

The REESSE1+ Public-key Cryptosystem

— A Multi-hard Public-key Cryptosystem

Shenghui Su¹, and Shuwang Lü²

¹School of Information Engineering, University of Science & Technology Beijing
Beijing 100083, P. R. China
sheenway@126.com

²School of Graduate, Chinese Academy of Sciences
Beijing 100039, P. R. China.

Abstract: This paper gives the definition of a coprime sequence and the concept of the lever function, describes the five algorithms and six characteristics of the REESSE1+ public-key cryptosystem based on three new hardnesses: the modular subset product problem, the multivariate arrangement problem, and the super logarithm problem in a prime field, shows the correctness of the decryption algorithm, and infers that the probability that a plaintext solution is not unique is nearly zero. The authors analyze the security of REESSE1+ against recovering a related plaintext from a ciphertext, extracting a related private key from a public key or a signature, and faking a digital signature via a public key or a known signature with a public key, discuss the super logarithm problem, and believe that the security of REESSE1+ is at least equal to the time complexity of $O(2^n)$ at present. At last, the paper expounds the idea of optimizing REESSE1+ through binary compact sequences.

Keywords: Multi-hard public-key cryptosystem, Coprime sequence, Security, Lever function, Super logarithm problem, Double congruence theorem.

1 Introduction

The trapdoor functions for RSA^[1] and ElGamal^[2] public-key cryptosystems^[3] are computationally one-way^[4]. Along with the elevation of computer speeds, such one-wayness will be weakened. Hence, to enhance the one-wayness of a trapdoor function, sometimes a public-key cryptosystem is transplanted to a complex algebraic system from a simple one. For example, the ElGamal analogue in an elliptic curve group, namely the ECC cryptosystem, is more one-way or more secure than ElGamal itself^[5]. However, this method is not suitable for all the existing cryptosystems.

In some public-key cryptosystems, trapdoor functions can prevent a related plaintext from being recovered from a ciphertext, but cannot prevent a related private key from being extracted from a public key. For instance, in the MH knapsack cryptosystem^[6], the subset sum problem which serves as a trapdoor function can not preclude a private key from being inferred through the Shamir method^[7].

Different from RSA, ElGamal and MH, REESSE1+ brings 3 independent variables into the general key transform. Its security is not based on classical hardnesses: the subset sum problem, the integer factorization problem and the discrete logarithm problem, but on three new hardnesses: the modular subset product problem, the multivariate arrangement problem, and the super logarithm problem. The modular subset product problem as a trapdoor function ensures the security of a plaintext encrypted, the multivariate arrangement problem triggered by the lever function $\ell(\cdot)$ ensures the security of a private key, and the super logarithm problem in a prime field ensures the security of a digital signature.

Obviously, REESSE1+ is a type of multi-hard cryptosystem.

A Multi-hard Cryptosystem: If the security of a public-key cryptosystem is based on not less than three hardnesses each of which can not solved in polynomial time, this public-key cryptosystem is called a multi-hard cryptosystem.

The security of a multi-hard cryptosystem is equivalent to that hardness whose time complexity is

smallest in all the involved hardnesses.

A multi-hard cryptosystem must be a multivariate cryptosystem or multiparameter cryptosystem because only multiple variables may bring in multiple hardnesses.

In this paper, unless otherwise specified, sign ‘%’ means ‘modulo’, and (a, b) represents the greatest common divisor of two integers.

2 A Coprime Sequence and the Lever Function

Definition 1 If $A_1, A_2, \dots,$ and A_n are n integers which are each greater than 1, pairwise distinct and relatively prime, this series of integers is called a coprime sequence, namely a relatively prime sequence, denoted by $\{A_1, \dots, A_n\}$, and shortly $\{A_i\}$.

Property 1 For any positive integer $m \leq n$, if we select randomly m elements from $\{A_i\}$ and construct a subset, i.e. a subsequence $\{A_{x_1}, \dots, A_{x_m}\}$, the subset product $G = A_{x_1} \dots A_{x_m}$ is uniquely determined, that is, the mapping from G to $\{A_{x_1}, \dots, A_{x_m}\}$ is one-to-one. G is also called a coprime sequence product.

Proof: We employ apagoge.

Because A_1, \dots, A_n are pairwise relatively prime, for arbitrary $A_j, A_k \in \{A_1, \dots, A_n\}$, there must exist $\gcd(A_j, A_k) = 1$, namely there is not the same prime divisor between A_j and A_k . It manifests that the prime divisors of every element do not belong to any other elements.

Presume that G is acquired from two different subsequences $\{A_{x_1}, \dots, A_{x_m}\}$ and $\{A_{y_1}, \dots, A_{y_h}\}$, hereby

$$G = A_{x_1} \dots A_{x_m} = A_{y_1} \dots A_{y_h}.$$

Since the two subsequences are unequal, there must exist a certain element A_q which does not belong to the two sub-sequences at one time.

Without loss of generality, let $A_q \in \{A_{x_1}, \dots, A_{x_m}\}$ and $A_q \notin \{A_{y_1}, \dots, A_{y_h}\}$.

In terms of the fundamental theorem of arithmetic^[8], there must exist a prime number p which is the divisor of A_q .

It is as above that the prime divisors of every element do not belong to any other elements, and thus the prime p must be the divisor of the product A_{x_1}, \dots, A_{x_m} but not the divisor of the product A_{y_1}, \dots, A_{y_h} . It means that the integer G has two distinct prime factorizations, which is contrary to the fundamental theorem of arithmetic.

Therefore, the mapping relation between G and $\{A_{x_1}, \dots, A_{x_m}\}$ is one-to-one.

In the REESSE1+ cryptosystem, the general key transform is $C_i \equiv A_i W^{\ell(i)} (\% M)$, where $\ell(i)$ is an exponential.

Definition 2 In a public key cryptosystem, the parameter $\ell(i)$ in the key transform is called the lever function, if it has the following features:

- $\ell(\cdot)$ is an injection from integers to integers, its domain is $[1, n]$, and codomain $(1, M)$. Let \mathcal{L}_n represent the collection of all injections from the domain to the codomain, then $\ell(\cdot) \in \mathcal{L}_n$ and $|\mathcal{L}_n| \geq A_n^n = n(n-1) \dots 1$.
- The mapping between i and $\ell(i)$ is established randomly without an analytical formula, so every time a public key is generated, the function $\ell(\cdot)$ is distinct.
- There does not exist any dominant or special mapping from $\ell(\cdot)$ to a public key.
- An attacker have to consider all the arrangements of the sequence $\{\ell(i) \mid i = 1, \dots, n\}$ when extracting a related private key from a public key. Thus, if n is large enough, it is infeasible for the attacker to search the arrangements exhaustively.
- The receiver owning a private key only needs to consider the accumulative sum of the sequence $\{\ell(i)\}$ when recovering a related plaintext from a ciphertext. Thus, the time complexity of decryption is polynomial in n , and the decryption is feasible.

Obviously, there is the large amount of calculation on $\ell(\cdot)$ at ‘a public terminal’, and the small amount of

calculation on $\ell(\cdot)$ at ‘a private terminal’.

3 Design of the REESSE1+ Public Key Cryptosystem

3.1 The Key Generation Algorithm

This algorithm is employed by a third-party authority. Every user is given a pair of keys.

Assume that S, T, D, d are pairwise coprime integers, where the binary form of S only contains two ‘1’ bits respectively at the beginning and the end, $T \geq 2^n, D \geq 2^n$, and d is a non-large integer. Let $|x|$ denote the order of $x \% M$, and ‘ $\% -1$ ’ denote ‘ $\% (M-1)$ ’.

- (1) Randomly generate a coprime sequence $\{A_1, \dots, A_n\}$, and compute $G = \prod_{i=1}^n A_i$.
- (2) Find a prime $M > G$ making $(S, M-1) = 1, dDT | (M-1)$,
and $q | (M-1)$ for any prime $q \in [1, n+4]$.
- (3) Pick δ making $(\delta, M-1) = 1$, and $|\delta| = dDT$.
- (4) Compute $\alpha \leftarrow \delta^{\delta^n}, W \leftarrow G^{-1}(\alpha \delta^{-1})^{1/S}, \beta \leftarrow \delta^{(\delta+1)W^S}$, and $\gamma \leftarrow \delta^{W^n} \% M$.
- (5) Produce pairwise distinct $\ell(1), \dots, \ell(n) \in \Omega = \{i \delta (\% -1) \mid i = 5, \dots, n+4\}$.
- (6) Compute $\{C_1, \dots, C_n \mid C_i \leftarrow A_i W^{\ell(i)} \% M, \text{ for } i = 1, \dots, n\}$.

At last, the public key is $(\{C_i\}, \alpha, \beta, \gamma)$, and the private key $(\{A_i\}, \{\ell(i)\}, W, \delta, D, d)$. S, T , and M are common.

Remark: $\Omega = \{i \delta (\% -1) \mid i = 5, \dots, n+4\}$ is not a unique selection — $\Omega = \{i + \delta (\% -1) \mid i = 5, \dots, n+4\}$ for example. The principles for selecting Ω is that 1) $\ell(i) \geq 5$; 2) the elements in Ω are pairwise distinct; 3) decryption time complexity does not exceed $O(n^3)$.

We know that in degree 5 or higher, the congruence $x^n \equiv c (\% M)$ has a non-solvable Galois group.

To seek a certain element x of order k , first do $x \equiv c^{(M-1)/k} (\% M)$, where $c < M$ is an arbitrary integer, then test x by the algorithm 4.80 in section 4.6 of reference [3].

3.2 The Encryption Algorithm

Assume that $(\{C_i\}, \alpha, \beta, \gamma)$ is the public key, and $b_1 \dots b_n$ is an n -bit plaintext block or symmetric key.

- (1) Set $\hat{G} \leftarrow 1, i \leftarrow 1$.
- (2) If $b_i = 1, \hat{G} \leftarrow \hat{G} C_i \% M$.
- (3) Let $i \leftarrow i + 1$. If $i \leq n$, go to (2), or else end.

After the algorithm is executed, the ciphertext \hat{G} is gained.

Note that in encryption, α, β , and γ are not helpful.

3.3 The Decryption Algorithm

Assume that $(\{A_i\}, \{\ell(i)\}, W, \delta, D, d)$ is the private key, and \hat{G} is the ciphertext.

- (1) Compute $\hat{G} \leftarrow \hat{G} (W^{-1})^\delta \% M$.
- (2) Set $b_1 \dots b_n \leftarrow 0, G \leftarrow \hat{G}, i \leftarrow 1$.
- (3) If $A_i | G$, set $b_i \leftarrow 1$ and $G \leftarrow G / A_i$.
- (4) Let $i \leftarrow i + 1$. If $i \leq n$ and $G \neq 1$, go to (3).
- (5) If $G \neq 1$, go to (1), or else end.

At last, the $b_1 \dots b_n$ is the original plaintext block or symmetric key.

This algorithm can always terminate normally as long as \hat{G} is a true ciphertext

Note that in decryption, $\{\ell(i)\}, D$, and d are not helpful.

3.4 The Digital Signature Algorithm

Assume that $(\{A_i\}, \{\ell(i)\}, W, \delta, D, d)$ is the private key, F is a file or message which will be signed, and $hash$ is a one-way compression function.

- (1) Let $H \leftarrow hash(F)$, whose binary form is $b_1 \dots b_n$.
- (2) Set $k_1 \leftarrow \sum_{i=1}^n b_i \ell(i)$, $G_0 \leftarrow \prod_{i=1}^n A_i^{-b_i}$.
- (3) Pick Q making $D \mid (\delta Q - W)$, $d \nmid ((\delta Q)^n - W^n) (\% -1)$.

Compute R such that $Q \equiv (R G_0)^S H \delta (\% M)$.

- (4) Compute $U \leftarrow (R W^{k_1-1} \delta^{\delta(\delta+1)})^{QT} \% M$.

If $d \nmid ((\delta+1)SU + \sum_{i=0}^{n-1} (\delta Q)^{n-1-i} W^i) (\% -1)$, go to (3).

At last, the signature (Q, U) on the file F is obtained, and sent to a receiver with F .

Note that Q must not be generated repeatedly.

In terms of the double congruence theorem (see sect. 3.6), we do not need $V \equiv (R^{-1} W G_1)^{QU} \delta^\lambda (\% M)$ in the signature, where $G_1 = \prod_{i=1}^n A_i^{b_i}$, and λ satisfies $\lambda S \equiv ((\delta+1)SU + \sum_{i=0}^{n-1} (\delta Q)^{n-1-i} W^i)(\delta Q - W) (\% -1)$. It indicates $d \mid \lambda$.

Clearly the probability of finding out suitable U is $1/d$. Since d is a non-large number, U can be found out at a good pace.

Due to $(S, M-1) = 1$, computing R by $Q \equiv (R G_0)^S H \delta (\% M)$ may resort to the following theorem 1.

Theorem 1 For the congruence $x^k \equiv c (\% p)$ with p is prime, if $(k, p-1) = 1$, every c has just one k -th root modulo p . Especially, let μ be the multiplicative inverse of k modulo $(p-1)$, then $c^\mu \% p$ is one k -th root.

Further, we have theorem 2.

Theorem 2 For the congruence $x^k \equiv c (\% p)$, if $k \mid (p-1)$ and $(k, (p-1)/k) = 1$, then when c is one k -th power residue modulo p , $c^\mu \% p$ is one k -th root, where μ is the multiplicative inverse of k modulo $(p-1)/k$.

The proofs of theorem 1 and 2 are referred to reference [9].

The solution obtained by theorem 1 and theorem 2 is called the trivial solution to the congruence $x^k \equiv c (\% p)$, namely that solution which may be written as c to a certain power modulo p .

3.5 The Identity Verification Algorithm

Assume that $(\{C_i\}, \alpha, \beta, \gamma)$ is the public key, F is the file, and (Q, U) is a signature on it.

- (1) Let $H \leftarrow hash(F)$, whose binary form is $b_1 \dots b_n$.
- (2) Compute $\hat{G} \leftarrow \prod_{i=1}^n C_i^{b_i} \% M$.
- (3) Compute $X \leftarrow (\alpha H Q^{-1})^{QU} \alpha^{QU} \% M$, $Y \leftarrow (\hat{G}^{QU} U^{-1})^{US} \beta^{UT} \gamma^T \% M$.
- (4) If $X \equiv Y$, the identity is valid and F intact,

otherwise the identity is invalid or F already modified.

By running this algorithm, a verifier can judge whether the signature is genuine or fake, prevent the signatory from denying the signature, and do an attacker from modifying the file.

The discriminant $X \equiv Y (\% M)$ at (4) is explained as follows:

It is known from sect. 3.1 that $\alpha \equiv \delta^{\delta^n} \equiv \delta(W G_0 G_1)^S (\% M)$, $\beta \equiv \delta^{(\delta+1)WS} (\% M)$, and $\gamma \equiv \delta^{W^n} (\% M)$.

Let $V \equiv (R^{-1} W G_1)^{QU} \delta^\lambda (\% M)$. Because λ meets $\lambda S \equiv ((\delta+1)SU + \sum_{i=0}^{n-1} (\delta Q)^{n-1-i} W^i)(\delta Q - W) (\% -1)$, may let $\lambda = k d \hat{D}$, where k is a certain integer, and then

$$Q^{QU} V^S \equiv (R G_0)^{SQU} H^{QU} \delta^{QU} (R^{-1} W G_1)^{QU} \delta^{\lambda S}$$

$$\begin{aligned}
 &\equiv (WG_0G_1)^{QU} \delta^{QU} H^{QU} \delta^{\lambda S} \\
 &\equiv \alpha^{QU} H^{QU} \delta^{((\delta+1)SU + \Sigma(\delta Q)^{n-1-i} W^i) (\delta Q - W)} \\
 &\equiv (\alpha H)^{QU} \delta^{-(\delta+1)WSU} \delta^{\delta(\delta+1)QUS} \delta^{\delta Q^n - W^n} \\
 &\equiv (\alpha H)^{QU} \beta^{-U} \delta^{\delta(\delta+1)QUS} \alpha^{Q^n} \gamma^{-1} (\% M).
 \end{aligned}$$

Transposition yields $V^S \equiv (\alpha H Q^{-1})^{QU} \alpha^{Q^n} \beta^{-U} \gamma^{-1} \delta^{\delta(\delta+1)QUS} (\% M)$. Therefore, we have

$$\begin{aligned}
 V^{ST} &\equiv (\alpha H Q^{-1})^{QUT} \alpha^{Q^n T} \beta^{-UT} \gamma^{-T} \delta^{\delta(\delta+1)QUST} \\
 &\equiv X \beta^{-UT} \gamma^{-T} \delta^{\delta(\delta+1)QUST} (\% M).
 \end{aligned}$$

In addition,

$$\begin{aligned}
 U^U V^T &\equiv (R W^{k_1-1} \delta^{\delta(\delta+1)})^{QUT} (R^{-1} W G_1)^{QUT} \delta^{\lambda T} \\
 &\equiv (W^{k_1} G_1)^{QUT} \delta^{\delta(\delta+1)QUT} \delta^{\lambda T} \\
 &\equiv \hat{G}^{QUT} \delta^{\delta(\delta+1)QUT} \delta^{*dDT} \\
 &\equiv \hat{G}^{QUT} \delta^{\delta(\delta+1)QUT} (\% M).
 \end{aligned}$$

Transposition yields $V^T \equiv (\hat{G}^{QT} U^{-1})^U \delta^{\delta(\delta+1)QUT} (\% M)$. Hence

$$V^{ST} \equiv (\hat{G}^{QT} U^{-1})^{US} \delta^{\delta(\delta+1)QUST} (\% M).$$

By the double congruence theorem, there is

$$V^{ST} \equiv X \beta^{-UT} \gamma^{-T} \delta^{\delta(\delta+1)QUST} \equiv (\hat{G}^{QT} U^{-1})^{US} \delta^{\delta(\delta+1)QUST}.$$

Making transposition and counteraction gives $X \equiv (\hat{G}^{QT} U^{-1})^{US} \beta^{UT} \gamma^T (\% M)$.

Namely, $X \equiv Y (\% M)$.

3.6 The Double Congruence Theorem

Theorem 3 (The Double Congruence Theorem) Assume that p is a prime, and that s, t satisfying $(s, t) = 1$ are two constants, then simultaneous equations

$$\begin{cases} x^s \equiv a (\% p) \\ x^t \equiv b (\% p) \end{cases}$$

have the unique solution if and only if $a^t \equiv b^s (\% p)$.

What follows is the proof of theorem 3.

Necessity: Assume that the simultaneous equations $x^s \equiv a (\% p)$ and $x^t \equiv b (\% p)$ have solutions.

Let x_0 be a solution to the two equations, then $x_0^s \equiv a (\% p)$ and $x_0^t \equiv b (\% p)$.

Further, $x_0^{st} \equiv a^t (\% p)$ and $x_0^{st} \equiv b^s (\% p)$ can be obtained.

Therefore, $x_0^{st} \equiv a^t \equiv b^s (\% p)$.

Sufficiency: Assume that $a^t \equiv b^s (\% p)$.

According to the greatest common divisor theorem^[8], there exists a pair of integers u and v making $us + vt = 1$. Thus,

$$\begin{aligned}
 x^{us} &\equiv a^u (\% p), \\
 x^{vt} &\equiv b^v (\% p).
 \end{aligned}$$

The above two equations multiplying yields

$$x^{us+vt} \equiv x \equiv a^u b^v (\% p).$$

Furthermore, we have

$$\begin{aligned}
 (a^u b^v)^s &\equiv a^{us} b^{vs} \equiv a^{us} a^{vt} \equiv a^{us+vt} \equiv a, \\
 (a^u b^v)^t &\equiv a^{ut} b^{vt} \equiv b^{us} b^{vt} \equiv b^{us+vt} \equiv b.
 \end{aligned}$$

Accordingly, $a^u b^v$ is a solution to the original simultaneous equations.

Uniqueness: Let $x_0 \equiv a^u b^v (\% p)$.

Assume that another value x_1 meets the equations $x^s \equiv a (\% p)$ and $x^t \equiv b (\% p)$ at one time.

Then, it holds that

$$x_1^s \equiv a (\% p), \text{ and } x_1^t \equiv b (\% p).$$

By comparison, we have $x_1^s \equiv x_0^s$ and $x_1^t \equiv x_0^t (\% p)$. Transposing gives

$$(x_0 x_1^{-1})^s \equiv 1 \text{ and } (x_0 x_1^{-1})^t \equiv 1 (\% p).$$

If at least one of s and t is relatively prime to $p-1$, by theorem 1, there must be $x_0 x_1^{-1} \equiv 1 (\% p)$, namely $x_0 \equiv x_1 (\% p)$.

If neither s nor t is relatively prime to $p-1$, let $k = (s, p-1)$, $l = (t, p-1)$. Then $(s/k, p-1) = 1$ and $(t/l, p-1) = 1$. Thus, there are $(x_0 x_1^{-1})^k \equiv 1$ and $(x_0 x_1^{-1})^l \equiv 1$. It is known from $(s, t) = 1$ that $(k, l) = 1$. In terms of the group theory^[10], when $(k, l) = 1$, only the element '1' belongs to two different sub-group at same time. Therefore, $x_0 x_1^{-1} \equiv 1$, namely $x_1 = x_0$, and x_0 bears uniqueness.

To sum up, we prove theorem 3.

3.7 Characteristics of REESSE1+

REESSE1+ owes the following characteristics compared with classical MH, RSA and ElGamal cryptosystems.

- The security of REESSE1+ is not based on a single hardness, but on multiple hardnesses: the modular subset product problem, multivariate arrangement problem, and super logarithm problem. Thus, it is a multi-hard public-key cryptosystem.
- The key transform $C_i \equiv A_i W^{\ell(i)} (\% M)$ is a compound function, and contains three independent variables, that is, C_1, C_2, \dots , and C_n contain $2n+1$ unknown variables. Hence, REESSE1+ is also a multivariate cryptosystem.
- If any of A_i, W and $\ell(i)$ is determined, the relation between the two remainders is still nonlinear — thus there is very complicated nonlinear relations among A_i, W and $\ell(i)$.
- There is indeterminacy of $\ell(i)$. On condition that C_i and W are determined, A_i and $\ell(i)$ can not be determined, and even have no one-to-one relation when W is a non-generator. On condition that C_i and A_i are determined, W and $\ell(i)$ can not be determined, and also have no one-to-one relation for $(\ell(i), M-1) > 1$. This is the radical reason that the continued fraction analysis method is ineffectual.
- There is insufficiency of the key mapping. A private key in REESSE1+ includes $\{A_i\}, \{\ell(i)\}, W$ and δ four main parts, but there is only a dominant mapping from $\{A_i\}$ to $\{C_i\}$. Thereby, the reversibility of the function is not obvious, and inferring a private key is intractable through mathematical methods.
- Since the length and the elements of the set Ω are not fixed, and different combinations among variables may bring in different hardnesses, REESSE1+ is a sort of flexible public key cryptosystem.

3.8 Correctness of the Decryption Algorithm

Because (\mathbb{Z}_M^*, \cdot) is an Abelian, namely exchangeable group, $\forall k \in [1, M)$, there is

$$W^k (W^{-1})^k \equiv W^k W^{-k} \equiv 1 (\% M).$$

Let $b_1 \dots b_n$ be an n -bit plaintext, and $k = (\sum_{i=1}^n \ell(i) b_i) \delta^{-1} \% -1$.

Note that due to $(\delta, M-1) = 1$, there exists $\delta^{-1} \% -1$.

We need to prove that $\hat{G}(W^{-\delta})^k \equiv G (\% M)$.

According to sect. 3.2, $\hat{G} \equiv \prod_{i=1}^n C_i^{b_i} (\% M)$, where $C_i \equiv A_i W^{\ell(i)} (\% M)$, hence

$$\begin{aligned} \hat{G}(W^{-\delta})^k &\equiv \prod_{i=1}^n C_i^{b_i} (W^{-\delta})^k \equiv \prod_{i=1}^n (A_i W^{\ell(i)})^{b_i} (W^{-\delta})^k \\ &\equiv \prod_{i=1}^n A_i^{b_i} (W^{\sum (\ell(i) b_i)}) (W^{-\delta})^k \end{aligned}$$

$$\begin{aligned}
 &\equiv \prod_{i=1}^n A_i^{b_i} (W^\delta)^k (W^{-\delta})^k \\
 &\equiv \prod_{i=1}^n A_i^{b_i} \\
 &\equiv G \pmod{M}.
 \end{aligned}$$

The above process gives out a method for seeking G .

Note that in practice, the plaintext $b_1 \dots b_n$ is unknowable in advance, so we have no way to directly compute k . However, because the range of $k \in [5, \delta^{-1} \sum_{i=1}^n \ell(i)]$ is very narrow, we may search k heuristically by multiplying $(W^{-1})^\delta \pmod{M}$, and verify whether G is equal to 1 after it is divided exactly by some items of $\{A_i\}$. It is known from sect. 3.3 that the original plaintext $b_1 \dots b_n$ is acquired at the same time the condition $G = 1$ is satisfied.

3.9 Uniqueness of a Plaintext Solution to a Ciphertext

Because $\{C_i\}$ is one non-coprime sequence, the mapping from the subsequence $\{C_{x_1}, \dots, C_{x_m}\}$ to the product \hat{G} is theoretically many-to-one. It might possibly result in the nonuniqueness of the plaintext solution $b_1 \dots b_n$ when \hat{G} is being unveiled.

Suppose that the ciphertext \hat{G} can be obtained from two different subsequence products, that is,

$$\hat{G} \equiv C_{x_1} \dots C_{x_m} \equiv C_{y_1} \dots C_{y_h} \pmod{M}.$$

Then,

$$(A_{x_1} \dots A_{x_m}) W^{k_1} \equiv (A_{y_1} \dots A_{y_h}) W^{k_2} \pmod{M},$$

where $k_1 = \ell(x_1) + \dots + \ell(x_m)$, and $k_2 = \ell(y_1) + \dots + \ell(y_h)$.

Without loss of generality, let $k_1 \geq k_2$. Because (\mathbb{Z}_M^*, \cdot) is an Abelian group, there is

$$W^{k_1 - k_2} \equiv (A_{y_1} \dots A_{y_h}) (A_{x_1} \dots A_{x_m})^{-1} \pmod{M},$$

which is written shortly as $W^{k_1 - k_2} \equiv \prod_{i=1}^h A_{y_i} \prod_{j=1}^m A_{x_j}^{-1} \pmod{M}$.

Let $\theta \equiv W^{k_1 - k_2} \equiv (W^\delta)^{(k_1 - k_2) \delta^{-1}} \pmod{M}$.

This formula means when the plaintext $b_1 \dots b_n$ is not unique, the value of W^δ must be relevant to θ . The contrapositive assertion equivalent to it is that if the value of W^δ is irrelevant to θ , $b_1 \dots b_n$ will be unique. Thus, we need to consider the probability that W^δ takes a value relevant to θ .

If an adversary tries to attack an 80-bit symmetric key through the exhaustive search, and a computer can verify trillion values per second, it will take 38334 years for the adversary to verify up all the potential values. Hence, currently 80 bits are quite enough for the security of a symmetric key.

When the length n of a key sequence is equal to 80, the number of the values containing the repeated in the form $\prod_{i=1}^h A_{y_i} \prod_{j=1}^m A_{x_j}^{-1}$ is at most $3^{80} \approx 2^{1.585 \times 80} = 2^{127}$. Because $A_1^{-1} \dots A_n^{-1}$ are not necessarily coprime, the value of $\prod_{i=1}^h A_{y_i} \prod_{j=1}^m A_{x_j}^{-1}$ may possibly occur repeatedly.

On the other hand, the first 80 primes may constitute a coprime sequence with a minimum sequence product, which makes it possible that the modulus M is roughly equal to 2^{552} or M roughly equal to 2^{384} with binary compact sequence optimization. Therefore, when $n \geq 80$, the probability that W^δ takes values relevant to θ is less than $(2^{13} 2^{127}) / 2^{384} = 1 / 2^{244}$, where 2^{13} is greater than or roughly equal to $(k_1 - k_2) \delta^{-1}$, namely the number of W^δ meeting $(W^\delta)^{(k_1 - k_2) \delta^{-1}} \equiv \prod_{i=1}^h A_{y_i} \prod_{j=1}^m A_{x_j}^{-1} \pmod{M}$ is at most 2^{13} . Clearly, it is almost zero. This probability will further decrease when W^δ is prime.

The above analysis shows that the probability that the plaintext solution $b_1 \dots b_n$ is not unique is almost zero; thus the decryption algorithm can always recover the original plaintext from the ciphertext \hat{G} , which is also verified by the program in C language.

4 Securities of Encryption and Decryption

4.1 Extracting a Private Key from a Public Key Being the Multivariate Arrangement Hardness

A public key may be regarded as the special cipher of a related private key. Since a ciphertext is the effect of a public key and a plaintext, the ciphertext has no direct help to inferring the private key.

In the REESSE1+ system, the key transform is $C_i \equiv A_i W^{\ell(i)} (\% M)$, and $\ell(i) \in \{i \delta (\% -1) \mid i = 5, \dots, n + 4\}$.

For a specific C_i , assume that the corresponding A_i and W are revealed under some extreme condition. Due to $\ell(i) \in (1, M)$, obviously, by $W^{\ell(i)} \equiv C_i A_i^{-1} (\% M)$ seeking $\ell(i)$ is the DLP. Thus, under normal situations, inferring a related private key from a public key is harder than the DLP.

In what follows, we discuss the case the n items of $\{C_i\}$ are considered all together.

If an attacker tries to extract a related private key from a public key, it is equivalent to solve the simultaneous equations

$$\begin{cases} C_1 \equiv A_1 W^{\ell(1)} (\% M) \\ C_2 \equiv A_2 W^{\ell(2)} (\% M) \\ \dots\dots\dots \\ C_n \equiv A_n W^{\ell(n)} (\% M). \end{cases}$$

The above equation system contains n known variables, and $2n + 1$ unknown variables.

Assume that the $P > n$ is the largest prime constant in the REESSE1+ cryptosystem, then each $A_i \in \Gamma = \{2, \dots, P\}$, where Γ contains at least n primes. Let \tilde{N} be the number of the potential coprime sequences in the interval $[2, P]$, then $\tilde{N} > A_n^n = n!$.

If let $\ell(1)\delta^{-1} = \dots = \ell(n)\delta^{-1} = 5$, and each A_i traverse Γ , then can obtain theoretically $5nP$ values where exists the true value of W^δ . Therefore, the number of potential values of W^δ will decrease to $5nP$. Note that in fact computing all five W^δ from $(W^\delta)^5 = C_i A_i^{-1} (\% M)$ is intractable when $5 \mid (M - 1)$ and $(M - 1) / 5 > 2^{80}$ (see sect. 5.1).

Suppose that the attacker guesses the sequences $\{A_i\}$ and $\{\ell(i)\delta^{-1}\}$, then figures out the n values of W^δ in $O(T_W)$ time. If these values are all identical, the guessing is thought right. Notice that for i -th equation, A_i is allowed to take any element of Γ as long as it is pairwise coprime to A_1, \dots, A_{i-1} , and $\ell(i)\delta^{-1}$ is allowed to take any element of $\{5, \dots, n + 4\}$ as long as it is pairwise different from $\ell(1), \dots, \ell(i - 1)$, which means that guessing $\{A_i\}$ or $\{\ell(i)\delta^{-1}\}$ is an arrangement problem. Thereby, the time complexity of this attack is $O(\tilde{N}(n!)T_W) > O(2^n)$.

Suppose that the attacker guesses W^δ and the sequences $\{\ell(i)\delta^{-1}\}$, then compute the sequence $\{A_i\}$ in $O(T_A)$ time. If $\{A_i\}$ is a coprime sequence, the guessing is thought successful. Because guessing $\{\ell(i)\delta^{-1}\}$ is also an arrangement problem, the time complexity of this attack is at least $O(5nP(n!)T_A) > O(2^n)$.

Suppose that the attacker guesses W^δ and the sequences $\{A_i\}$, then find out the sequence $\{\ell(i)\delta^{-1}\}$ in $O(T_\ell)$ time. If every $\ell(i)\delta^{-1} \in \{5, \dots, n + 4\}$ and is pairwise distinct, the guessing is thought successful. Because guessing $\{A_i\}$ is likewise an arrangement problem, the time complexity of this attack is at least $O(5nP\tilde{N}T_\ell) > O(5nP(n!)T_\ell) > O(2^n)$.

Since W , δ and (G_0G_1) are all unknown, it is impossible to infer W , δ or (G_0G_1) from $\alpha \equiv \delta(WG_0G_1)^S (\% M)$.

The expressions $\alpha \equiv \delta^{\delta^n}$, $\beta \equiv \delta^{(\delta+1)W^S}$ and $\gamma \equiv \delta^{W^n} (\% M)$ only contain two unknown variables, but the time complexity of finding δ and W will be at least $O(2^n)$ (see sect. 5.2.3).

Further, we can argue that the time complexity of the continued fraction attack is $O(n!) > O(2^n)$.

In summary, the time complexity of inferring a related private key from a public key is at least $O(2^n)$.

4.2 Recovering a Plaintext from a Ciphertext and a Public Key Being the Modular Subset Product Hardness

In terms of sect. 3.2, the ciphertext is $\hat{G} \equiv \prod_{i=1}^n C_i^{b_i} (\% M)$, where $b_1 \dots b_n$ is a plaintext block or a

symmetric key, and $\{C_1, \dots, C_n\}$ is a public key.

Observe an extreme case. Assume that $C_1 = \dots = C_n = C$, then $\hat{G} \equiv \prod_{i=1}^n C^{b_i} (\% M)$. It can be written as

$$\hat{G} \equiv C^{\sum b_i} (\% M),$$

where i is from 1 to n .

Because we need not only to figure out the value of $\sum_{i=1}^n b_i$ but also to find out the position of every $b_i = 1$, we express equivalently the sum $\sum_{i=1}^n b_i$ as $\sum_{i=1}^n b_i 2^{i-1}$, and let $x = \sum_{i=1}^n b_i 2^{i-1}$. Correspondingly,

$$\hat{G} \equiv C^x (\% M).$$

The above manifests that seeking the exponent x of C is the DLP.

The above process is reversible. It shows that if the plaintext recovery problem can be solved, the DLP can be solved. Therefore, when $C_1 \neq \dots \neq C_n$, attempting to recover a related plaintext from a known ciphertext and public key is more intractable than the DLP, which is essentially different from the subset sum problem or the knapsack problem.

On the other hand, there exists the exhaustive search attack on $b_1 \dots b_n$, and clearly, the time complexity of this attack is $O(2^n)$.

Notice that when \hat{G} is converted into the coprime sequence product G , computing the plaintext $b_1 \dots b_n$ is tractable. Namely there is a trapdoor for \hat{G} . Hence, the plaintext security of REESSE1+ is built on a trapdoor function such that computing a subset product from subset elements is tractable while seeking the involved elements from the subset product is intractable.

4.3 Avoiding the Adaptive-chosen-ciphertext Attack

Absolute most of public key cryptographies will probably be faced by the adaptive-chosen-ciphertext attack apart from ElGamal, ECC and so on, although this attack method still stays on a concept level, and is not implemented in the concrete applications.

It is lucky that REESSE1+ can avoid the adaptive-chosen-ciphertext attack. In the REESSE1+ cryptosystem, a public key is a sequence, and the number of its potential arrangements is $n!$, and thus a secret time function may be chosen. Every time encryption is done, a different arrangement of the public key according to the time is employed to encrypt a plaintext. In this way, even if the plaintexts encrypted are the same, due to different encrypting times, the related ciphertexts are likely unequal. Of course, should let those C_i to which values of the plaintext bits corresponding are all 1 permute as much as possible. So and so only can the multiple ciphertexts of the identical plaintext be different from one another.

Another approach to avoiding the above attack is to append a stochastic fixed-length binary sequence to the terminal of every plaintext block when it is encrypted.

5 Securities of Digital Signature and Identity Verification

5.1 Extracting a Related Private Key from a Signature Being a Hardness

Assume that p is a prime, and $k \mid (p - 1)$. In terms of the probabilistic algorithm in section 1.6 of reference [11], the time complexity of finding out a random solution to $x^k \equiv c (\% p)$ is $\max(O(2^{k-1}), O(p/k))$. Thus, when $k > 80$ or $p/k > 2^{80}$, this algorithm is ineffectual currently.

However, when $(k, (p - 1) / k) = 1$ with $k \mid (p - 1)$ or $(k, p - 1) = 1$, the trivial solution to $x^k \equiv c (\% p)$ can be acquired in terms of theorem 1 and 2.

It is known from the digital signature algorithm that

$$Q \equiv (R G_0)^S H \delta (\% M), \text{ and } U \equiv (R W^{k_1-1} \delta^{\delta(\delta+1)})^{Q^T} (\% M).$$

When an attacker wants to seek $R G_0$, or $R W^{k_1-1} \delta^{\delta(\delta+1)}$, it is equivalent to solving the congruences

$$x^S \equiv Q H^{-1} \delta^{-1} (\% M), \text{ and } x^{Q^T} \equiv U (\% M).$$

For the first equation, because δ is unknown, and the right of the equation is not a constant, it is

impossible to solve the equation for RG_0 . If δ is guessed, the probability of hitting δ is $1/|\mathcal{J}| < 1/2^n$.

For the second equation, due to $T \mid (M-1)$, there is $(QT, M-1) \geq T$. If there exists the trivial solution to the equation, the probability that it is just the specific solution $(RW^{k_1-1}\delta^{\delta(\delta+1)})$ is equal to or less than $1/T \leq 1/2^n$. Even though $(RW^{k_1-1}\delta^{\delta(\delta+1)})$ is found out, due to the randomness of $R \in (1, M-1)$, neither of W^{k_1-1} and $\delta^{\delta(\delta+1)}$ can be determined.

Therefore, the time complexity of extracting a related private key from a signature is at least $O(2^n)$.

5.2 Faking a Digital Signature only through a Public Key Being a Hardness

Assume H is the hash value of F , (Q, U) is a signature on F , then the condition discriminant

$$(\alpha H Q^{-1})^{QU^T} \alpha^{Q^T} \equiv (\tilde{G}^{QU^T} U^{-1})^{US} \beta^{UT} \gamma^T \pmod{M}$$

holds. Due to an equation with the two variables, the value of a variable may be supposed by an attacker. However, supposing a value and seeking the other is the super logarithm problem.

5.2.1 The Super Logarithm Problem

Assume that $g \in \mathbb{Z}_p^*$ is a generator, where p is prime, then $\{y \mid y \equiv g^x \pmod{p}, x = 1, \dots, p-1\} = \mathbb{Z}_p^*$ [10].

Assume that n satisfying $(k, p-1) = 1$ is an integer, then also $\{y \mid y \equiv x^k \pmod{p}, x = 1, \dots, p-1\} = \mathbb{Z}_p^*$ [9].

Namely, $\forall x \in [1, p-1], y \equiv g^x \pmod{p}$ or $y \equiv x^k \pmod{p}$ with $(k, p-1) = 1$ is a self-isomorph of the group \mathbb{Z}_p^* .

However, for the x^x operation, $\{y \mid y \equiv x^x \pmod{p}, x = 1, \dots, p-1\} = \mathbb{Z}_p^*$ does not hold, that is,

$$\{y \mid y \equiv x^x \pmod{p}, x = 1, \dots, p-1\} \neq \mathbb{Z}_p^*.$$

For example, when $p = 11$, $\{y \mid y \equiv x^x \pmod{p}, x = 1, \dots, p-1\} = \{1, 3, 4, 5, 6\}$, where $3^3 \equiv 6^6 \equiv 8^8 \equiv 5 \pmod{11}$.

When $p = 13$, $\{y \mid y \equiv x^x \pmod{p}, x = 1, \dots, p-1\} = \{1, 3, 4, 5, 6, 9, 12\}$, where $7^7 \equiv 11^{11} \equiv 6 \pmod{13}$, and $1^1 \equiv 3^3 \equiv 8^8 \equiv 9^9 \equiv 12^{12} \equiv 1 \pmod{13}$.

When $p = 17$, $\{y \mid y \equiv x^x \pmod{p}, x = 1, \dots, p-1\} = \{1, 2, 4, 8, 9, 10, 12, 13, 14\}$, where $2^2 \equiv 12^{12} \equiv 4 \pmod{17}$, $6^6 \equiv 15^{15} \equiv 2 \pmod{17}$, and $10^{10} \equiv 14^{14} \equiv 2 \pmod{17}$.

The above examples illustrate that $\{y \equiv x^x \pmod{p} \mid x = 1, \dots, p-1\}$ cannot construct a complete set for a group. Furthermore, mapping from x to y is one-to-one sometimes, and many-to-one sometimes. That is, inferring x from y is indeterminate, x is non-unique, and even inexistent. Thus, x^x has extremely strong irregularity, and is essentially distinct from g^x and x^k .

We consider two functions over the real set \mathbb{R} : $y = f(x) = g^x$, and $y = \varphi(x) = x^x$.

Their inverse functions, the derivations of which are omitted, are respectively

$$x = f^{-1}(y) = \log_g y,$$

$$\text{and } x = \varphi^{-1}(y) = y \log_g y / ((y^y - y) \log_g e),$$

where $y^y = x^x (1 + \log_g x / \log_g e)$ denotes a derivative, and the constant $e = 2.7182818\dots$

Further, $\varphi^{-1}(y) = y f^{-1}(y) / ((y^y - y) \log_g e)$.

Assume that we already know y_0 satisfying $y_0 = x_0^{x_0}$ and $y_0 = g^{x_0}$. Apparently, if $\varphi^{-1}(y_0)$, namely x_0 can be found out, $f^{-1}(y_0)$, namely x_0 can be figured out. This means that the time complexity of computing $f^{-1}(y)$ is less than or equal to that of computing $\varphi^{-1}(y)$.

Contrariwise, if $f^{-1}(y_0)$, namely x_0 can be found out, $\varphi^{-1}(y_0)$, namely x_0 can not be figured out since y_0 is the function of x_0 , and has no solution in polynomial time. This means that the time complexity of computing $f^{-1}(y)$ is not equal to that of computing $\varphi^{-1}(y)$.

To sum up, the time complexity of seeking $\varphi^{-1}(y)$ is greater than that of seeking $f^{-1}(y)$.

Similarly, this fact should hold in the finite field $\mathbb{GF}(p)$, because the discreteness of a finite field does not weaken the computational complexity of the identical problem over a continuous interval. For instance, computing $x = f^{-1}(y) = \log_g y$ is easy in a continuous interval while hard in a finite field.

In summary, we think that the $x^x \equiv c \pmod{p}$ problem is harder than the $g^x \equiv c \pmod{p}$ problem. Hence, the former is called the super logarithm hardness. It is emphasized that the super logarithm hardness is more suitable for doing signature since it owns non-uniqueness.

Note that to attempt to solve the super logarithm problem in light of the Chinese Remainder Theorem is specious.

At present there is no better method for seeking a super logarithm than the exhaustive search, and thus the time complexity of the solution to $x^x \equiv c \pmod{p}$ may be expected to be $O(p) > O(2^n)$, where n is the length of a message digest.

5.2.2 Faking a Signature by the Verification Algorithm Being the Super Logarithm Problem

Assume that F is an arbitrary file, and H is its hash output. In terms of the discriminant

$$(\alpha H Q^{-1})^{QU T} \alpha^{Q^N T} \equiv (\hat{G}^{QT} U^{-1})^{US} \beta^{UT} \gamma^T \pmod{M},$$

an attacker may suppose the value of a variable.

If suppose the value of Q , no matter whether U exists or not, seeking U is the super logarithm hardness.

Similarly, if suppose the value of U , seeking Q is the super logarithm hardness.

If the attacker hits the small \bar{d} , and raises either side of the discriminant to the power of \bar{d} , then when $\bar{D} \mid (\bar{\delta}Q - W)$, there is

$$(\alpha H Q^{-1})^{\bar{d}QU T} \equiv (\hat{G}^{QT} U^{-1})^{\bar{d}US} \beta^{\bar{d}UT} \pmod{M}.$$

Further, $(\alpha H Q^{-1})^{\bar{d}QU T} \equiv (\hat{G}^{QT} U^{-1})^{\bar{d}S} \beta^{\bar{d}T} \pmod{M}$.

Hereat, if suppose the value of Q which is not repeated and known, U may possibly be worked out in polynomial time. However, Q and U must satisfy both the discriminant and $\bar{D} \mid (\bar{\delta}Q - W)$, which is impossible since $\bar{\delta}$ and W are unknown.

5.2.3 Faking a Signature by the Signature Algorithm Being the Exponential Time Problem

Due to $Q \equiv (R G_0)^S H \delta$, $U \equiv (R W^{k_1-1} \delta^{\delta(\delta+1)})^{QT}$ and $V \equiv (R^{-1} W G_1)^{QU} \delta^\lambda \pmod{M}$, an attacker may attempt the following attack method.

Let $a c \equiv (\alpha \delta^{-1})^{1/S}$, and $b c \equiv \hat{G} \pmod{M}$. When δ is sought, with supposing the value of a , c and b may be figured out.

Let $Q \equiv a^S H \delta$, $U \equiv b^{QT} \delta^{\delta(\delta+1)QT}$, and $V \equiv c^{QU} \delta^\lambda \pmod{M}$, where λ meets $\lambda S \equiv ((\delta+1)SU + \sum_{i=0}^{n-1} (\delta Q)^{n-1-i} W^i)(\delta Q - W) \pmod{-1}$, and Q, U make $\bar{D} \mid (\bar{\delta}Q - W)$ and $\bar{d} \mid ((\delta+1)SU + \sum_{i=0}^{n-1} (\delta Q)^{n-1-i} W^i) \pmod{-1}$. If the attacker seeks δ and W such taht

$$\begin{cases} \alpha \equiv \delta^{\delta^n} \pmod{M} \\ \beta \equiv \delta^{(\delta+1)WS} \pmod{M} \\ \gamma \equiv \delta^{W^n} \pmod{M}, \end{cases} \quad (*1)$$

then the attack is successful.

Can δ and W be sought? If δ exists, there is

$$\begin{cases} \alpha^{W^n} \equiv \gamma^{\delta^n} \pmod{M} \\ \beta^{W^n} \equiv \gamma^{(\delta+1)WS} \pmod{M}. \end{cases} \quad (*2)$$

Note that because possibly $(W^n, \delta^n) > 1$, (*2) is not a sufficient condition for (*1) to have solutions.

Let g be a generator of (\mathbb{Z}_M^*, \cdot) , by means of Index-calculus method for discrete logarithms^[3], figure out q , u , and v such that $g^q \equiv \alpha$, $g^u \equiv \beta$, and $g^v \equiv \gamma \pmod{M}$, obtain

$$\begin{cases} g^{qW^n} \equiv g^{v\delta^n} \pmod{M} \\ g^{uW^n} \equiv g^{v(\delta+1)WS} \pmod{M}. \end{cases}$$

Accordingly,

$$\begin{cases} qW^n \equiv v\delta^n (\% -1) \\ uW^n \equiv v(\delta+1)WS (\% -1). \end{cases}$$

If $(q, M-1) \mid v$, and the trivial solution to $x^n \equiv c (\% -1)$ exists, then $W \equiv (q^{-1}v)^{1/n} \delta (\% -1)$. Further, if $W^{-1} \% -1$ exists, there is

$$u(q^{-1}v)^{n-1/n} \delta^{n-1} \equiv v(\delta+1)S (\% -1).$$

Transparently, the above equation is a true polynomial of degree $n-1$, which can not be solved via discrete logarithms. If Euler phi function $\phi(M-1)$ is worked out, the attacker may seem to resort to the probabilistic algorithm in section 1.6 of reference [11]. However, the running time of this algorithm is $\max(O(2^{n-2}), O(\phi(M-1)/(n-1)))$. When $n > 80$ or $\phi(M-1)/(n-1) > 2^{80}$, seeking δ is infeasible at present. Furthermore, when $(u(q^{-1}v)^{n-1/n}, M-1) \nmid v$, doing division of polynomials is infeasible.

5.3 Faking Another Signature through a Known Signature with a Public Key Being a Hardness

Given the file F and a signature (Q, U) on it, and assume that there exists another file F' with corresponding H' and \hat{G}' . Then, if an arbitrary (Q', U') satisfies

$$(\alpha H' Q'^{-1})^{Q' U'^T} \alpha^{Q'^n T} \equiv (\hat{G}'^{Q'^T} U'^{-1})^{U' S} \beta^{U'^T} \gamma^T (\% M),$$

it is a signature fraud on F' .

According to the stipulation in the signature algorithm, Q can not be generated repeatedly, but the value of U is allowed to be utilized repeatedly.

If let $U' = U$, no matter whether the preceding equation has solutions or not, seeking Q' is the super logarithm hardness.

If several of the pair (Q, U) are known, because R involved in U is random, and there is not functional or statistical relation among different U , they are not helpful to solving the super logarithm problem.

Therefore, forging another signature via a known signature with a public key is the super logarithm hardness.

5.4 Faking a Signature through a Chosen-plaintext Being a Hardness

Due to $H = b_1 \dots b_n$, $\hat{G} \equiv \prod_{i=1}^n C_i^{b_i} (\% M)$, there is not a linear expression relation between H and \hat{G} .

It is understood from the discriminant $(\alpha H Q^{-1})^{Q U^T} \alpha^{Q^n T} \equiv (\hat{G}^{Q^T} U^{-1})^{U S} \beta^{U^T} \gamma^T (\% M)$ that

$$H^{Q U^T} \equiv (\hat{G}^{Q^T} U^{-1})^{U S} \beta^{U^T} \gamma^T (\alpha^{-1} Q)^{Q U^T} \alpha^{-Q^n T} (\% M).$$

Assume that Q and U are known. Since \hat{G} is unknown, it is impossible to seek H by the above congruence.

It is also understood from the discriminant

$$\hat{G}^{Q U^T} \equiv U^{U S} \beta^{-U^T} \gamma^{-T} (\alpha H Q^{-1})^{Q U^T} \alpha^{Q^n T} (\% M).$$

Assume that H is a constant irrelevant to \hat{G} , and then the trivial solution for \hat{G} may be possibly found out. However, it is harder than the DLP to infer H from \hat{G} (see sect. 4.2).

6 Conclusions

The REESSE1+ cryptosystem may be applied to data encryption and digital signature. If it is only applied to digital signature, the constraint $M > G$ at step 2 in the key generation algorithm may be removed.

If the constraint $M > G$ can not be removed, then when $n \geq 80$, the modulus M will be comparatively large. It will cause both increase in key lengths and, worse than all, decrease in algorithmic speeds. For example, $\log_2 M \geq 553$ (to select the first 80 primes as a coprime sequence make the sign '=' hold) when $n = 80$, $\log_2 M \geq 694$ when $n = 96$, $\log_2 M \geq 840$ when $n = 112$, and $\log_2 M \geq 990$ when $n = 128$. Thus, the theoretical algorithms of REESSE1+ must be optimized.

The basic idea of optimization is to make use of binary compact sequences. That is, the mapping is not

regarded as one to one but as three items to two bits between a non-coprime sequence and a symmetric key. (or a plaintext block). The n -bit symmetric key is partitioned evenly into $n/2$ units in order, and every unit has 2 bits and 4 combinations, namely 00, 01, 10, and 11. Let 00 map to 1, and the others do respectively to the three consecutive items of $\{C_i\}$. After optimized, the bit-length of M will decrease to 384 when $n = 80$, and 544 when $n = 112$.

Through the optimization based on the binary compact sequence, the decryption time complexity of the REESSE1+ cryptosystem is reduced to $O(n^2)$ from $O(n^3)$. Through the integration of a coprime sequence and the lever function $\ell(\cdot)$, we exchange the very slight cost of operations for the great security of the REESSE1+ cryptosystem.

Acknowledgment

The authors would like to thank Mulan Liu, Huanguo Zhang, Dengguo Feng, Xiulin Zheng, Lequan Min, and Maozhi Xu for their important suggestions, corrections, and encouragements.

References

- [1] R. L. Rivest, A. Shamir and L. M. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," *Communications of the ACM*, 21(2), 1978, pp.120-126.
- [2] T. ElGamal, "A Public-key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, v. 31, n. 4, 1985, pp. 469-472.
- [3] A. J. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, London: CRC Press, 1997, ch. 2, 8.
- [4] Oded Goldreich, *Foundations of Cryptography: Basic Tools*, Cambridge, UK: Cambridge University Press, 2001, pp. 31–35.
- [5] I. F. Blake, G. Seroussi, and N. P. Smart, *Elliptic Curves in Cryptography*, Cambridge, UK: Cambridge University Press, 1999, pp. ch. 1, 6.
- [6] R. C. Merkle and M. E. Hellman, "Hiding information and Signatures in Trapdoor Knapsacks," *IEEE Transactions on Information Theory*, v. 24, n. 5, 1978, pp. 525-530.
- [7] A. Shamir, "A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem," *Proceedings of the 23th IEEE Symposium on the Foundations of Computer Science*, 1982, pp. 145-152.
- [8] Song Y. Yan, *Number Theory for Computing*, 2nd ed., Berlin: Springer-Verlag, 2002, ch. 1.
- [9] Paul Garrett, *Making, Breaking Codes: An Introduction to Cryptology*, New Jersey: Prentice-Hall, 2001, ch. 12.
- [10] Thomas W. Hungerford, *Algebra*, New York: Springer-Verlag, 1998, ch. 1–3.
- [11] Henri Cohen, *A Course in Computational Algebraic Number Theory*, Berlin: Springer-Verlag, 2000, ch. 1, 3.