On trapdoor Kolmogorov one-way functions and elliptic curves cryptography

Francisco Mantero Morais Pavão Martins
francisco.pavao.martins@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

April 2014

Abstract

The purpose of this dissertation is to study the security of cryptographic systems based on elliptic curves, using Kolmogorov complexity and one-way functions as the main tools to analyze the security of each scheme.

The main motivation for this thesis is understand the individual approach of analyzing one-way functions using Kolmogorov complexity. Knowing that trapdoor one-way functions are sufficient to the construction of public key encryption and signature schemes and understanding elliptic curves and cryptography was also a motivation to analyze a cryptographic scheme based on elliptic curves and defined through trapdoor Kolmogorov one-way functions.

We will define a new family of functions and will call them *trapdoor Kolmogorov one-way functions*. We will also prove that for each function, the number of trapdoors is always lower, (by a polynomial fraction), than the number of possible trapdoors.

We will present a public key cryptographic system based on elliptic curves and we will denote by f a function that emulates the system. We will draft conclusions on the security of the cryptographic scheme, based on observations made on the size of each private key as we arrive to an asymptotic result that yields a lower bound on the length of each private key.

Assuming that ECDLP is not in **P**, we will prove that a function f is a Kolmogorov one-way function and, with the the help of auxiliar function, can be extended to an element of a family of trapdoor Kolmogorov one-way functions.

Keywords: Kolmogorov complexity. Trapdoor Kolmogorov one-way function. Elliptic curves cryptography. Cryptographic security.

1. Introduction

This thesis consists on defining a new family of functions called *trapdoor Kolmogorov one-way functions* and exploring the application of this new class in public key encryption. We look in detail to the case where the cryptographic scheme is based on elliptic curves.

The main motivation for this thesis is understand the individual approach of analyzing one-way functions using Kolmogorov complexity. Knowing that trapdoor one-way functions are sufficient to the construction of public key encryption and signature schemes and understanding elliptic curves and cryptography was also a motivation to analyze a cryptographic scheme based on elliptic curves and defined through trapdoor Kolmogorov one-way functions.

The Kolmogorov complexity, $K(x)$, (see [6] and [5]) of an object x is the length of the shortest program producing x in a universal Turing machine. The time-bounded version of Kolmogorov complexity $K^t(x)$, is the length of the shortest program producing x within time $(|x|)$.

Intuitively, a one-way function is a function that is easy to compute but hard to invert. The existence of these functions is an open question which implies $\mathbf{P} \neq \mathbf{NP}$, [7]. Given the importance of one-way functions and the impact of their applications, we analyze them at an individual level using Kolmogorov complexity. Classically there are several definitions of one-way functions, namely: strong, weak and deterministically, (see [4]). An interesting fact about strong and weak one-way functions is that, although their definitions are not equivalent, weak one-way functions exist if and only if strong one-way functions exist, see Proposition 5.

We introduce a new family of functions, that we call Trapdoor Kolmogorov one-way function. These are are Kolmogorov one-way functions f as in [1] with the extra property that there exists a polynomial time function h that for each function of the family, provides as input an extra information that one can use to invert the function f in polynomial

time. Following the same rational we define what trapdoor strong one-way function, trapdoor weak one-way function and trapdoor deterministic one-way function are. We set an upper bound for the number of possible trapdoors that each trapdoor Kolmogorov one-way function has.

The subject of elliptic curves is one of the jewels of the nineteenth-century mathematics, originated by Abel, Gauss, Jacobi and Legendre. In order to use elliptic curves in computational problems, one has to look to the case where elliptic curves are defined over a finite field. One of many problems studied when working with elliptic curves over finite fields, is the *elliptic curves discrete logarithm problem* defined as follow: Given two rational points P and Q one desires to find an integer x such that $xP = Q$, see details in Definition 10. This problem yields an EL Gamal encryption scheme 2.2. See more details in [2], [8] [3] and [10].

The main issue on each encryption scheme is the security of the scheme. One simple does not work with one scheme if this is not strong enough against different attacks. How secure an encryption scheme is and how can one evaluate this, are some of the questions that we attempt to answer through this work. We aim to build a cryptographic system that corresponds to the ECDLP and does not lie in **P**, we will see that in order to ensure this, we will impose restrictions to our set of private keys. Using the cryptographic system based on the elliptic curves we prove that the function f that emulates the system is honest, (the object and the image are polynomial related), injective and computable in polynomial time. These results will help us prove that if ECDLP is not in **P** then f is a Kolmogorov one-way function, see more details in Theorem 21.

Finally using the results obtained and our initial definition of trapdoor Kolmogorov one-way function, we can build a polynomial time function that outputs a trapdoor for f. Using this we can extend the definition of f to a trapdoor Kolmogorov one-way function candidate, see more in 23. It is important to note that all these results are obtained under the assumption that the ECDLP is not in **P**.

2. Backgroud

We present the basic definitions and results needed for the rest of this paper. In this first section we will present results on one-way functions using Kolmogorov complexity and on cryptographic schemes based on elliptic curves.

2.1. One-way functions

Definition 1. A function f is said to be *honest* if $|f(x)|$ and $|x|$ are polynomially related, i.e. for some $k > 0$ and for every $x \in \Sigma^*$ we have:

$$(|f(x)| \leq |x|^k + k) \wedge (|x| \leq |f(x)|^k + k).$$

From now on, we will consider f to be an honest function.

Definition 2 (Deterministic one-way function)**.** A function $f : \Sigma^* \to \Sigma^*$ is said to be a *deterministic one-way function* if the following two conditions hold:

1. There is a deterministic polynomial time algorithm A such that on every input x we have that $A(x) = f(x)$.
2. For any deterministic polynomial time algorithm B, for some polynomial $q(.)$ and for every sufficiently large n,

$$pr_{x\in\Sigma^n}[f(B(f(x), n)) \neq f(x)] > \frac{1}{q(n)}.$$

Definition 3 (Weak one-way function)**.** A function $f : \Sigma^* \to \Sigma^*$ is said to be a *weak one-way function* if the following two conditions hold:

1. There is a deterministic polynomial time algorithm A such that on every input x we have that $A(x) = f(x)$.
2. For any polynomial $t(.)$, there is a polynomial $q(.)$ such that for every probabilistic t-time bounded algorithm B and for every sufficiently large n,

$$pr_{x\in\Sigma^n}[f(B(f(x), r, n)) \neq f(x)] > \frac{1}{q(n)}.$$

Definition 4 (Strong one-way function)**.** A function $f : \Sigma^* \to \Sigma^*$ is said to be a *strong one-way function* if the following two conditions hold:

1. There is a deterministic polynomial time algorithm A such that on every input x we have that $A(x) = f(x)$.
2. For any polynomial $t(.)$, for every positive polynomial $q(.)$, for every probabilistic t-time bounded algorithm B and for every sufficiently large n,

$$pr_{x\in\Sigma^n}[f(B(f(x), r, n)) = f(x)] < \frac{1}{q(n)}.$$

In the previous definitions r denotes the randomness used by the algorithm B that tries to invert f. As we will see in the next proposition, it is very easy to relate these three definitions.

Proposition 5. Take $f : \Sigma^* \to \Sigma^*$.

1. If f is a strong one-way function, then f is a weak one way function.
2. If f is a weak one-way then f is a deterministic one-way function.

The next result relates the notion of strong one-way function, presented in Definition 4 and Kolmogorov complexity. This result will be used later to establish an upper bound for the number of trapdoors a function can have.

Theorem 6. Let f be an injective and polynomial time computable function. If f is a strong one-way function, then for every constant c and for every polynomial $t(.)$, the expected value of $K_f^t(x|f(x), r, n)$ over pairs $(x, r) \in \Sigma^n \times \Sigma^{t(n)}$, is larger than $c \log n$ for every sufficiently large n.

We will define a class of one-way functions using Kolmogorov complexity.

Definition 7. Let $f : \Sigma^* \to \Sigma^*$ be an injective and polynomial time computable function such that $|f(x)| = m(n)$ for all $x \in \Sigma^n$, where m is some polynomial. We say that f is a *Kolmogorov one-way function* if for every polynomial $t(.)$, for every positive integer c, for every sufficiently large n and for every x of length n,

$$K_f^t(x|n) - K_f^t(x|f(x), n) \leq c \log n.$$

We can easily relate a Kolmogorov one-way function with a deterministic one-way function. This result is presented in the next theorem.

Theorem 8. If f is a Kolmogorov one-way function then f is a deterministic one-way function.

The interested reader can find the proof of Proposition 5, the proof of Theorem 6 and the proof of Theorem 8 in [1].

2.2. Elliptic curves cryptography

Public key cryptographic systems are systems where the enciphering function is public. We will see, as an example, the El Gamal method for elliptic curves. This is a special method for the *discrete logarithm problem* as we present in the next definition.

Definition 9. Let $G = \langle g \rangle$ be a cyclic abelian finite group and $h \in G$. The *discrete logarithm problem (DLP)* is the following: Knowing G, g, h and finding whether there is $x \in \mathbb{Z}$ such that $h = g^x$ or not.

Let us consider E to be an elliptic curve defined over a finite field $\mathbb{F}_q$, where $q \in \mathbb{Z}$ and let us denote by $E(\mathbb{F}_q)$ the set of rational points of E.

Definition 10. Let E be an elliptic curve over a field $\mathbb{F}_q$ and P a point in $E(\mathbb{F}_q)$, then the *Elliptic Curve Discrete Logarithm Problem* or *ECDLP* on E is the following:

- Instance: Given a base point $P \in E(\mathbb{F}_q)$ and a point $Q \in E(\mathbb{F}_q)$.
- Question: Find an integer $x \in \mathbb{Z}$ such that $xP = Q$, if such an integer exists.

It is interesting to see that if one has access to factorization into primes p_i of:

$$n = |G| = \prod_{i=1}^{k} p_i^{e_i}$$

then one can reduce the ECDLP into a DLP. In fact, one has to reduce G into $G \mod p_i$ for each prime factor p_i of n and then apply the Chinese Remainder Theorem to build a DLP. Later we will find an example of such reduction. Using this reduction one can easily conclude that studying the security of an ECDLP is the same as studying the security of the DLP from which the ECDLP reduces to.

Definition 10 yields the cryptosystem that we will study. Let E be an elliptic curve over $\mathbb{F}_q$ and $P \in E(\mathbb{F}_q)$. The ECDLP is the question to know if a given point $Q \in E(\mathbb{F}_q)$, there exists an integer n with $Q = nP$ and if one can compute this n. The main point of this El Gamal method is that the DLP is hard to solve. The interaction in cryptosystem between the sender and the receiver is described as follows:

1. Sender has a secret message m that wants to send to the receiver.
2. Sender and Receiver agree on $P \in E(\mathbb{F}_q)$ public.
3. Receiver chooses $n \in \mathbb{Z}$ secretly, computes nP and sends it to the sender.
4. Sender picks $k \in \mathbb{Z}$ secretly, computes kP and $m + k(nP)$ and sends it to the receiver.
5. The receiver computes $m+k(nP)-n(kP) = m$ and obtains the secret message.

Both the *MOV attack* and the *anomalous attack* yield some necessary conditions regarding the security of the cryptographic system. Although we will not study these conditions deeply, (the interested reader can learn more about them in [3]), we will take them into consideration when building a cryptographic system. From now on we will only work with elliptic curves generated using an algorithm presented on Chapter 6 of [3].

We will now present an example of an attack that uses the baby step giant step method and the Chinese Remainder Theorem. This example will motivate us to impose a new restriction to the cryptographic system in order to make it more secure.

Example 11. *Consider the following elliptic curve,*

$$E : Y^2 = X^3 + 71X + 602$$

over the finite field $\mathbb{F}_{1009}$*. The group order of* $E(\mathbb{F}_{1009})$ *is* $1060 = 2^2.5.53$*. Suppose the two points:*

$$P = (1, 237), \quad Q = (190, 271)$$

are given and the solution to the $Q = [m]P$*. First notice that* P *has order* $530 = 2.5.53$ *in the group* $E(\mathbb{F}_{1009})$*. Hence by the above reduction of Pohlig and Hellman, the computation of* m *can be reduced to the computation of* m *modulo 2,5 and 53. Lets start by computing the solution modulo 2.*

One multiplies P *and* Q *by* $\frac{n}{2} = \frac{530}{2} = 265$*. This leads to:*

$$P_2 = [265]P = (50, 0)$$

$$Q_2 = [265]Q = (50, 0).$$

The system to find $m \mod 2$ *is simply given by:*

$$Q_2 \equiv [m \mod 2]P_2,$$

hence $m \equiv 1 \mod 2$*.*

We will now do the same for $m \mod 5$*. One multiplies* P *and* Q *by* $\frac{530}{5} = 106$*. This leads to the following:*

$$P_5 = [106]P = (639, 160)$$

$$Q_5 = [106]Q = (639, 849)$$

$$Q_5 \equiv [m \mod 5]P_5.$$

Hence $m \equiv 4 \mod 5$*.*

Finally we do the same for modulo 53. We multiply P *and* Q *by* $\frac{530}{53} = 10$ *and obtain the following:*

$$P_{53} = [10]P = (32, 737)$$

$$Q_{53} = [10]Q = (592, 97).$$

Clearly we could use brute force to calculate the value of m *module 53 but instead we will use the baby step giant step method.*

As we have seen, we have to calculate the value of a *and* b *in the following equation:*

$$m = \lceil \sqrt{n} \rceil a + b.$$

In this case we take $n = 53$ *and* $\lceil \sqrt{53} \rceil = 8$*, this means that one needs 8 baby steps. After computing the baby steps one computes one giant step at a time and compares it with the baby steps computed before. One notices an identity with* $a = 6$ *and* $b = 0$*, what leads to:*

$$\begin{aligned} m &= 8a + b \\ &= 8.6 + 0 \\ &= 48 \\ \implies & m \equiv 48 \mod 53. \end{aligned}$$

Using the three results and Chinese Remainder Theorem one has that $m = 419$*.* ▷

3. Results

We present a new class of functions called *trapdoor Kolmogorov one-way functions*. We study the security of a cryptographic system and propose a candidate function that emulates the system and is a trapdoor Kolmogorov one-way function.

3.1. Trapdoor Kolmogorov one-way function

We will look in detail to the case where our one-way function has a trapdoor that provides extra information. The trapdoor will be important to extract some extra information about the function in study. We will define these functions through Kolmogorov complexity and prove some results.

We will study one-way functions that map objects from a set of arity n to a set of arity $m(n)$, meaning

$$f : \Sigma^n \to \Sigma^{m(n)},$$

where m is some polynomial. We will consider these functions as a family $\{f_n\}_{n\in\mathbb{N}}$.

Definition 12. Let $\{f_n\}_{n\in\mathbb{N}}$ be a family of Kolmogorov one-way functions, such that:

$$f_n : \Sigma^n \to \Sigma^{m(n)}.$$

Where m is some polynomial. We say that $\{f_n\}_{n\in\mathbb{N}}$ is a *trapdoor Kolmogorov one-way function family* if there is a function:

$$h : \mathbb{N} \to \Sigma^t(n),$$

for t polynomial time function, such that

$$K^t_{f_n}(x|f_n(x), h(n), n) \in \mathcal{O}(1).$$

We call h our *trapdoor function.*

Taking advantage of this new definition we can define strong, weak and deterministic one-way functions with the trapdoor property.

Definition 13. We say that a family of $\{f_n\}_{n\in\mathbb{N}}$ is a family of *trapdoor strong one-way function*, *trapdoor weak one function* or *trapdoor deterministic one-way function* if each element f_n of the family is respectively a strong one-way function, weak one-way function or deterministic one-way function and if there is a computable function $h : \mathbb{N} \to \Sigma^n$ such that:

$Pr_{(x,h(n))\in\Sigma^n\times\Sigma^{t(s_n)}}[f_n(B(f_n(x), h(n), n)) = f_n(x)] = 1.$

From now on, every time we say that f trapdoor one-way function, we will be referring to a function f that belongs to a family of trapdoor one-way functions.
We can prove a result similar to the result of Theorem 8 for this new definition.

Proposition 14. If f is a trapdoor Kolmogorov one-way function, then f is a trapdoor deterministic one-way function.

Proof. From Theorem 8 we know that if f is a kolmogorov one-way function, then f is a deterministic one-way function. We just have to proof the trapdoor property. Since f is a trapdoor Kolmogorov one-way function we know that:

$$K^t_{f_n}(x|f_n(x), h(n), n) \in \mathcal{O}(1).$$

Then for any x there is an algorithm A that with $f_n(x), h(n)$ and n, it outputs x, i.e. for a universal Turing machine U we have that $U(A(f_n(x), h(x), n) = x$. We know that any A has constant size. Lets take $c = max\{|A| : \forall x, U(A(f_n(x), h(x), n) = x\}$. There are 2^c possible algorithms A. Lets consider B to be an algorithm such that it runs all possible algorithm A with $f(x), h(n)$ and n and it checks whether the output is x or not. Then the following holds:

$$Pr_{(x,h(n))\in\Sigma^n\times\Sigma^{t(s_n)}}[(B(f_n(x), h(n), n)) = x] = 1.$$

It remains to show that B runs in polynomial time. Since c is a constant we have that 2^c is a constant and since A runs in polynomial time, we have that B runs in $2^c TIME(A)$ which is also polynomial, where by $TIME$ we understand the polynomial running time of A. □

For a given one-way function f, if for every $s \in \mathbb{N}$, we have that s is a trapdoor for f, then f does not provide any security for the problem. Therefore we aim to find one-way functions with a limited number of trapdoors. The next theorem imposes some limits to the number of trapdoors for each problem. We prove this theorem for strong one-way functions. From Proposition 5 we can extend this result to weak and deterministic one-way functions.

Proposition 15. Let f be a trapdoor strong one-way function and

$$H = \{r \in \Sigma^{t(s_n)} : K^t_f(x|f(x), r, n) \in \mathcal{O}(1)\}$$

Then for n large enough and $x \in \Sigma^n$ we have that:

$$\#H = h < 2^{t(s_n)} - \frac{c\log(n)q(n)}{2^{n-1}}$$

Proof. From Theorem 6 we know that if f is injective and a strong one-way function, then for every constant c and for every polynomial $t(.)$ we have that:

$$E[K^t_f(x|f(x), r, n)] > c\log(n)$$

Lets explore the expected value:

$$\begin{aligned}
&\textstyle\sum_{x\in\Sigma^n}\sum_{r\in\Sigma^{t(s_n)}} Pr(B(f(x), r, n) = x)k^t_f(x|f(x), r, n)\\
&> c\log n\\
&\Leftrightarrow\\
&\textstyle\sum_{x\in\Sigma^n}\sum_{r\in\Sigma^{t(s_n)}} \frac{1}{q(n)}k^t_f(x|f(x), r, n) > c\log(n)\\
&\Leftrightarrow\\
&\textstyle\sum_{x\in\Sigma^n}\sum_{r\in H} \frac{1}{q(n)}k^t_f(x|f(x), r, n)\\
&\textstyle+\sum_{x\in\Sigma^n}\sum_{r\in\Sigma^{t(s_n)}\setminus H} \frac{1}{q(n)}k^t_f(x|f(x), r, n)\\
&> c\log(n)\\
&\Leftrightarrow\\
&2^{n-1}h\mathcal{O}(1) + \textstyle\sum_{x\in\Sigma^n}\sum_{r\in\Sigma^{t(s_n)}\setminus H} k^t_f(x|f(x), r, n)\\
&> c\log(n)q(n)\\
&\Leftrightarrow\\
&h\mathcal{O}(1) + (2^{t(s_n)} - h)(n + \mathcal{O}(1)) > \tfrac{c\log(n)q(n)}{2^{n-1}}\\
&\Leftrightarrow\\
&-hn > \tfrac{c\log(n)q(n)}{2^{n-1}} - 2^{t(s_n)}(\mathcal{O}(1) + n)\\
&\Leftrightarrow\\
&h < 2^{t(s_n)} - \tfrac{c\log(n)q(n)}{n2^{n-1}}.
\end{aligned}$$

□

3.2. On security of an El Gamal scheme
Based on this Example 11, we can draw some conclusions on the security of the cryptographic system based on elliptic curves. The first conclusion one can obtain is that in order for the system to be secure the largest prime p dividing $\#E(\mathbb{F}_q)$ has to be very large. Other assumptions, to obtain, in principal a more secure curve, can be made on the type and size of the private keys as we will see in the following results.

Proposition 16. Let us take an elliptic curve E over $\mathbb{F}_q$, p the largest prime dividing $\#E(\mathbb{F}_q)$ and an ECDLP associated to E and m a private key to be used in that curve. If we take $m \equiv 0, 1, -1$ mod p then the ECDLP is in **P**.

Proof. Without loss of generality lets assume that $m \equiv 1 \mod p$, where p is the largest prime dividing $\#E(\mathbb{F}_q)$. By Pohlig and Hellman reduction one can find the value of m. This is done in polynomial time. The other cases are similar to prove. □

Example 11 establishes a cryptosystem based on a single elliptic curve. From now on, we will be interested in working with a chain or family of elliptic curves. For each elliptic curve we will have a cryptosystem associated as the one presented in Example 11.

Consider $\mathcal{P}$ to be set of prime numbers in $\mathbb{N}$. For each prime $p \in \mathcal{P}$, we consider the following family of elliptic curves:

$$\{E(\mathbb{F}_p)\}_{p\in\mathcal{P}}.$$

For $i, j \in \mathcal{P}$, if $i > j$, then $\#E(\mathbb{F}_i) > \#E(\mathbb{F}_j)$. To ease on notation, from now on, we will denote $E(\mathbb{F}_i)$ by E_i and $\#E(\mathbb{F}_i)$ by n_i.

For each elliptic curve E_i, we will associate a cryptographic system as the one presented in Scheme 2.2. For each E_i we will fix a pair $(m_i, l_i) \in \mathbb{N}^2$ of private keys and an element $P_i \in E_i$, the public key, and we will denote the cryptographic system by $(E_i, (m_i, l_i), P_i)$. Our goal is to assure that each $(E_i, (m_i, l_i), P_i)$ is secure. For this and based on Example 11 we will impose some restrictions on the set of private keys as the following proposition will denote.

Proposition 17. Consider $(E_i, (m_i, l_i), P_i)$ and a polynomial time function t such that $K^t(m_i) \in \mathcal{O}(\log\log n_i)$, where $K^t(m_i)$ is the Kolmogorov Complexity of m_i. Then the ECDLP associated to this system is in $\mathbf{P}$.

Proof. Lets take $P, Q \in E_i$ such that $Q = [m_i]P$ and consider the polynomial time function t. We denote $\#E_i$ by n_i. We know that

$$K^t(m_i) \in \mathcal{O}(\log\log n_i).$$

Consider the algorithm A that for each candidate x for m_i, tests if

$$Q = xP.$$

There are $2^{c\log\log n_i}$ possible candidates, for $c \in \mathbb{R}^+$. Manipulating this result we get that:

$$\begin{aligned} 2^{c\log\log n_i} &= 2^{\log\log {n_i}^c} \\ &= \log {n_i}^c \\ &= c\log n_i. \end{aligned}$$

Therefore there is a polynomial number of candidates and one can easily conclude that $A \in \mathbf{P}$. □

Based on this result, for each elliptic curve E_i, we associate a cryptosytem $(E_i, (m_i, l_i), P_i)$ such that $m_i, l_i \in \mathcal{O}(\log n_i)$. One can also notice that Proposition 16 is a particular case of Proposition 17.

3.3. Building a Kolmogorov one-way function candidate

Let us consider the elliptic curve E and establishes as basis for the presentation a cryptosystem $(E, (m, l), P)$. We consider a function f such that for a fixed elliptic curve E we have that for a fixed $P \in E(\mathbb{F}_q)$ and for $m, l \in \mathbb{N}$ that:

$$\begin{aligned} f : E(\mathbb{F}_q) &\to E(\mathbb{F}_q)^4 \\ x &\mapsto (x + [l][m]P, [l]P, [m]P, P). \end{aligned} \tag{1}$$

This function emulates the last interaction of the cryptosystem presented in Scheme 2.2.

Our goal is to build a Kolmogorov one-way function. We will start by showing that f is an function.

Proposition 18. Function f 1 is honest.

Proof. Recall that by $|x|$ we understand the length of the binary string representing x. Lets take $P = (x_1, y_1) \in E(\mathbb{F}_q)$ and assume that:

$$|P| = |x_1| + |y_1|.$$

Lets now take $f(x) = (x+[l][m]P, [l]P, [m]P, P)$ and consider the following:

- $[l][m]P = P_1$
- $[l]P = P_2 \in E(\mathbb{F}_q)$.
- $[m]P = P_3 \in E(\mathbb{F}_q)$.

Then the length of the binary string representing $f(x)$ is given by:

$$\begin{aligned} |f(x)| &= |x + [l][m]P| + |P_2| + |P_3| + |P| \\ &\leq |x| + |P_1| + |P_2| + |P_3| + |P|. \end{aligned}$$

Lets now set $\#E(\mathbb{F}_q) = n$. We know that $|n| \approx \log n$, then for any $Q \in E(\mathbb{F}_q)$, we have that $Q \leq 2\lceil\log n\rceil$ and $Q > 0$. Following the same rational, for $m, l \in \mathbb{N}$ and for t polynomial time function we have:

$$K^t(m), K^t(l) \leq \log n.$$

Therefore one can easily see that $|m| \leq \log n$ and the same for l. Then we have that:

- $|P_1| \leq 2\log n$.
- $|P_2| \leq 2\log n$.
- $|P_3| \leq 2\log n$.
- $|P| \leq 2\log n$.

We will consider $k = 8\lceil\log n\rceil$, then one can easily see that:

1. $|f(x)| \leq |x|+|P_1|+|P_2|+|P_3|+|P| \leq (|x|)^k+k$.
2. On the other hand since $|x| < k$ by definition of k and $|f(x)|¿0$, we have that $|x| \leq |f(x)|^k + k$.

The two arguments ensure that f is honest. □

Our next step is to prove that f 1 is injective. Notice that for a fixed $P \in E(\mathbb{F}_q)$ and for $m, l \in \mathbb{N}$, we have that:

$$f(x) = (x + [l][m]P, [l]P, [m]P, P)$$

where $[l]P$, $[m]P$ and P are independent from x and therefore always take the same value independently of x. Therefore when studying the injectivity of f, one only have to consider the first entrance of the output. Lets consider the function:

$$\begin{array}{rcl} f' : E(\mathbb{F}_q) & \to & E(\mathbb{F}_q) \\ x & \mapsto & x + [l][m]P \end{array} \quad (2)$$

where $P \in E(\mathbb{F}_q)$ is a fixed element and $l, m \in n$ are also fixed. If we prove that f' is injective, then f is obviously injective.

Proposition 19. Function f' 2 is injective.

Proof. Lets take $a, b \in E(\mathbb{F}_q)$ such that $a \neq b$ and consider

- $f'(a) = a + [l][m]P$.
- $f'(b) = b + [l][m]P$.

Lets assume by absurd that $f'(a) = f'(b)$, then this implies that

$$a + [l][m]P = b + [l][m]P$$

This is the same as saying that

$$a + [l][m]P - (b + [l][m]P) = \mathcal{O}.$$

We know that $E(\mathbb{F}_q)$ an algebraic group, therefore is associative and commutative, hence we have that

$$\begin{array}{ll} & a - b + [l][m]P - [l][m]P = \mathcal{O} \\ \Leftrightarrow & a - b = \mathcal{O} \\ \Leftrightarrow & a = b, \end{array}$$

which is a contradiction with our initial assumption. □

One can then easily conclude that f as in 1 is injective.

The next property we will show is that f is computable in polynomial time. The computations of $[l][m]P, [l]P$ and $[m]P$ are pre computed and therefore one does not consider them when calculating the computational power requisites of the the function f, however they are computed in polynomial time as the interested reader can see in Chapter IV of [3].
Since $E(\mathbb{F}_q)$ is an algebraic group, there is a rule for the sum under the group, which is computable in polynomial time, find more in [9]. Then $x + [l][m]P$ is computed in polynomial time and one can easily conclude that f is computed in polynomial time.

The last property that we check is that the length of the binary string representing $f(x)$ is given by a polynomial $m(n)$, where $n = |x|$.

Proposition 20. There exists a polynomial $m : \mathbb{Z} \to \mathbb{Z}$ such that for $|x| = n$ we have that

$$|f(x)| = m(n).$$

For f 1.

Proof. We fix $P \in E(\mathbb{F}_q)$ and $m, l \in \mathbb{N}$ and calculate the value of $f(x)$, therefore $|[l][m]P|, |[l]P|, |[m]P|, |P|$ are fixed constants.
From the definition of f, we know that:

$$|f(x)| = |x + [l][m]P| + |[l]P| + |[m]P| + |P|.$$

On the other hand by triangle inequality we know that

$$|x + [l][m]P| \leq |x| + |[l][m]P|.$$

Then using this fact and since $|x + [l][m]P| \geq 0$, one can easily see that

$$\begin{array}{l} |[l]P| + |[m]P| + |P| \\ \leq |f(x)| \\ \leq |x| + |[l][m]P| + |[l]P| + |[m]P| + |P|. \end{array}$$

One can easily conclude that there exists a polynomial $m(n)$ such that $|f(x)| = m(n)$, where $n = |x|$, that lies between these values. □

We have seen that f presented in 1 is honest, injective, computable in polynomial time and exists a polynomial m such that $|f(x)| = m(|x|)$.
We are finally on a stage where we have all the machinery to build a connection between our function f and the notion of Kolmogorov one-way function presented in Definition 7. The next theorem states that under the assumption that the ECDLP $\notin \mathbf{P}$ then f is a Kolmogorov one-way function. We will consider again our family of elliptic curves and to each curve E_i we will associate a function f_i that behaves as the function f that we have just built.

Theorem 21. Consider a family of elliptic curves

$$\{E_i\}_{i \in \mathcal{P}}.$$

For each curve E_i take $n_i = \#E_i$ and consider a function f_i that emulates the ECDLP associated to E_i and is honest, injective, computable in polynomial time and there exists a polynomial m such that $m_i(|x|) = |f_i(x)|$.

If the ECDLP $\notin \mathbf{P}$ then there is a polynomial time function t and an infinity set of keys (m_i, l_i) such that:

- $K^t(m_i) > \mathcal{O}(\log\log n_i)$,
- $K^t(l_i) > \mathcal{O}(\log\log n_i)$,
- $K^t(m_i l_i) > \mathcal{O}(\log\log n_i)$,

and for each E_i the function f_i is a Kolmogorov one-way function.

Proof. We will prove this theorem by contraposition. Suppose that there exists $i \in \mathbb{N}$ such that for all $j > i$ one has:

$$(E_j, (m_j, l_j), P_j)$$

where $min\{K^t(m_j), K^t(l_j), K^t(m_j l_j)\} \leq \mathcal{O}(\log\log n_j)$ for some t polynomial time function. Take $E = E_h$, such that $h > i$ and consider the following,

$$\begin{aligned} K_f^t(x|f(x), n) &= K_f^t(x|x + lmP, lP, mP, P, n) \\ &\leq K_f^t(lmP|lP, mP, P, n) \\ &\leq min\{K_f^t(l|mP, lP, P, n), \\ &\quad K_f^t(m|mP, lP, P, n)\} \quad (*) \end{aligned}$$

By Proposition 17 there exists an algorithm A that solves equation $(*)$ in polynomial time, hence the following is true,

$$K_f^t(x|f(x), n) \in \mathcal{O}(1).$$

One can conclude that f is not Kolmogorov one-way function. □

Corollary 22. Under the assumption that ECDLP $\notin \mathbf{P}$, f is a deterministic one-way function.

Corollary 22 is an immediate result from Theorem 8.

3.4. Building a trapdoor Kolmogorov one-way function candidate

From Example 11 we understand that the security of an El Gamal cryptosystem over an elliptic curve is based on the fact that it is hard to solve the logarithm problem. In this section, we will study the case where we are provided with extra information, a trapdoor, that helps us solve the problem.

As a result of last section we will consider the following set up for our problem.

Consider a family of elliptic curves of the following form,

$$\{E_i\}_{i\in\mathbb{N}}$$

We set $\#E_i = n_i$ and for $i, j \in \mathbb{N}$ with $i > j$, we have that $n_i > n_j$. For each elliptic curve we consider an El Gamal cryptosystem represented as:

$$(E_i(m_i, l_i), P_i),$$

such that for a polynomial time function t we have that

- $K^t(m_i) \in \mathcal{O}(\log n_i)$.
- $K^t(l_i) \in \mathcal{O}(\log n_i)$.
- $K^t(l_i m_i) \in \mathcal{O}(\log n_i)$.

Each cryptosystem $(E_i, (m_i, l_i), P_i)$ has a function f_i associated that emulates the last interaction in the cryptosystem and is given by,

$$\begin{aligned} f_i : E_i &\to E_i^4 \\ x &\mapsto (x + [l_i][m_i]P, [l_i]P, [m_i]P, P). \end{aligned}$$

As a result of last Section, we have seen that f_i is injective, honest, computable in polynomial time and if ECDLP $\notin \mathbf{P}$, then f_i is a Kolmogorov one-way function.

It is important to note that if one has access to the value of m_i, l_i or m_i, l_i, then one can easily extract the value of x. This notion is very simple, but fundamental for our notion of trapdoor. We will denote m_i by trapdoor for the system $(E_i, (m_i, l_i), P_i)$.

Since we are working with a family of elliptic curves, we want to define a function, that given an elliptic curve returns the value of a trapdoor of the cryptosystem associated to our elliptic curve. We will consider the following:

$$\begin{aligned} \varphi : \mathcal{P} &\to \mathbb{N} \\ i &\mapsto m_i \end{aligned} \tag{3}$$

Recall the definition of trapdoor Kolmogorov one-way function presented in Definition 12. We will show in the next theorem, that if the ECDLP is not in $\mathbf{P}$, then the family of functions $\{f_i\}_{i\in\mathbb{N}}$ that emulates cryptosystems are trapdoor Kolmogorov one-way functions.

Theorem 23. Assume that the ECDLP is not in $\mathbf{P}$ then there are infinitely many m_i such that each function of the family of $\{f_i\}_{i\in\mathbb{N}}$ associated to the $(E_i, (m_i, l_i), P_i)$ is a trapdoor Kolmogorov one-way function as in Definition 12.

Proof. For each system $(E_i, (m_i, l_i), P_i)$ consider the function f_i that emulates the system. From Theorem 13, we know that if ECDLP $\notin \mathbf{P}$, then f_i is a Kolmogorov one-way function.

Consider the function φ 3 presented before. We will take φ to be our trapdoor function. We can consider the following godelization

$$g : \mathbb{N} \to \mathcal{P}.$$

To ease on notation we will take

$$\begin{aligned} f_i(x) &= (x + [l_i][m_i]P_i, [l_i]P_i, [m_i]P_i, P_i) \\ &= (f_{i,1}, f_{i,2}, f_{i,3}, f_{i,4}). \end{aligned}$$

Consider the algorithm A that receives $(g(\varphi(i)), f_i(x))$ as an input and returns:

$$f_{i,1} - g(\varphi(i))f_{i,2} = x.$$

Clearly A runs in polynomial time, hence $K^t_{f_i}(x|f_i(x), \varphi(i), i) \in \mathcal{O}(1)$ and we conclude that $\{f_i\}_{i n \mathbb{N}}$ is a trapdoor Kolmogorov one-way function family. □

Corollary 24. Assume that ECDLP is not in **P**, then each f_i is a trapdoor deterministic one-way function.

Corollary 24 is an immediate result from Proposition 14.

4. Conclusions

In this work we have introduced the concept of trapdoor Kolmogorov one-way function family and proved that for each function of this family, the number of trapdoors is always lower, (by a polynomial fraction), than the number of possible trapdoors.

We have also presented a public key cryptographic system based on elliptic curves and we defined a function f that emulates the system.

Based on results from Kolmogorov complexity, we provided restrictions on the set of private keys of the cryptographic system. These restrictions ensure securer system against possible attacks.

With the assumption at hand that ECDLP is not in **P** we have shown that every function that emulates our cryptographic system is in fact a Kolmogorov one-way function. Furthermore we have seen that each of these functions is an element of a family of trapdoor Kolmogorov one-way functions.

These results leads us to an individual way to approach security that might not rely on a computational hardness assumption.

As part of future work, interesting open questions consist in relate the notion of Kolmogorov one-way functions with the notion probability functions and establish relations between Kolmogorov one-way functions and strong and weak one-way functions.

The result obtained in this work regarding the number of trapdoors a trapdoor Kolmogorov one-way function has is not very restrict and it is in our interest to research for a more strict result.

Finding trapdoor Kolmogorov one-way function candidates using different encryption schemes as well as using Kolmogorov complexity to ensure a more secure system is another possibility for future work.

Acknowledgements

I would like to express my grattitude to my supervisor, Paulo Mateus, for presenting me an interesting path to research and for guidance through the elaboration of this dissertation.

I also want to thank my co-supervisor, André Souto for his outstanding dedication and support when help was needed.

I want to thank Professor Klaus Altmann for guiding me in the first steps of this work and to motivate me to learn more about algebraic geometry.

I want to take this opportunity to thank all the people in the Department of Mathematics of IST, and in the Computer Science Section, for their support. In special, I wish to thank my classmates and former classmates that have always motivated and supported me through out this path.

I am also grateful to all my friends who provided me unconditional support, and for giving me a space to relax and maintain my health of mind and spirt.

Last but not least I wish to give my very special thanks to my family, for their interest and love, specially to my parents who have always supported me and provided me with the best resources and opportunities and to my grandfather who has always motivated me to achieve success in my life.

References

[1] L. F. C. Antunes, A. Matos, A. Pinto, A. Souto, and A. Teixeira. One-way functions using algorithmic and classical information theories. *Theory Comput. Syst.*, 52(1):162–178, 2013.

[2] R. Balasubramanian. *Elliptic curves and cryptography.*, pages 325–345. New Delhi, IN: Hindustan Book Agency, 2003.

[3] I. F. Blake, G. Seroussi, and N. P. Smart. *Elliptic curves in cryptography.* London Mathematical Society lecture note series. Cambridge University Press, Cambridge, New York, 1999. Autres tirages : 2000, 2001, 2002, 2004.

[4] O. Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques.* Cambridge University Press, 2001.

[5] T. J. Lee. Kolmogorov complexity and formula size lower bounds. 2006.

[6] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications.* Texts in computer science. Springer, 2009.

[7] L. Longpré, S. Mocas, and M. C. o. C. S. Northeastern University (Boston. *Symmetry of Information and One-way Functions.* Technical report (Northeastern University (Boston,

Mass.). College of Computer Science)). Northeastern University, College of Computer Science, 1991.

[8] S. Schmitt and H. Zimmer. *Elliptic Curves: A Computational Approach*. De Gruyter studies in mathematics. Walter de Gruyter, 2003.

[9] J. Silverman. *The Arithmetic of Elliptic Curves*. Graduate texts in mathematics. Springer, 2009.

[10] D. Stinson. *Cryptography: Theory and Practice, Third Edition*. Discrete Mathematics and Its Applications. Taylor & Francis, 2002.