Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 27 (2024), No. 1, pp. 151–174
DOI : 10.47974/JDMSC-1638

On a Vernam cipher scheme based on elliptic curve L-functions

Sami Omar *
Abdulla Eid †
Department of Mathematics
University of Bahrain
P.O. Box 32038
Sukhair
Bahrain

Abstract

In this paper, we present a new Vernam-like cryptosystem based on the L-functions attached to elliptic curves. Actually, we show that for infinitely many elliptic curves, the coefficients of the L-series can be considered as a secure cryptographic cipher key when combined with linear and classical pseudo-random number generators. Hence, fast and widely known classical generators, like linear congruential generators, should be reintroduced into our list of cryptographic design choices.

Subject Classification: *(2010) 94A60, 11G05, 11T71, 14G50, 14H52.*

Keywords: *Elliptic curves, L-functions, LCGs, Vernam cipher.*

1. Introduction

In cryptography, the Vernam cipher or one-time pad, the mother of all encryption schemes, is a type of encryption that is very difficult to crack if used properly. The Vernam cipher is a stream cipher in which the plaintext is XORed with another piece of text (called key-tape) at least as long as the message to be encrypted. If the key-tape used is **truly random** and **used only once** then this is called a one-time pad. Claude Shannon [22] proved mathematically that one-time pad is completely secure. A cipher is considered computationally secure if the probability of cracking the encryption key within a reasonable time is extremely small, even more, impossible. In theory, as being the only one mathematically proved to be completely secure, any other cryptographic algorithm cipher can be

This article has been corrected with minor changes. These changes do not impact the academic content of the article.

* *E-mail:* **somar@uob.edu.bh (Corresponding Author)**
† *E-mail:* **aeid@uob.edu.bh**

cracked given enough time and cipher-text. Despite its formal Shannon's security proof, the one-time pad presents many disadvantages in practice. Indeed, the following conditions should be met:

- The key-tape is used only once (hence "one time").
- The key-tape must be immediately destroyed after any use.
- The key-tape must be as long as the message to be encrypted.
- The key-tape contains truly random characters.

An obvious disadvantage of the one-time pad is that the key-tape should be as long as the plain text, which increases the difficulty of key-tape distribution and key-tape management. One idea is to use a small cryptographic key, to be exchanged instead of a huge key-tape.

Furthermore, the Vernam cipher is vulnerable to the known-plain-text attack and key reuse. Indeed, let two messages M_1 and M_2, both encrypted using the same key-tape $\mathcal{K}$. An adversary who intercepts $E(M_1) = M_1 \oplus \mathcal{K}$ and $E(M_2) = M_2 \oplus \mathcal{K}$, can easily compute $E(M_1) \oplus E(M_2) = M_1 \oplus M_2$. So, if one message is longer than the other, then our adversary just truncates the longer message to the size of the shorter one and his attack will reveal that portion of the longer message.

It is also known that the one-time pad shares a feature with so-called deniable encryption schemes introduced in [8]. Deniable encryption allows an encrypted message to be decrypted to different sensible plain-texts, depending on the key used. Let $C = M_1 \oplus \mathcal{K}_1$, then Alice is able to produce another key $\mathcal{K}_2 = C \oplus M_2$, forcing the cipher-text C to be decrypted to the plain-text M_2. Therefore, the holder of the cipher-text will not have the means to differentiate between the true plain-text, and the bogus-claim plain-text. This scenario is especially dangerous when a member of the group is not always honest. Of course, deniablity can also be used in good scenarios and causes. For instance, this can be used as a trapdoor information to Eve. More precisely, if Bob sends to Eve a key $\mathcal{K}$ to the cipher-text C sent by Alice, then there is no reason for her to believe that he gave her the right key to decipher the original plain-text. Also, deniable encryption will be helpful when authorities uses force to give them the one-time pad $\mathcal{K}$. Instead, one can provide a fake key... etc.

In this paper, we present a new Vernam-like encryption scheme based on Fourier coefficients of the L-series of elliptic curves over the rational numbers. The study of the L-functions in cryptology was limited to the construction of Hash functions: we may mention the ECOH algorithm [7], the work of Omar et al. [18] and the pioneering work of M. Anshel and

D. Goldfeld [2]. In this study, we show how the techniques of encryption, decryption and digital signature can be integrated in a new authenticated encryption scheme based on the *L*-series of elliptic curves cryptosystem. We show that our candidate is secure against known plain-text attack, much less vulnerable against key reuse and represents a non-deniable encryption scheme. That is, we demonstrate that neither Alice nor Bob can produce a second plain-text M_2 which can be decrypted from *C* by using some other key-tape $\mathcal{K}_2$.

2. Elliptic Curve and One-way Function

In computer science, a one-way function must be easy to compute for every input and hard to invert with a prescribed image of a random input. Informally, a function *f* is a one-way function if

1. The function *f* is known publicly and does not need any secret information for its operation.
2. Given *x*, it is fast to compute *f*(*x*).
3. Given *y*, in the range of *f*, it is hard to find *x* such that *f*(*x*) = *y*.

The existence of such one-way functions remains an open problem. One-way functions, in this sense, are fundamental tools for cryptography, personal identification, authentication, and other data security applications. In modern cryptography, there are several candidates based on number theoretical problems, such as integer factorization and the computation of discrete logarithms, that have withstood for many years. Therefore, some of them turn out to be essential for telecommunications, e-commerce, and e-banking systems around the world.

In 1997, M. Anshel and D. Goldfeld [2] introduced a new problem arising from the theory of zeta functions which yields a new class of one-way functions based on the arithmetic theory of zeta functions. To explain this new candidate, we need to introduce some background related to *L*-functions arising from the elliptic curves.

An elliptic curve over $\mathbb{Q}$ is a non-singular cubic plane curve given by the Weierstrass equation

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

where the coefficients $a_i \in \mathbb{Q}$. From these coefficients, we derive the auxiliary quantities

$$\begin{cases} b_2 = a_1^2 + 4a_2, \\ b_4 = 2a_4 + a_1 a_3, \\ b_6 = a_3^3 + 4a_6, \\ b_8 = a_1^2 a_6 + 4a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2, \end{cases}$$

the invariants

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

and the discriminant $\Delta_E = -b_2^2 b_8 - 8b_4^3 - 27b_6^2 + 9\, b_2 b_4 b_6$.

The discriminant Δ_E must be non zero for the curve to be non-singular. For such non-singular elliptic curve the *j*-invariant can be defined as $j = \frac{c_4^3}{\Delta_E}$. By completing the square on the left hand side of the Weierstrass equation and performing an appropriate variable substitution eliminates the *xy* and *y* terms. In addition, a similar trick eliminates the term x^2 from the right hand side so that we obtain the following simple equation

$$E : y^2 = x^3 + ax + b,$$

where $a = -27\, c_4$ and $b = -54\, c_6$. One easily verifies that $\Delta_E = -4a^3 - 27b^2$. The set of $\mathbb{Q}$ -rational points of *E* is denoted by

$$E(\mathbb{Q}) = \text{point at infinity} \bigcup \{(x, y) \in \mathbb{Q} \times \mathbb{Q};\ y^2 = x^3 + ax + b\}.$$

It is well known that $E(\mathbb{Q})$ is a group with identity the point at infinity. Two elliptic curves

$$E : y^2 = x^3 + ax + b,\ \ E' : y^2 = x^3 + a'x + b'$$

are isomorphic over $\mathbb{Q}$ if there exists a nonzero $u \in \mathbb{Q}$ for which $a' = u^4 a$ and $b' = u^6 b$. Therefore, every elliptic curve over $\mathbb{Q}$ is isomorphic to an integral elliptic curve, i.e. $a, b \in \mathbb{Z}$. (for more details we refer to [24]). Now suppose that *E* is integral. For each prime *p*, we define $a_p = p + 1 - |E(\mathbb{F}_p)|$ where $\mathbb{F}_p$ is a finite field with *p* elements and $|E(\mathbb{F}_p)|$ is the order of

$$E(\mathbb{F}_p) = \text{point at infinity} \bigcup \{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p;\ y^2 = x^3 + ax + b\}.$$

When *p* divides Δ_E, one finds $a_p = 0, \pm 1$ where the exact value is determined according to the type of singularity of $E (\text{mod } p)$. The *L*-function of *E* is defined as

$$L_E(s) = \prod_{p | \Delta_E} \frac{1}{1 - a_p p^{-s}} \prod_{p \square \Delta_E} \frac{1}{1 - a_p p^{-s} + p^{1-2s}}.$$

The Hasse theorem states that for each prime p, a_p satisfies the following inequality

$$|a_p| \leq 2\sqrt{p}. \tag{1}$$

As a consequence of the Hasse bound (1), we find out that the Euler product $L_E(s)$ converges for all s with $\mathrm{Re}(s) > \frac{3}{2}$ and it is given by the following absolutely convergent series

$$L_E(s) = \sum_{n=1}^{\infty} a_n n^{-s}.$$

In summary, we can define the Fourier coefficients a_n, for all $n \geq 1$ using the following recurrence relations: $a_1 = 1$. If $p \geq 2$ is prime, we define

$$a_p = \begin{cases} p+1-|E(\mathbb{F}_p)| & \text{if } E \text{ has good reduction at } p, \\ 0 & \text{if } E \text{ has additive reduction at } p, \\ 1 & \text{if } E \text{ has split multiplicative reduction at } p, \\ -1 & \text{if } E \text{ has non-split multiplicative reduction at } p. \end{cases} \tag{2}$$

For $r \geq 2$

$$a_{p^r} = \begin{cases} a_p.a_{p^{r-1}} - p.a_{p^{r-2}} & \text{if } E/\mathbb{Q} \text{ has good reduction at } p, \\ (a_p)^r & \text{if } E/\mathbb{Q} \text{ has bad reduction at } p. \end{cases} \tag{3}$$

Finally, if $(m, n) = 1$, then we define

$$a_{mn} = a_m a_n. \tag{4}$$

Furthermore, the generalization of the Hasse bound (1) yields the following upper bound for $|a_n|$ due to Hasse-Weil [10]

$$|a_n| \leq d(n)\sqrt{n}, \tag{5}$$

where $d(n)$ denotes the number of positive integer divisors of n. It is known that $L_E(s) = L'_E(s)$ if and only if E is isogenous to E' [24].

The work of Schoof [21] says that for a given elliptic curve E over $\mathbb{Q}$, the coefficients a_p of $L_E(s)$ can be computed in polynomial time. Nevertheless, it is not easy to go in the other direction as explained below. Next, we recall the same notations and definitions as in [18]. Let B be a very large number and let μ, ν be fixed positive constants. Let $\mathfrak{L}^B$ be the set of all elliptic curves E over $\mathbb{Q}$ with $B \leq |\Delta_E| \leq 2B$. Let k and m be integers satisfying

$$(\log B)^{\mu} \leq k, m \leq (\log B)^{\nu}. \tag{6}$$

The conjecture of M. Anshel and D. Goldfeld [2] asserts that the projection map

$$F : \mathfrak{L}^B \to \mathbb{Q}^{k+1}$$
$$E \mapsto (a_m, a_{m+1}, ..., a_{m+k}) \tag{7}$$

is a one way function. More precisely, let C be a fixed positive constant with $C \geq \frac{\log B}{B}$ and B satisfying the inequality (6). Let denote by $\mathcal{L}^B$ the set of all elliptic curves E over $\mathbb{Q}$ with $B \leq |\Delta(E)| \leq 2B$. Let $\mathcal{L}_C^B$ be the subset of $\mathcal{L}^B$ consisting of all elliptic curves $E : y^2 = x^3 + ax + b$, $a, b \in \mathbb{Q}$ and $0 \leq |a|, |b| \leq CB$. In [18], it is shown the following theorem

Theorem 1: *Let the projection map*

$$H_1 : \mathcal{L}_C^B \to \mathbb{Q}^{k+1}$$
$$E \mapsto (a_m, a_{m+1}, ..., a_{m+k})$$

where $L_E(s) = \sum_{n=1}^{\infty} \frac{a_n}{n^s}$. *Then for any* $(c_m, c_{m+1}, ..., c_{m+k}) \in \mathbb{Q}^{k+1}$ *with* $|c_n| \leq \sqrt{n} d(n)$ *and for any* $\varepsilon > 0$, *there exists an algorithm which computes an elliptic curve E in* $\mathcal{L}_C^B$ *such that* $a_p = c_p$ *for all primes* $m \leq p \leq m + k$ *with a running time of order* $(CB)^{\frac{3}{2}+\varepsilon}$.

It is well known that the number of isomorphism classes of an elliptic curve E over $\mathbb{F}_p$ such that $a_p = c_p$ is at most $O(\sqrt{p} \log p (\log \log p)^2)$ [15] and there are also at most $\frac{p-1}{2}$ elliptic curves isomorphic to a given elliptic curve over $\mathbb{F}_p$. Therefore, the number of pairs $(a, b) \in \mathbb{F}_p^2$ for which the elliptic curve $E : y^2 = x^3 + ax + b$ satisfies $a_p = c_p$ is at most $O(p^{\frac{3}{2}+\varepsilon})$. In [8], the authors state an explicit algorithm which constructs such a curve and prove its complexity in time.

M. Anshel and D. Goldfeld in [2] tells us that the cryptographic security of the candidate one way function presented above is probably superior to other known candidate one way functions, such as those based on factoring or the computation of discrete logarithms and they described a few of numerous cryptographic applications based in this candidate like authentication, coin flipping by telephone and pseudo-random number generator. In another way, this candidate one way function inspired us to construct a new Vernam cipher which is described in the next section.

3. The Elliptic Curve *L*-Functions Integrated Encryption Scheme

In this section, we give a full description of our Elliptic Curve *L*-Functions Integrated Encryption Scheme (**ECLFIES**) for encryption and

decryption. First, as our cryptosystem is a Vernam cipher and therefore a symmetric key cryptosystem, then the secret key must be securely shared between the sender and the receiver. So, by following the RSA key exchange process, let's denote by $[C_1(K_{pb}), D_1(K_{pr})]$ and $[C_2(K_{pb}), D_2(K_{pr})]$ the public encryption and the private decryption functions of Alice and Bob respectively. In such a scenario, the secret keys are exchanged securely online by encrypting them with the intended recipient's public key.

Let Hash() denote a general hash function and let $(a_4, a_6) \in \mathbb{Z}^2$ represent the secret keys for the elliptic curve used in our cryptosystem. In the following, we restrict ourselves to integral elliptic curves of the form:

$$E: y^2 + xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6; a_2 \in \{-1, 0, 1\}, \text{and } a_3 \in \{0, 1\} \tag{8}$$

that is, we fix $a_1 = 1$, $a_2 \in \{-1, 0, 1\}$ and $a_3 \in \{0, 1\}$ in the Weierstrass equation. In this case, the elliptic curve E is said to be given by its minimal model which means that its discriminant $|\Delta_E|$ is minimal among all Weierstrass models for the same integral elliptic curve E. Actually, for elliptic curves over $\mathbb{Q}$, minimal models exist, and there is a unique reduced minimal model which satisfies the additional constraints $a_1, a_3 \in \{0, 1\}$, $a_2 \in \{-1, 0, 1\}$ [9].

In order to send an encrypted message to Bob, Alice must first request a token $A = C_1(K_{pb}, a_4)$ from Bob. That is, Bob pick a secret (random) integer a_4 and send it encrypted ($A = C_1(K_{pb}, a_4)$) using the public-key encryption system of Alice. Receiving the token A, Alice encrypt her message '$\mathfrak{msg}$' by the following algorithm (ECLFIES Encrypt):

Algorithm 1: ECLFIES Encrypt

Input: Message $\mathfrak{msg}$, Token $A = C_1(K_{pb}, a_4)$

$a_4 \leftarrow D_1(K_{pr}, A)$

$(a_6, a_2, a_3) \leftarrow$ FindSuitableEllipticCurve(a_4)

$X_0 \leftarrow$ a random integer of size 32 bits

step $\leftarrow$ a random integer of size 64 bits

$E \leftarrow$ EllipticCurve([1, a_2, a_3, a_4, a_6])

$C \leftarrow$ ECLFEncrypt ($\mathfrak{msg}$, E, X_0, step)

$H \leftarrow$ Hash (C)

Send: [C, H, C_2 (K_{pb}, (X_0, step, (a_6, a_2, a_3)))]

The X_0 is a random integer of size 32 bits, used to initialize the linear congruential generator:

$$X_{n+1} = (aX_n + c) \bmod 32,$$

where the multiplier a = 1664525 and the increment c = 1013904223 (see[19]). Actually, the considered sets of parameters provide a linear congruential generator (LCG) with a good performance with respect to the spectral test.

The FindSuitableEllipticCurve sub-function is based on trial-and-error routine, it takes a_4 as argument and returns three parameters: $a_6 \in \mathbb{Z}$, $a_2 \in \{-1,0,1\}$ and $a_3 \in \{0,1\}$ so that the density of zeros in the list of Fourier coefficients of the modular form $L_E(s)$ attached to the elliptic curve E defined by the equation of the form (8) is less than a fixed bound, let's say <1%. Of course, the chosen elliptic curve E must also verify the security condition: $|\Delta_E| > B$, i.e. the descriminant of E is very large. Since $(a_2, a_3) \in \{-1,0,1\} \times \{0,1\}$, Δ_E is one of the following:

$$\begin{aligned}
\Delta_E\{-1,0\} &= -64a_4^3 + 9a_4^2 - 216a_4a_6 - 432a_6^2 + 27a_6, \\
\Delta_E\{-1,1\} &= -64a_4^3 - 87a_4^2 - 216a_4a_6 - 432a_6^2 - 93a_4 - 297a_6 - 53, \\
\Delta_E\{0,0\} &= -64a_4^3 + a_4^2 + 72a_4a_6 - 432a_6^2 - a_6, \\
\Delta_E\{0,1\} &= -64a_4^3 - 95a_4^2 + 72a_4a_6 - 432a_6^2 - a_6 - 29a_4 - 181a_6 - 26, \\
\Delta_E\{1,0\} &= -64a_4^3 + 25a_4^2 + 360a_4a_6 - 432a_6^2 - 125a_6, \\
\Delta_E\{1,1\} &= -64a_4^3 - 71a_4^2 + 360a_4a_6 - 432a_6^2 + 67a_4 - 161a_6 - 15.
\end{aligned}$$

Thus, according to the previous equations of $\Delta_E\ a_2, a_3$, to ensure the discriminant condition relating to **ECLFIES** security, it is more convenient to take $a_4 > 0$ and $a_6 < 0$ both of them having size approximately 32 bits. In summary, to find a correct elliptic curve for encryption, this function take a large positive integer a_4 and tries to find integers $a_6 \in]-2^{32}, -2^{30}[$, $a_2 \in \{-1,0,1\}$ and $a_3 \in \{0,1\}$ suitable for such purpose. The FindSuitableEllipticCurve is described as follows:

Algorithm 2: FindSuitableEllipticCurve

Input: large integer $a_4 > 0$

Output: (a_6, a_2, a_3)

loop

$a_6 \leftarrow$ random(]-2^{32}, -2^{30}[)

for a_2 in (–1, 0, 1) **do**
for a_3 in (0, 1) **do**
$E \leftarrow$ EllipticCurve([1, a_2, a_3, a_4, a_6])
if ZeroDensity(E) < 1% AND $|\Delta_E| > B$ **then**
return (a_4, a_2, a_3)
end if
end for
end for
end loop

In the next section, we discuss this generate-and-check approach and show that it is actually an efficient heuristic and very practical for finding a suitable candidate elliptic curve for encryption.

Now, the ECLFEncrypt(msg, E, X_0, step) is our main Vernam encrypt function. It takes 4 parameters:

– msg: The message to be encrypted.
– E: The elliptic curve built by FindSuitableEllipticCurve function
– X_0 and step

and return the ciphe-text C. Of course, the function does not require padding because it effectively uses a block cipher as a stream cipher.

The procedure of encryption consists of the following steps:

1. The message msg is divided into a sequence S_{msg} of k blocks $\{m_i; i = 1..k\}$ where $|m_i| = 32$ bits ($i = 1..k-1$) and $|m_k| \leq 32$ bits.
2. We insert the two 32-bits integers a_4 and a_6 into the S_{msg} sequence at position t = step mod (k – 1) and t + 1 respectively. So S_{msg} is now a sequence of k + 2 blocks:
$$S_{msg} = (m_1, m_2, ..., m_{t-1}, a_4, a_6, m_t, ..., m_k).$$
3. Let $\mathcal{K}_E$ be the sequence of k + 2 Fourier coefficient of the modular form attached to the elliptic curve E:
$$\mathcal{K}_E = (e_m, e_{m+1}, ..., e_{m+k+1})$$
where the index m (the first index in $\mathcal{K}_E$ sequence) is given by: m = 55 x 10^{14} + step mod (45 x 10^{14}) (see Paragraph 4 for the choice of m). Here we denote by e_i the Fourier coefficients of the L-function of E to avoid any ambiguity with the integers a_2, a_3, a_4 and a_6.

4. Let $\mathcal{K}_{LCG}$ be the sequence of $k+2$ pseudo-random numbers calculated by our linear congruential generator:

$$\mathcal{K}_{LCG} = (x_1, x_2, ..., x_{k+2}).$$

5. The result cipher-text is the sequence $C = (c_1, c_2, \ldots, c_{t-1}, A, B, c_t, \ldots, c_k)$ defined by:

$$C[i] = S_{msg}[i] \oplus \mathcal{K}_E[i] \oplus \mathcal{K}_{LCG}[i]; (i = 1..k+2).$$

The encryption procedure can be sketched by the following chart:

The decryption algorithm begins by recovering the secret-key of encryption which consists of the elliptic curve E defined by the coefficients (a_2, a_3, a_4, a_6) and the integers X_0 and step. Then, it verify the integrity of the received cipher-text by checking it's hash value. If it is correct, we start the main decryption process: ECLFDecrypt which is the same as the XORed encryption algorithm. After XORed iterations end, we get the following sequence of blocks:

$$(m'_1, m'_2, ..., m'_{t-1}, a'_4, a'_6, m'_t, ..., m'_k).$$

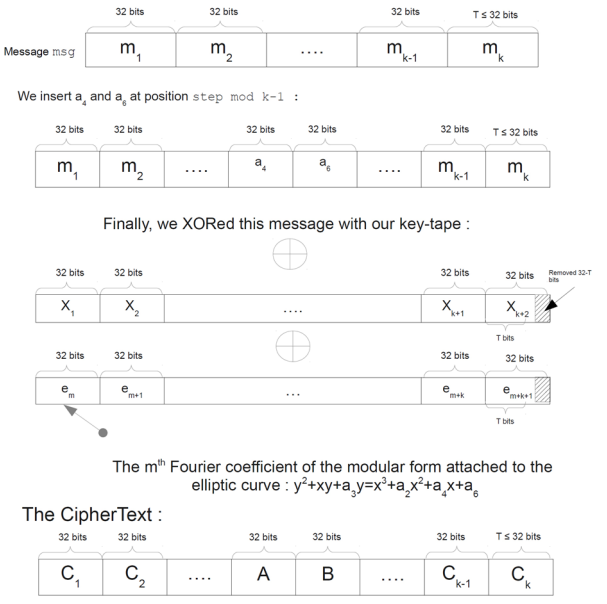

Figure 1
ECLFEncrypt Function

Then, it takes off the two coefficients a'_4 and a'_6 and returns the triplet (a'_4, a'_6, msg') where $msg' = (m'_1, m'_2, ..., m'_{t-1}, m'_t, ..., m'_k)$. Now, if $a'_4 = a_4$ and $a'_6 = a_6$ then the decryption is successful otherwise we are against a fake or non-authentic message.

We summarize the decryption process by the following algorithm ECLFIES Decrypt:

Algorithm 3 : ECLFIES Decrypt

Input: [C, H, $C_2(K_{pb}, (X_0, \text{step}, (a_6, a_2, a_3)))$]

$(X_0, \text{step}, (a_6, a_2, a_3)) \leftarrow D_2(K_{pr}, C_2(K_{pb}, (\text{salt}, \text{step}, (a_6, a_2, a_3))))$

$E \leftarrow$ EllipticCurve ([1, a_2, a_3, a_4, a_6)])

if Hash(C) $\neq H$ **then**

return INVALID

else {No integrity errors, then we decrypt C using the key $\mathcal{K} = (E, X_0, \text{step})$}

$(a'_4, a'_6, \mathfrak{msg}') \leftarrow$ ECLFDecrypt (C, E, X_0, step)

if $a'_4 = a_4$ and $a'_6 = a_6$ **then**

return $\mathfrak{msg}'$ {a successful decryption}

else

return non authentic or fake message

end if

end if

Concerning the digital signature in our **ECLFIES** cryptosystem, we can see that it is less expensive than the standard signature-and-encryption schemes in terms of the length of the signature (i.e. communication overhead) and the modular exponentiation (i.e. computational cost).

Actually, let consider the **ECLFIES** keys exchange algorithms to be both the RSA public-key cryptosystem with modulus n_1 and n_2, and let p, q be the keys of the NIST digital signature algorithm with $512 \leq |p| \leq 1024$ bits and $|q| = 160$ bits. Then, we summarize the comparison with standard signature-then-encryption schemes and the signcryption scheme[27] in Table 1:

Thus, we get: cost(Signcryption) < cost(**ECLFIES** sig)< cost(RSA, DL based sig-then-enc).

Table 1
Signature's cost comparison table

Schemes \ Cost	**Comp. cost**	**Comm. overhead**
RSA based sig-then-enc	$2 + 2 = 4$	$\|n_1\| + \|n_2\|$
DL based Schnorr sig + ElGamal enc	$3 + 2.17 \simeq 6$	$\|\text{hash}\| + \|p\| + \|q\|$
ECLFIES sig	$2 + 2 = 4$	$\|\text{hash}\| + \|n_1\| + 64$ bits because $\|X_0$, step, $(a_6, a_2, a_3)\| = 144 < \|n_1\|$ and $\|C\| = \|\text{msg}\| + 64$ bits
DL based Signcryption	$1 + 1.17 \simeq 3$	$\|\text{hash}\| + \|q\|$

Finally, we suggest the short name **ECLFIES** in conjunction with the Elliptic Curve Integrated Encryption Scheme (**ECIES**) proposed by Vector Shoup in 2001, see [23] and [1].

4. Parameters choices and algorithms analysis

In standard one-time pad, each character of 8 bits in a given plain-text is encrypted by combination with the corresponding character from the pad using modular addition. However, in our algorithm, we take a 32-bit block (4 characters) of plain-text as input, and output the corresponding 32-bit block of cipher-text. Thus, it is necessary to ensure that the Fourier coefficients of the modular form attached to the elliptic curve fit into a 32-bit signed integer.

Let's denote $a(n)$ the n-th Fourier coefficient of the Hasse-Weil L-series of the elliptic curves E. By the Hasse-Weil bound (5), one can find a theoretical upper bound so that $a(n)$ does not exceed the 32 bits limit. The first integer n for which the bound $\lfloor\sqrt{n}\rfloor \times d(n)$ exceeds 32 bits size is $n = 562194032400 \simeq 10^{11.75}$. Hence, in theory, our first index in the Fourier coefficients key-tape vector can be chosen in the neighborhood of 10^{12}. But, in practice, Fourier coefficients around the position 10^{12} are too small than 32 bits, namely the left-most significant bits are always set at 0 and therefore, not suitable for encryption purpose. To overcome this security threat, we made many experiments to find a good neighborhood starting position $\mathcal{V}$, where the coefficients $\{|a(n)|, n \in \mathcal{V}\}$ are not too small and do not exceed 32-bit size. Our experiments and heuristics suggest that a good neighborhood is around 10^{16}. Thus, we choose to take the first index in the Fourier coefficients key-tape vector as $m = 55 \times 10^{14} + [\text{step mod } (45 \times 10^{14})]$

$< (55 + 45)\ 10^{14} = 10^{16}$. Since $45 \times 10^{14} \simeq 52$ bits, then we can take "step" to be a random integer of 64-bit size.

4.1 *The Linear Congruential Generator*

Classical Pseudo-Random Number Generator (PRNG) like Linear Congruential Generators (LCGs), first proposed by Lehmer in 1949 [14], are easy to implement, produce good empirical results and require few energy-time consumption. However, generators based on linear congruencies (including LCGs) are not cryptographically secure and therefore not suitable for cryptographic applications. Actually, if just one value X_i of the LCG's sequence is known, then the whole LCG can therefore be calculated. Subsequently, a based dictionary landing attack will be very efficient. For instance, let's have a Vernam-encrypted email of 100 characters-size using an LCG sequence as a key-tape. Then, we guess that the word 'the' occurs in the encrypted email. Moreover, we can suppose that all delimiters of the plain-text are preserved for the receiver's pleasure to read the email. Let's suppose that our LCG is a 32-bit (i.e. four characters) generator. Then, there exist a segment $c_1c_2c_3c_4$ of four characters in the cipher-text obtained by XORing the plain-text word 'the' by the value X_i at some unknown position i. So, we may expect to find some ciphered segment $c_1c_2c_3c_4$ having one of the following form:

$$c_1c_2c_3c_4 = \begin{cases} \square\,t\,h\,e \\ t\,h\,e\,\square \\ \star\,\square\,t\,h \\ h\,e\,\square\,\star \end{cases} \oplus X_i$$

where $\star$ is any possible character (let's say 26 possibilities).

Hence, after trying the different possible positions of i ($54 \times (100/4) = 1350$ possible positions), the word will be landed in the exact position and consequently the value of X_i is recovered and so is the whole key-tape. Of course, we may also try other words like 'when', 'kill', ..etc. Therefore, using today's computer power, the attacker can search an entire dictionary and so the key-tape can be broken easily.

Many other attacks on LCG can be found. For instance, Boyar [6] and Krawczyk [13] showed that without prior knowledge of LCG constants a, c and modulus m, one can predict the generator value within a polynomial bounded number of earlier values. Furthermore, Frieze et al. [12] showed that with the knowledge of a and modulus m, the truncated values generated by an LCG can be successfully reconstructed with a

certain probability depending on the number of given significant bits. In [25], Stern showed that the latter construction can be done without the knowledge of a and m as long as more information in the form of number of values and bits of values can be had.

In order to make LCGs suitable for cryptographic applications, recent techniques try to break the linear structure of the LCG sequence. For example, the design proposed in [3] transforms the LCG sequence $(X_n)_n$ to another sequence $(Y_n)_n$, where $Y_i = X_i \oplus T(X_{i+1})$ and T is an operator which reverses the order of the bits, i.e. if $X = (b_0, b_1, \ldots, b_{n-1})_2 = b_0, b_1 2 + .. + b_{n-1}2^{n-1}$ in bit-representation, then $T(X) = (b_{n-1}, b_{n-2}, .., b_0)_2$. Therefore, without using techniques like RSA or Blum-Blum-Shub (which are too slow in practice), the previous algorithm makes the output sequence fast and essentially non-linear. However, we propose here a simple modelisation and an arising attack. Let's suppose, like before that an attacker knows two internal states, say Y_0 and Y_1. So, we have $Y_0 \oplus T(Y_1) = X_0 \oplus X_2 = D$.

But, $X_2 = aX_1 + c \bmod m = a^2X_0 + (ac + c) \bmod m$. Since, $X_2 = X_0 \oplus D$, we get the equation: $\lambda X_0 + B = X_0 \oplus D \bmod m$, where $\lambda = a^2 \bmod m$ and $B = ac + c \bmod m$.

Let's use the authors parameter [3]: $m = 2^{31} - 1$ and let $N = 31$.

Let's denote:

$$X_0 = (a_0, a_1, .., a_{N-1})_2,\ B = (b_0, b_1, .., b_{N-1})_2$$
$$\lambda = (\lambda_0, \lambda_1, .., \lambda_{N-1})_2,\ D = (d_0, d_1, .., d_{N-1})_2$$

and $Z = (z_0, z_1, .., z_{N-1})_2 = X_0 \oplus D$. Then, the attacker has to solve the equation:

$$\lambda X_0 + B = Z \bmod m. \tag{9}$$

Since $2^N = 1$ in the finite field $\mathbb{Z}_m$, the convolution product of the two polynomials $\lambda \times X_0 = (\lambda_0 + \lambda_1 2 + .. + \lambda_{N-1}2^{n-1}) \times (a_0 + a_1 2 + .. + a_{N-1}2^{n-1})$ mod m can be represented by the matrix product of the square circulant matrix of λ with the vector representation of X_0:

$$\lambda \times X_0 \leftrightarrows C_\lambda \times X_0] = \begin{bmatrix} \lambda_0 & \lambda_{N-1} & \cdots & \lambda_1 \\ \lambda_1 & \lambda_0 & \cdots & \lambda_2 \\ \vdots & & \ddots & \\ \lambda_{N-1} & \lambda_{N-2} & \cdots & \lambda_0 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{N-1} \end{bmatrix}$$

Thus, we can interpret equation (9) by:

$$\begin{aligned}(C_\lambda \mid -I_N)\times X_0 \mid Z] &= -B \bmod m \\ AY &= -B \bmod m,\end{aligned} \tag{10}$$

where $A = C_\lambda \mid -I_N$ is an $N \times 2N$ matrice over $\mathbb{Z}_m$ and Y is a $2N$ column vector (the concatenation of the two column vectors X_0 and Z).

At this point, we have to solve a system of N linear equations with $2N$ unknowns $\{a_0, .., a_{N-1}, z_0, .., z_{N-1}\}$ over the finite field $\mathbb{Z}_m$. Of course the two extra conditions: $a_i \in \{0,1\}$ and $z_i = a_i \oplus d_i$ lead to solve certain integer lattice problem. If the increment $c \neq 0$, then finding X_0 is actually related to solving the closest vector problem (CVP). Many attacks, including lattice reduction attacks can be successful for small N in reasonable time see [16], [4]. If $c = 0$, then we try to find a shortest non-zero vector (in the l_∞ norm) of that lattice (i.e. the SVP problem). In this case, since $z_i = a_i \oplus d_i$, the k-list/generalized-birthday attack suggested by Wagner [26] can be very successful. In summary, since the coefficients of the matrix A are only {–1, 0, 1}, then in order to get 100 bits of security, one needs to set $2N > 1000$, so N must be greater than 500, see [17].

In such case, this design may lose a lot of speed because the modulus m is greater than 2^{500}. Therefore, the multiple-precision arithmetic is mandatory and the invert operator T will be slower than its 32-bit similar because the rotate operator is only designed for 8, 16, 32 or 64 bits in length (cpu registers restriction) and a big key (the seed) must be used.

In fact, even recent advances in random number generation like the construction of Multiple Recursive Generator(MRG) with large orders, Fast Multiple Recursive Generator (FMRG) and DX (system of multiple recursive generators proposed by Deng and Xu [11]) are not cryptographically strong as the linear system can be predicated using a system of unique k equations.

The **ECLFIES** presents a new design, very different from others already broken cryptosystems based on LCGs pseudo-random number generator. Actually, our LCG sequence is mixed with the Fourier coefficients of the L-series of an elliptic curve. This mixing yields a final output sequence that resists to linear cryptanalysis, very difficult to predict and enough strong for cryptographic purpose.

4.2 *The FindSuitableEllipticCurve Function*

Suppose that the sequence of Fourier coefficients $(a_n)_n$ contains many zeros, let's say 20%. In such scenario, 20% of our plain-text is only ciphered with LCG's coefficients. So, some possible frequent "english" words

would be only encrypted by the LCG's coefficients. Thus, by a dictionary attack using the previous landing technique, one may get chance to find some correct coefficient of the LCG sequence and consequently retrieve the whole LCG. Then, 20% of the ciphered text can be revealed. Thus, it is crucial to choose an elliptic curve with a low density of zeros. This means we should avoid elliptic curves with many "small" primes p such as $a_p = 0$. Indeed, let $\{p_1, p_2, .., p_r\}$ the list of primes $\subseteq 1,2,..,N] = \bar{N}$ where $a_p = 0$. Then, using equations (2, 3, 4) and the principle of inclusion-exclusion, we deduce that the density of zeros in the list $\{a_1, a_2, \ldots, a_N\}$ verifies the inequality:

$$0 - density[\bar{N}] < \frac{\sum_{\{p_{i_1},..,p_{i_k}\} \subseteq \{p_1,..,p_r\}} (-1)^{k+1} \left\lfloor \frac{N}{p_{i_1} p_{i_2} ... p_{i_k}} \right\rfloor}{N}.$$

Since our cryptographic sequence starts in a neighborhood of 10^{16}, denoted by $\mathcal{V}(10^{16})$, then in order to get a good approximation of the "ZeroDensity(E)" function in $\mathcal{V}(10^{16})$, we take $N \sim 10^{12}$ and we consider that most of primes p such as $a_p = 0$ are big (e.g. $p \geq 1000$), that is our elliptic curve E has no additive reduction at small primes. So, for the most of combinations $\{p_{i_1},..,p_{i_k}\} \subseteq \{p_1,..,p_r\}$ we get $\left\lfloor \frac{N}{p_{i_1} p_{i_2} ... p_{i_k}} \right\rfloor = 0$. Therefore, it is reasonable to suppose that:

$$0 - \text{density}[\mathcal{V}(10^{16})] \simeq 0 - \text{density}[\overline{10^4}].$$

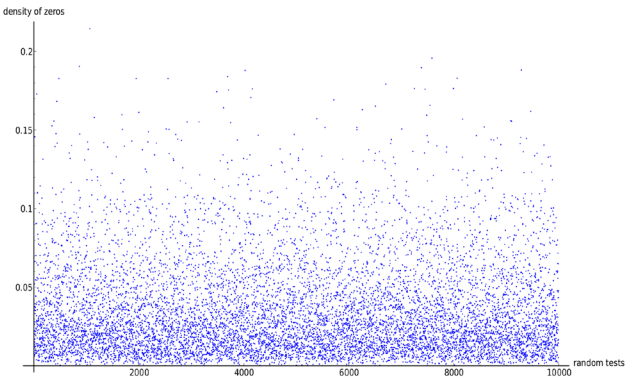

Figure 2

Density of zeros for the best minimal models for random a_4 and a_6.

Then, the function "ZeroDensity(E)" can be heuristically computed within the range of 10^4 instead of $\mathcal{V}(10^{16})$. By our simulation, even the range [1..5000] provides a good approximation. Of course, one may expect that the function "FindSuitableEllipticCurve" needs many loop iterations to find a "good" elliptic curve with low zeros density and consequently it is too slow to be used in practice. However, the following experimental analysis (see. Figure 2) shows that the probability of finding a such elliptic curve is good when it is given by a minimal Weierstrass equation of the form (8).

Technically, a simple and even a fast way to calculate the *0-density* $[\overline{m}]$ is to use the function **ellan (*E*, *m*)** in **PARI/GP**[1] which returns an array with the first m coefficients a_k, for $k = 1, .., m$.

5. Security Analysis

In **ECLFIES** design, the private cryptographic key consists of the 6-tuple (a_2, a_3, a_4, a_6, X_0, step) then, from a brute-force attack perspective, our cryptosystem offer a $(6*32^5)$-bits encryption security which is largely sufficient for the NIST recommendations. Nevertheless, the main goal is to satisfy the Shannon security requirements.

Above all, the security depends on the random aspect of key-tape's sequence. Of course, "proving" the randomness of the output sequence is a very difficult task. However, the pragmatic approach is to take many sequences generated by the given generator and subject them to a battery of statistical tests. As the sequences pass mostly the tests, the confidence in the randomness of the numbers increases and so does the confidence in the generator. To test our **ECLFIES** sequences, we use the NIST statistical test suite for random and pseudo-random number generators, which is the state of the art in tests for randomness, see [20]. In Table 2, we give the results of our **ECLFIES** output for 10^6 binary digits. In this test, the LCG is defined by $a = 1664525$, $c = 1013904223$, $m = 2^{32}$ and the seed $X_0 = 615818293$. The elliptic curve is $E : y^2 + xy = x^3 + x^2 + 1981703639x - 326744285$ and step = 46541354646. The decision rule is based on the computed P-value: if the P-value is < 0.01, then we conclude that the sequence is non-random. Otherwise, we conclude that the sequence is random.

Here are also some further statistical results for our same 10^6 binary digits sequence:

– Entropy = 7.998438 bits per byte.

Table 2
The NIST statistical tests for 10^6-bits ECLFIES sequence

Test	*P*-value	Result
The Frequency (Monobit) Test	0.2316	SUCCESS
Frequency Test within a Block	0.9996	SUCCESS
The Runs Test	0.1430	SUCCESS
Tests for the Longest-Run-of-Ones in a Block	0.5859	SUCCESS
The Binary Matrix Rank Test	0.7739	SUCCESS
The Discrete Fourier Transform (Spectral) Test	0.0000	**FAILURE**
The Non-overlapping Template Matching Test	[0.9413; 0.2767; …]	SUCCESS
The Overlapping Template Matching Test	0.2126	SUCCESS
Maurer's "Universal Statistical" Test	0.4275	SUCCESS
The Linear Complexity Test	0.9278	SUCCESS
The Serial Test	[0.5658; 0.4614]	SUCCESS
The Approximate Entropy Test	0.3946	SUCCESS
The Cumulative Sums (Cusums) Test	0.1828	SUCCESS
The Random Excursions Test	0.377 0.410 0.626 0.333 0.530 0.466 0.363 0.225	SUCCESS
The Random Excursions Variant Test	[0.5578; 0.5598; …]	SUCCESS

- Chi square (χ^2) distribution for 125000 samples (bytes) is 271.54, and randomly would exceed this value 22.77% of the times (so this distribution is considered genuine random).
- Arithmetic mean value of data bytes is 127.5416 (127.5 = random).
- Monte Carlo value for π is 3.142898286 (error 0.04%).
- Serial correlation coefficient is – 0.000663 (totally uncorrelated = 0.0).

Secondly, concerning the one-time key-tape condition, we can see that the design itself is responsible for preventing the key-tape to be used more than once. Actually, it is worth noting that most of Internet decryption attack methods are based on an improper use of protocols and encryption keys, not the cryptography itself.

First, in the **ECLFIES** algorithm, there is no bad encryption keys to be avoided, except the public methods for exchanging the secret key online. Secondly, if we suppose that Bob is always using the same secret key a_4,

then the risk that Alice methods choose the same keys $\{E(1, a_2, a_3, a_4, a_6); X_0; \text{step}\}$ and consequently to produce the same key-tape is very minimal. Thus, the whole secret key will always be a new random one.

Finally, regarding the length of the key-tape, which must be as long as the message to be encrypted, our LCG produces a sequence with maximum period length and the sequence of the Fourier coefficients is not a periodic one. Hence, for messages of reasonable size, our key-tape meets the length condition.

It is important to note that most of Fourier coefficients are increasing in size. Therefore, it is necessary to choose a range in the corresponding L-series where there is no overflow of 32-bit integers. As said before, if we start our sequence in the neighborhood of 10^{16}, then by our experimental tests, we always obtain a reasonable range of 32-bit coefficients which allows us to encrypt messages of hundreds of megabytes.

5.1 *The Known Plain-text Attack*

The known plain-text attack is an attack model for cryptanalysis where the attacker has samples of both the plain-text and its encrypted version. In this case, the attacker attempts to determine a key by analyzing the relationship between a decrypted message and the encrypted output.

In practice, we suppose that the attacker may know only a "small" portion of the encrypted message, for example, a message header. It should be noted that obtaining a known portion of the encrypted message is much easier than it appears. Actually, most of modern programs, have a fixed header on all their files so that when the program goes to open it, the program knows for certain the file type, format, parameters, etc. Therefore, after some time by analyzing the relationship between the encrypted and the decrypted portion the attacker determines the key. Hence, the entire message is compromised.

Classical ciphers are typically vulnerable to known plain-text attack. This includes of course stream ciphers based on classical random number generators such as LFSR (linear feedback shift register), LCG, MRG and DX.

In our case, let's suppose that the attacker knows the LCG sequence and knows a partial consecutive sequence $[a_j, a_{j+1}, \ldots, a_{j+k}]$ of the Fourier coefficients of the modular form attached to the elliptic curve E. So, one might ask the question whether it is possible for an attacker to calculate the next Fourier coefficients $[a_{j+k+1}, a_{j+k+2}, .., a_{j+s}]$. To our knowledge such an attack doesn't exist, and we expect that it is very difficult to predict the sequence. Actually, the recurrence relations (2), (3) and (4) are not helpful

in decomposition techniques for integers, because j (the starting point of the known portion) is too large $(j \in \mathcal{V}(10^{16}))$ therefore, getting the value of a_p for small primes p from the values $\{a_t\}_{t=j..j+k}$ is not possible. In the other hand, according to the conjecture of M. Anshel and D. Goldfeld, finding the right elliptic curve E under the knowledge of the Fourier sequence is not possible. Even if we assume that the attacker succeeds to find the right elliptic curve E, it remains hard to find the right sequence $\mathcal{K}_E = (e_m, e_{m+1}, ..., e_{m+k+1})$ because the starting point is $m = 55 \times 10^{14} +$ [step mod (45×10^{14})] is not known for the attacker as it is hidden by the secret key "step" which offers, on its own, a 64-bits security level. Therefore, looking at all these restrictions it is fair to assume that our **ECLFIES** is resistant to known plain-text attacks.

5.2 *Non-Deniable Encryption*

A deniable encryption is when a possible scenario can create a fake key able to decrypt the ciphered text to a random sensible fake plain-text. To simplify the scenario in our case, let us remove the LCG sequence from our key-tape. Let $\{m'_1, m'_2, ..., m'_k\}$ be the blocks of the fake plain-text to be created and let $\{c_1, c_2, ..., c_{\lambda-1}, A, B, c_\lambda, .., c_k\}$ be the blocks of the ciphered text (see Figure 1). Thus, to create such "fake" blocks we must be able to find a "fake" elliptic curve $E' : y^2 + xy + a'_3 y = x^3 + a'_2 x^2 + a'_4 x + a'_6$ so that the Fourier coefficients of the modular form $L_{E'}(s)$ attached to the "fake" elliptic curve E' sequence $[e'_m, e'_{m+1}, .., e'_{m+k+1}]$ fulfills the following conditions:

$$\begin{cases} e'_{m+i} = m'_{i+1} \oplus c_{i+1}; \forall i = 0...\lambda-2, \\ e'_{m+\lambda-1} = a'_4 \oplus A, \\ e'_{m+\lambda} = a'_6 \oplus B, \\ e'_{m+i} = m'_{i+1} \oplus ci-1; \forall i = \lambda+1...k+1, \end{cases}$$

where $\lambda = step \bmod (k-1)$.

However, as we have already seen in the second section, we deduce that the class of elliptic curves satisfying $a_p = e'_p$ would require $\theta = O((CB)^{\frac{3}{2}+\varepsilon})$. Besides, any potential solution E' from this class must also satisfy the two additional constraints: $a'_4 = e'_{m+\lambda-1} \oplus A$ and $a'_6 = e'_{m+\lambda} \oplus B$. For this reason, it is reasonable to conclude that our encryption system is a non-deniable one.

6. Conclusion and Remarks

One-time pad cipher had been always considered not practical for cryptographic applications due to the key-tape length restriction and management difficulty and pitfalls. Moreover, cryptanalysis methods show that classical generators like LCGs that generate pseudo-random linear sequences are not suitable for cryptographic purposes. Indeed, it is shown that a sequence produced by LCG is not only predictable, but should be used with extreme caution even within cryptographic applications that would appear to protect this sequence, see [5]. On the other hand, cryptographically secure generators like BBS and RSA are not good enough for most classical and daily routine tasks like email encryption because they are very slow in execution.

Our integrated encryption scheme tries to provide an alternative attitude toward the use of pseudo-random number generator in Vernam ciphers. Our encryption will be considered secure if the projection map (7) is one-way and the L-series sequence $(a_n)_{n\geq m}$ is difficult to predict when m is too large. The high probability of finding an elliptic curve with low zero density of the L-series coefficients enables us to make the implementation of the cryptosystem with great efficiency and speed.

Concerning the speed, our experimental tests have shown that our generator is approximately 800% faster than the BBS generator. Optimized implementations with a predefined target architecture, avoiding multiple precision arithmetic are also possible in our design. Hence, we can effectively gain more speed and estimate that it would be more than 1500% faster than secure BBS and RSA generators. We note that in the case of BBS and RSA generators the modulus $M = pq$ should be very large to avoid factorization attacks. Thus, multiple precision arithmetic is mandatory in their cases.

Finally, we believe that we still can improve quite a lot our design in order to be more effective even for traditional applications of hashing, especially for big size files. So, further studies and technical aspects should be revisited and improved.

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] Abdalla, M., Bellare, M., Rogaway, P.: The oracle Diffie-Hellman assumptions and an analysis of DHIES. In: Topics in Cryptology-

CT-RSA 2001, *Lecture Notes in Computer Science*, vol. 2020, pp. 143–158 (2001).

[2] Anshel, M., Goldfeld, D.: Zeta functions, one-way functions, and pseudorandom number generators. *Duke Mathematical Journal* 88(2), 371–390 (1997).

[3] Anyanwu, M.N., Deng, L.Y., Dasgupta, D.: Design of cryptographically strong generator by transforming linearly generated sequences. *International Journal of Computer Science and Security* 3(3), 186–200 (2009).

[4] Becker, A., Gama, N., Joux, A.: Solving shortest and closest vector problems: The decomposition approach. IACR Cryptology ePrint Archive (2013). Available at http://eprint.iacr.org/2013/685.

[5] Bellare, M., Goldwasser, S., Micciancio, D.: Pseudo-random generators within cryptographic applications: the DSS case. In: Advances in Cryptology, *Lecture Notes in Computer Science*, vol. 1294, pp. 277–291. Springer (1997).

[6] Boyar, J.: Inferring sequences produced by pseudo-random number generators. *Journal of the ACM* 36(1), 129–141 (1989).

[7] Brown, D., Antipa, A., Campagna, M., Struik, R.: Ecoh: the elliptic curve only hash. Submission to NIST (2008).

[8] Canetti, R., Dwork, C., Naor, M., Ostrovsky, R.: Deniable encryption. In: CRYPTO, *Lecture Notes in Computer Science*, vol. 1294, pp. 90–104. Springer (1997).

[9] Cremona, J.E.: Algorithms for modular elliptic curves, second edn. Cambridge University Press, Cambridge (1997).

[10] Dale, H.: Elliptic Curves, *Graduate Texts in Mathematics*, vol. 111. Springer (1987).

[11] Deng, L.Y., Xu, H.: A system of high-dimensional, efficient, long-cycle and portable uniform random number generators. ACM Trans. Model. Comput. Simul. 13(4), 299–309 (2003).

[12] Frieze, A.M., Hastad, J., Kannan, R., Lagarias, J.C., Shamir, A.: Reconstructing truncated integer variables satisfying linear congruences. *SIAM J. Comput.* 17(2), 262–280 (1988).

[13] Krawczyk, H.: How to predict congruential generators. *J. Algorithms* 13(4), 527–545 (1992).

[14] Lehmer, D.H.: Mathematical methods in large-scale computing units. In: Proc. 2nd Symp. on Large-Scale Digital Calculating Machinery, pp. 141–146. Harvard University Press (1951).

[15] Lenstra, H.: Factoring integers with elliptic curves. *Annals of Mathematics* 126(2), 649–673 (1987).

[16] Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: a cryptographic perspective, *The Kluwer International Series in Engineering and Computer Science*, vol. 671. Kluwer Academic Publishers (2002).

[17] Micciancio, D., Regev, O.: Lattice-based cryptography. In: *Post-Quantum Cryptography*, pp. 147–191. Springer (2009)19.

[18] Omar, S., Ouni, R., Bouanani, S.: Hashing with elliptic curve L-functions. In: F. Ozbudak, F. Rodriguez-Henriquez (eds.) WAIFI, *Lecture Notes in Computer Science*, vol. 7369, pp.196–207. Springer (2012).

[19] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes 3rd Edition: The Art of Scientific Computing, 3 edn. Cambridge University Press (2007).

[20] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., Vo, S.: A statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic applications (2010). Available at http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf.

[21] Schoof, R.: Elliptic curves over finite fields and the computation of square roots mod p. Mathematics of Computation 44, 483–494 (1985).

[22] Shannon, C.: Communication theory of secrecy systems. *Bell Systems Techn. Journal* 28, 656–719 (1949).

[23] Shoup, V.: A proposal for an iso standard for public key encryption. IACR Cryptology ePrint Archive (2001). Available at http://eprint.iacr.org/2001/112.

[24] Silverman, J.H.: The Arithmetic of Elliptic Curves, *Graduate Texts in Mathematics*, vol. 106. Springer (1986).

[25] Stern, J.: Secret linear congruential generators are not cryptographically secure. In: 28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987, pp. 421–426 (1987).

[26] Wagner, D.: A generalized birthday problem. In: Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology, CRYPTO, pp. 288–303. Springer (2002).

[27] Zheng, Y.: Digital signcryption or how to achieve cost(signature & encryption) cost(signature) + cost(encryption). In: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO, pp. 165–179. Springer (1997).

Received March, 2022

Revised May, 2022