements in the set of points of this curve. The cardinality is denoted as $|E(\mathbb{F}_q)|$

1.2 Operation between points of an elliptic curve

Later will be used a operation to define the hard problem that allows us to use the elliptic curves in cryptography. Here there is only it's abstract mathematical definition as operation, that will be later on used.

With the set of points of an elliptic curve defined in 9 a bijective application can be defined between two points and a third one, of the same set, even if they are points of the elliptic curve or the point at infinity $\mathcal{O}_E$.

$$\begin{aligned} + : E\left(\mathbb{F}_p\right) &\rightarrow E\left(\mathbb{F}_p\right) \\ P + Q &\mapsto \quad R \end{aligned} \tag{10}$$

This application, that will be called *addition*, has an algebraic structure of an *Abelian group* because satisfies some requirements: Closure in the group, associativity, identity element, inverse element, and commutativity. In the elliptic curve cryptography, this operation is called *Basic operation*.

TODO: "*As mention before, the projective coordinates have the best algorithm to do this addition operation.*"

Definition 6. *Given this addition operation applied to a point together with itself, this point is* duplicated*; this addition with itself can be done many times and define it as the* scalar product *where the scalar represents the number of times the point is added to itself:*

$$\begin{aligned} * : \quad E\left(\mathbb{F}_p\right) \quad &\rightarrow \quad E\left(\mathbb{F}_p\right) \\ \underbrace{P + \cdots + P}_{x} &\mapsto Q = x * P \end{aligned} \tag{11}$$

The common notation for this operation is $Q = [x]P$ and it is called as the *main operation* in the elliptic curve cryptography because over it is where the hard problem is defined.

TODO: "*Describe the algorithm that loops over the binary representation of the scalar (doing something is it's* 0 *or something else if it's* 1*, that speeds up this operation.*"

TODO: "*This algorithm introduces de* point doubling*, where the projective coordinates are the ones that have the best algorithm (like was projective coordinates for point addition).*"

1.3 Characteristics cryptographically good elliptic curves

As Koblitz proposes [10] an elliptic curve defined over finite fields can be used in cryptology. But to be a cryptographically good elliptic curve when it is defined over a finite field like the ones described in 2, but it needs a bit more to avoid curves susceptible of known attacks. There are some characteristics to be checked.

Definition 7. *Given an elliptic curve with enough point, it must have a cardinality in the* Hasse interval*:*

$$|E(\mathbb{F}_q)| \in (q + 1 - 2\sqrt{q}, q + 1 + 2\sqrt{q}) \tag{12}$$

In the elliptic curve definition 2 it requires that the curve does not have any singularity, but to use an elliptic curve in cryptography the curve not only has to be *non-singular*, the curve can not be *supersingular*.

Definition 8. *An elliptic curve is* not *supersingular when the cardinal is* almost-primer. *And this almost-primarity is when, if the cardinality can be decomposed it must has a prime divisor sufficiently big.*

Following the notation of the Certicom's SEC1 [11] a almost-prime cardinal of an elliptic curve have, at most, two factors:

$$|E(\mathbb{F}_q)| = h \cdot n \tag{13}$$

Where the h is called cofactor and is much more smaller than n ($h \lll n$) and both are prime. Also can be said that the cardinal is *almost* the primer n ($|E(\mathbb{F}_q)| \approx n$). Is in IEEE P1363 [12] where is recommended to not use other cofactors that 1 or 2 or, in the biggest case, 4.

1.4 Cyclic subgroup to define a Discrete Logarithm Problem

From the set of points of a cryptographically good elliptic curve is necessary to distinguish a subset of points who are part of a cyclic group of an sufficiently big order. A generator point must be founded $G \in E(\mathbb{F}_p)$, who is a point capable to produce:

$$\langle G \rangle = \{G, 2G, 3G, \ldots, nG = \mathcal{O}\} \mid \mathrm{ord}(G) = n \tag{14}$$

Corollary 1. *The elements defined for the elliptic curve cryptosystem can be described in a tuple structure. First of all the curve is defined over a primer finite field $\mathbb{F}_p$, the curve can be described with the parameters a and b following the equation 2. In this set of points there is a generator G with order n and a cofactor h. All of them can be resumed in a sextuple:*

$$\{p, a, b, G, n, h\} \tag{15}$$

This set of parameters is what describes de *cryptosystem setup* and the operation computationally hard over we define the cryptography is with in the cyclic subgroup mention before. The *discrete logarithm problem* can be defined over elliptic curves as how to solve the number of times that a point must be added with it self to have the other as a result, following the definition 6:

$$\begin{array}{rrl} \mathit{ECDLP}: & \langle G \rangle & \to \mathbb{Z} \\ & Q = [x]P & \mapsto x \end{array} \tag{16}$$

As mention at the beginning, the standard X9.62-1998 [1], recommends to avoid many different key pair coexisting in the use of the same cryptosystem setup. This setup is has is summarized with the tuple 15. Because of that we have started to propose a way to introduce much more good elliptic curves to distribute and dissolve this risk in between the users of the general propose public key software.

2 Get an elliptic curve from scratch

Generate an elliptic curve is not a hard job, what is hard is to generate an cryptographically good elliptic curve who satisfies the cryptosystem requirements. And also do this in a computationally reasonable time, for all the current possible lengths of the finite field. This procedure must be run in the execution time of the user key generation, and we must assume than this machines will be non very powerful, perhaps an embedded system or an smart card.

Algorithm 1 Generate a cryptographically good elliptic curve

```
Input: Prime $p$ how defines $\mathbb{F}_p$.
Output: Struct with cryptosystem parameters (see 15)
 1: repeat
 2:    Generate $a$ and $b \in_R \mathbb{F}_p$
 3:    Check non zero discriminant $\Delta$
 4:    Calculate the cardinal $|E(\mathbb{F}_p)|$
 5:    Check the cardinal is in between the Hasse interval
 6:    Get the cardinal factors $n, h$ or go to step 1
 7: until $h \in [1, 2, 4]$
 8: repeat
 9:    repeat
10:       Generate $x \in_R \mathbb{F}_p$
11:    until $\exists y \in \mathbb{F}_p$ s.t. $y = \sqrt{x^3 + a * x + b}$
12:    Select randomly $y$ or $-y$
13: until $[n]G = \mathcal{O}_E$
14: Return $\{p, a, b, G, n, h\}$;
```

First approach is the algorithm 1, that consist simply by generate random parameters and check if once has found a good one. Notice that his algorithm output have all the parameters mention in the 15, in fact this algorithm 1 is a *many in one* because the other examples you can get from [12], section A.12.4 (transcribed in algorithm 2) or from X9.62-1998 [1], section A.3.3.2 (transcribed in algorithm 3). Previously to describe those algorithms we have to mention that this algorithm 1 also sets a restriction than cofactor must be 1, 2 or 4, that comes from the P1363 rules ([12]). The cofactor size affects on the group of points of the cyclic subgroup where the cryptography will be.

TODO: “*from where in the P1363 is this cofactor restriction*”

TODO: “*Explain the procedure of the algorithm 2 and its pros and cons*”

TODO: “*This algorithm 3 has many similarities with the 2, but excludes the calculation of n,h and G. Explain the procedure and its pros and cons*”

The steps of the algorithm 1 can be performed in many ways to try to reduce the execution time. The cardinal calculation can be using the *SEA* algorithm and the *Hasse interval* (definition 7) check can be optimized to avoid unnecessary operation, but in a final implementation all this code must be written aware of the side channel attacks. The factorization of the cardinal allows the possibility

Algorithm 2 Constructing a verifiably pseudo-random elliptic curve, P1363 section A.12.4

```
Input: Prime $p$, bounds pair $(n_{min}, n_{max})$ for $n$, $lmax$, division bound $(l_{max} < r_{min})$
Output: A bit string X, struct with cryptosystem parameters (see 15)
 1: Choose an arbitrary bit string $X$ of bit length $L$
 2: Compute $\hbar = H(X)$
 3: Let $W_0$ be the bit string obtained by taking the $w$ rightmost bits of $\hbar$
 4: Convert the length-L bit string $X$ to an integer $z$ (via BS2IP)
 5: for $i = 1$to $s$ do
 6:     Convert the integer $(z + i)$ $(mod\ 2^L)$ to a length-L bit string $X_i$ (via I2BSP)
 7:     Compute $W_i = H(X_i)$
 8: end for
 9: $W = W_0 \| W_1 \| \ldots \| W_s$, bit string obtained by concatenation
10: Convert length-$(v - 1)$ bit string $W$ to an integer $c$ (via BS2IP)
11: if $c = 0$ or $4c + 27 \equiv 0\ (mod\ p)$ then
12:     Go to 1
13: end if
14: Choose $a, b \in \mathbb{F}_p$ s.t. $cb^2 \equiv a^3\ (mod\ p)$
15: Compute order $u$ of $E(\mathbb{F}_p)$
16: Test $u$ for near-primality (via A.15.5)
17: if $u$ not nearly prime then
18:     Go to 1
19: else
20:     calculate $n$,$h$ (the output of A.15.5)
21: end if
22: Generate point $G$ on $E(\mathbb{F}_p)$
23: Return $X$ and $\{p, a, b, G, n, h\}$;
```

Algorithm 3 Constructing a verifiably pseudo-random elliptic curve, X9.62 section A.3.3.2

```
Input: Prime $p$ how defines $\mathbb{F}_p$.
Output: A bit string $SEED$ and field elements $a, b \in \mathbb{F}_p$
    {Let $t = \lfloor log_2 p \rfloor$, $s = \lfloor \frac{t-1}{160} \rfloor$, $\hbar = t - 160 * s$}
 1: Choose arbitrary bit string $SEED$ of length 160 or more. $g = length(SEED)$
 2: Compute $H = SHA-1(SEED)$, and let $c_0$ denote the bit string of length $\hbar$ bits
    obtained by taking the $\hbar$ rightmost bits of $H$
 3: Let $W_0$ denote the bit string of length $h$ bits obtained by setting the leftmost bits
    of $c_0$ to 0 (This ensures that $r < p$)
 4: for i=1 to s do
 5:     Compute $W_i = SHA-1((SEED + i)\ mod(2^g)$
 6: end for
 7: $W = W_0 \| W_1 \| \ldots \| W_s$, bit string obtained by concatenation
 8: Let $w_1, w_2, \ldots, w_t$ be the bits of $W$ from leftmost to rightmost. Let r be the integer
    $r = \sum_{i=1}^{t} w_i 2^{t-1}$
 9: Choose $a, b \in \mathbb{F}_p$s.t. $rb^2 \equiv a^3\ (mod\ p)$
10: if $4a^3 + 27b^2 \equiv 0\ (mod\ p)$ then
11:     Go to 1
12: end if
13: Return $(SEED, a, b)$
```

to only tries to find a little cofactor and check the primary on the n candidate. FIXME: "*with the SEA algorithm, the citation of the article is need*"

TODO: "*Compare the costs of those three algorithms, in time and memory. To do that make them equivalents, by adding to algorithm 3 the calculation of* n *and G. Use the same than algorithm 2, because they look very similar*"

TODO: "*Approach to the number of good curves instead of the number of possible curves in a list. Get it form a huge list (can be OIDs) excludes too much curves, and I cannot be sure from where they come from (extreme paranoia, try to dress this)*" [13]

3 Find a good elliptic curve from another good one

Another method than get a curve from scratch of section 2 is to find a cryptographically good curve based on another curve using a procedure who maintains the good skills and protect the new one from the bad ones as much as possible. The main objective is to avoid the portability of the cryptanalysis from the first curve to the new one, without create any other weakness in the procedure or the new one it self.

TODO: "*cite [14] and [15] about a cryptosystem using isogenies*"

One proposed way to have this requirements are the elliptic curve isogenies. The isogenies have the good skill that they maintain the cardinal from one curve to another, as it will be describe next. And about the main objective to block any type of portability of the cryptanalysis from one curve to another seems to be ok, because this transformation is a hard problem on isogenies.

3.1 What is elliptic curve isogeny?

The concept of *isogeny* describes a particular case of an *isomorphism*, the first thing to be formally defined is the isomorphism.

Definition 9. *Given two elliptic curves $E/\mathbb{K}$ and $E'/\mathbb{K}$ with an equations in the Weierstraß Normal Form described in 1, an isomorphism over the field is when exist $u, r, s, t \in \mathbb{K}$ with $u \neq 0$, such a variable change:*

$$(x, y) \to (u^2x + r, u^3y + u^2sx + t) \tag{17}$$

In case that the curves have the equation in the Weierstraß Reduced Form describer in 2, the variable change is simplified being:

$$(x, y) \to (u^2x, u^3y) \tag{18}$$

The relation of isomorphism is an equivalence relation in a set of the elliptic curves, defined over the same field, where all the curves in the class have the set of $\mathbb{K}$-rational points isomorphic also. Remember from the definition 4, the $\mathbb{K}$-rational points defines the *cardinality* of the elliptic curve, then when two curves are isomorphic, they have the same cardinality.

As has been say at the beginning of this section, the isogenies are a particular case of isomorphisms.

Definition 10. *Given two elliptic curves $E/\mathbb{K}$ and $E'/\mathbb{K}$, they are* isogenies over $\mathbb{K}$ if and only if *exist an isomorphism with coefficients in the field $\mathbb{K}$ (exist a map between all the points of the initial elliptic curve E to the final elliptic curve E') where the neutral element of the elliptic curve $E/\mathbb{K}$ is mapped to the neutral element of the elliptic curve $E'/\mathbb{K}$.*

$$\begin{aligned} \mathcal{I} : E/\mathbb{K} &\to E'/\mathbb{K} \\ (x, y) &\mapsto (X, Y) \\ \mathcal{O}_E &\mapsto \mathcal{O}_{E'} \end{aligned} \tag{19}$$

Where X and Y responds to $X = f_1(x)$ and $Y = f_2(x, y)$. As f_1 like f_2 are functions $\mathbb{K}$-rational.

An isogeny $\mathcal{I}$ is an exhaustive application. Except in the case that the isogeny is like $\mathcal{I}(E) = \{\mathcal{O}\}$, when is said as constant, then exist a unique isogeny: $\widehat{\mathcal{I}} : E' \to E$ called the *dual isogeny.*

FIXME: “*Modular equation (symmetric polynomial of degree $\ell + 1$)?*”

From this definition 10 must be extracted that the cardinality of the elliptic curve is maintained, and they have the same value for both curves. Also is important to remark that the neutral element is constant in the transformation:

$$\mathcal{I}\left(\mathcal{O}_E\right) = \mathcal{O}_{E'} \tag{20}$$

And, finally, remark that for all point $P \in E/\mathbb{K}$ exist a $P' \in E'/\mathbb{K}$. Where can be assumed for the generator $G \in E/\mathbb{K}$ exist a $G' \in E'/\mathbb{K}$ and for the public key $P = [d]G \in E/\mathbb{K}$ exist a $P' = [d']G' \in E'/\mathbb{K}$ with the same secret key, because of $\mathcal{I}(P+Q) = \mathcal{I}(P) + \mathcal{I}(Q)$.

TODO: "*Given two elliptic curves $E/\mathbb{K}$ and $E'/\mathbb{K}$, with an isogeny application $\mathcal{I} : E/\mathbb{K} \to E'/\mathbb{K}$, the cyclic subgroup generator $G \in E/\mathbb{K}$ can be* translated *into $E'/\mathbb{K}$ as $G' = \mathcal{I}(G)$.*
The same way the public key $P \in E/\mathbb{K}$ becomes $P' = \mathcal{I}(P)$. But what about the secret key d? If it's preserved as the same value, the transformation is simply an automorphism, isn't it?"

TODO: "$P = [d]G \in E/\mathbb{K}$ *and* $\mathcal{I} : E/\mathbb{K} \to E'/\mathbb{K}$ *and* $P' = \mathcal{I}(P) = \mathcal{I}([d]G) = [d']\mathcal{I}(G) = [d']G'$ *then how to* migrate *d to d' from one curve to another?*"

FIXME: "*isogeny degree, composition of isogenies (prime degree)*"

Definition 11. *Given two isogenic elliptic curves $E/\mathbb{K}$ and $E'/\mathbb{K}$, the degree of the isogeny between this two curves is the degree of the rational map between them.*

For all the isogenies with a degree greater than 1, they can be factored into a composition of isogenies of a prime degree over the base finite field $\mathbb{F}_q$.

TODO: "*isogeny cases: Ascending, descending, horizontal*"

TODO: "*Isogeny class*"

Definition 12. *Given an elliptic curve $E/\mathbb{K}$ an endomorphism class or isogeny class of degree ℓ is the set of elliptic curves...*

Volcano, volcanoes ranges and starts FIXME: "*define a volcano: as a graph of isogenies of a particular degree.*"

To describe the structure of the ℓ degree isogeny class it has been proposed a graph. Using the different cases of isogenies in the same isogeny class two main parts of the graph can be described. There is a part of the graph that is cyclic, and from each node an edge goes down with a balanced ℓ-tree. All the nodes in the circumference have each tree with the same altitude. The number of edges of each node is $(\ell + 1)$

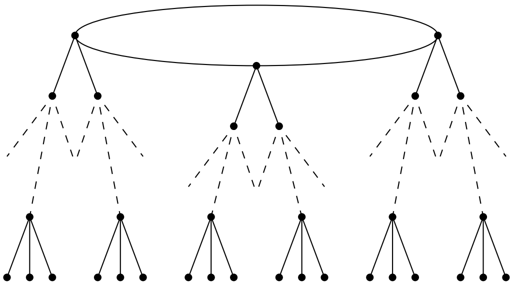

Fig. 1. A regular ℓ-volcano, with $\ell = 3$

FIXME: “*Is true the number of edges per node in figure 1?*”

FIXME: “*define “serralada” (volcanoes range or system): as the set of graphs of isogenies with many primer degrees.*”

Different isogeny degrees recreates different volcanoes, and a set of volcanoes can be also represented in a graph like a mountain system. A composed isogeny degree describes this type of system of volcanoes of the factors of this composed degree.

TODO: “*define isogeny stars: as a particular case of volcanoes with only crater. But can we use a volcano as a star? (Perhaps if the crater is big enough?)*”

FIXME: “*endomorphism ring*”

3.2 How the isogenies can help to avoid curve sharing?

To avoid to share a curve between all the users of the standard of the *rfc 6637* [2] using elliptic curve isogenies, it is not necessary to build completely the volcano or the volcanoes system. With a definition of a path to *walk* through the volcanoes system, and then *forget* the path (to avoid a cryptoanalyst to mode any attack calculation), will be enough *if and only if*, the path is large enough.

TODO: “*given an elliptic curve, generate an isogeny path through the volcanoes system, follow the path, forget it.*”

The isogenies has been already proposed to solve a problem in cryptology, in [14] a new cryptosystem is proposed who uses the isogenies for Diffie-Hellman key agreement [16] and ElGamal public key [17]. In this paper, the proposal of the authors is to use a route in an isogenies star as a secret key, and the public key is the destination of this route from a public well known elliptic curve. The fundamental of this cryptosystem is the strength of the problem to find a path between two curves, but is easy to go from one to another if you know the path.

The problem is alive, there are bibliography [18] to work in the direction of speed up the route walk in a volcanoes range (or in stars).

TODO: “*computational data about isogeny path evaluations*”

3.3 Front and side channel attacks to this schemas

The most important thing in cryptology, specially when an schema is proposed, is to think on how an adversary can try to break it. The question is what of the known things of the schema gives any advantage to this adversary and measure statistically the impact of the hypothetical advantage. In this proposed example of use elliptic curve isogenies to allow the users of this cryptography to avoid sharing the same elliptic curve but saving the security of use a cryptographically good elliptic curve, the mathematical cryptanalysis is by try to take an advantage on the task to discover the walked path in the isogeny star.

TODO: “*cryptanalysis on this matter: Given two curves isogenies between them, find a path*”[19]

Also there are other possible ways to break a cryptosystem that, further than attack directly the mathematics below, it tries to get the advantage from take a creative point of view over the application of the schema and that is called the side channel attacks.

TODO: "*possible side channel attacks (attacking the isogeny and to prevent attacks like* 'Zero-Value Points' *(ZVP))*"

TODO: "*Another prevention to side channel attacks is the use of projective and jacobian coordinates. Say also about protections over the possible attacks to the scalar product from section 1.2.*"

TODO: "*computational data about compute isogeny between to elliptic curves*"

4 Conclusion

TODO: "*what about the simulated times to generate curves in mathematical software? are they inside a usable range?*"

TODO: "*cryptanalysis of this creations*"

TODO: "*close with a remember of the security risk cited in the X9-62*"

TODO: "*remember the good skills of the elliptic curves over smart cards and the possibility of the reset of the cryptosystem setup*"

FIXME: "*where must be said this? other usages of curve change*"

References

1. "Ansi x9.62, public key cryptography for the financial services industry: The elliptic curve digital signature algorithm (ecdsa)."
2. A. Jivsov, "Elliptic Curve Cryptography (ECC) in OpenPGP." RFC 6637 (Proposed Standard), June 2012.
3. J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer, "OpenPGP Message Format." RFC 4880 (Proposed Standard), Nov. 2007. Updated by RFC 5581.
4. P. Gallagher, D. D. Foreword, and C. F. Director, "Fips pub 186-3 federal information processing standards publication digital signature standard (dss)," 2009.
5. V. Dolmatov, "GOST R 34.10-2001: Digital Signature Algorithm." RFC 5832 (Informational), Mar. 2010.
6. "Ecc brainpool standard curves and curve generation," October 2005.
7. "Sec 2. standards for efficient cryptography group: Recommended elliptic curve domain parameters."
8. S. Blanch-Torné and R. Moreno, "Análisis del cifrado elgamal de un módulo con curvas elípticas propuesto para el gnupg," in *II Simposio sobre Seguridad Informática*, (Zaragoza, Spain), pp. 35–41, Congreso Español De Informática, 2007.
9. L. C. Washington, *Elliptic Curves, Number Theory and Cryptography.* CRC Press, 2008.
10. N. Koblitz, "Elliptic curve cryptosystems," 1987.
11. "Sec 1. standards for efficient cryptography group: Elliptic curve cryptography."
12. "Ieee p1363 standard specifications for public key cryptography," January 2000.
13. D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography.* Springer, 2003.
14. A. Rostovtsev, E. Rostovtsev, and A. Stolbunov, "Public-key cryptosystem based on isogenies," 2006.

15. R. Arias and J. Miret, “Implementación de un criptosistema de clave pública basado en estrellas de isogenias,” Septembre 2007. TFC Ingeniería Informática UdL.
16. W. Diffie and M. E. Hellman, “New directions in cryptography,” 1976.
17. T. ElGamal, “A public-key cryptosystem and a signature scheme based on discrete logarithms.”
18. D. Jao and V. Soukharev, “A subexponential algorithm for evaluating large degree isogenies,” April 2010. 1002.4228v2.
19. L. D. Feo, “Fast algorithms for computing isogenies between ordinary elliptic curves in small characteristic,” February 2010. 1002.2597v1.