The REESSE1+ Public Key Cryptosystem Version 2.2 *

Shenghui Su [1], and Shuwang Lü [2]

[1] College of Computer Science, Beijing University of Technology, Beijing 100124, P.R.China
[2] School of Graduate, Chinese Academy of Sciences, Beijing 100039, P.R.China

Abstract: The authors give the definitions of a coprime sequence and a lever function, and describe five algorithms and six characteristics of a prototypal cryptosystem called REESSE1+, used for encryption and signature, and based on the three new hardnesses: multivariate permutation problem (MPP) $C_i \equiv (A_i W^{\ell(i)})^{\delta}$ (% M), anomalous subset product problem (ASPP) $\bar{G} \equiv \prod_{i=1}^{n} C_i^{\underline{b}_i}$ (% M), and transcendental logarithm problem (TLP) $y \equiv x^x$ (% M). Some evidences show that MPP, ASPP, TLP are harder than DLP in the same prime field, namely they can not be solved in DLP subexponential time. Prove correctness of the decryption and verification, deduce the probability of a plaintext solution being nonunique is nearly zero, discuss necessity and sufficiency of the lever function for resisting continued fraction attacks, expound a relation between the lever function and a random oracle, analyze exact securities of REESSE1+ against recovering a plaintext from a ciphertext, extracting a private key from a public key or a signature, and forging a signature from known signatures, public keys, and messages on the assumption that IFP, DLP, SSP can be solved, and find that running times of effectual attack tasks are greater than or close to $O(2^n)$ when n = 80, 96, 112, or 128 with $\log_2 M \approx 696$, 864, 1030, or 1216. As viewed from utility, it should be researched further how to decrease the length of the modulus and to increase the speed of the decryption.

Keywords: Public key cryptosystem, Coprime sequence, Lever function, Digital signature, Double congruence theorem, Transcendental logarithm problem, Polynomial time reduction

1 Introduction

The trapdoor functions for RSA [1] and ElGamal [2] public key cryptosystems [3] are computationally one-way [4][5], which indicates that there always exists one sufficiently large setting of the security dominant parameter that makes utilizing a cryptosystem feasible and breaking the cryptosystem infeasible in polynomial time [6]. Instancing RSA based on the integer factorization problem (IFP), when the bit-length of its modulus reaches 1024, the decomposition is infeasible but the encryption and decryption are feasible in polynomial time. Such a security is referred to as asymptotic security, which is distinguished from exact security or concrete security. The exact security is practice-oriented, and aims at giving more precise estimates of the time complexities of attack tasks [7].

In some public key cryptosystems, trapdoor functions can prevent a related plaintext from being recovered from a ciphertext, but can not prevent a related private key from being extracted from a public key. For instance, in the MH knapsack system [8], the subset sum problem (SSP) which contains trapdoor information can not prevent a related private key from being extracted from $c_i \equiv a_i W$ (% M) through the accumulation points of minima [9], and moreover, when the knapsack density is less than 1, SSP will degenerate to a polynomial time problem from a NPC problem owing to the L^3 lattice base reduction algorithm [10][11] which is employed for finding the shortest vector or an approximately shortest vector in a lattice [12].

Along with elevation of computer CPU speeds, the security dominant parameter of a system becomes larger and larger. As an example, the bit-length of the modulus of the ElGamal system based on the discrete logarithm problem (DLP) is already up to 1024. Currently, there are four manners of decreasing the bit-length of the security dominant parameter and increasing the one-wayness of the trapdoor function.

The first manner is to transplant known cryptosystems to a complex algebraic system from a simple one — the elliptic curve analogue of ElGamal referable to elliptic curve cryptography (ECC) for example. By now, any effectual algorithm which can find out elliptic curve discrete logarithms in subexponential

* Manuscript first received Nov. 15, 2006 (version 1.0), revised Mar. 15, 2007, and last revised Oct. 30, 2010, the essence of which is final, nevertheless other rational analyses of ver. 2.2 are expected. For pragmatizing, refer to the successor of REESSE1+.

E-mail of the authors: sheenway@126.com (Shenghui Su); swlu@ustc.edu.cn (Shuwang Lü).

This research is supported by MOST with project 2007CB311100 and 2009AA01Z441.

time in the bit-length of a modulus has not been discovered yet [13].

Theoretically, almost every existing cryptosystem may have an elliptic curve analogue. However, not every analogue can bring the same effect as the analogue of ElGamal ─ the analogue of RSA of which the security still relies on the two large prime factors [14] for example.

The second manner is to design cryptosystems over polynomial rings ─ the NTRU system for example. The shortest vector problem (SVP) is the security bedrock of NTRU since it is impossible to seek a NTRU secret polynomial or a NTRU plaintext polynomial through the L^3 base reduction on condition that two special parameters c_h and c_m are fitly selected [15].

The third manner is to construct cryptosystems based on the tame automorphism of multivariate quadratic polynomials over a small field ─ the TTM scheme [16] and the TTS scheme [17] ordinarily referred to as the multivariate cryptosystems for example.

The fourth manner is to devise cryptosystems over small prime fields through finding or constructing novel computational problems which should be harder than IFP, DLP, or SSP with low density according to evidences. Some threads of this manner are given by the authors.

In the paper, ① the authors bring forward the three new computational problems: multivariate permutation problem (MPP) $C_i \equiv (A_i W^{\ell(i)})^{\delta}$ (% M), anomalous (modular) subset product problem (ASPP) $\bar{G} = \prod_{i=1}^{n} C_i^{b_i}$ (% M), and transcendental logarithm problem (TLP) $y \equiv x^x$ (% M), where M is a prime, and find some evidences which show that they are separately harder than DLP in the same prime field; ② over a prime field, design and analyze a prototypal public key cryptosystem called REESSE1+, including five algorithms for keys, encryption, and signature; ③ give the definition and properties of a coprime sequence and a lever function; ④ discuss the ineffectualness of the continued fraction attack; ⑤ propose and prove the double congruence theorem.

MPP which owns the indeterminacy assures the security of a private key, ASPP as a trapdoor function which can resist L^3 lattice base reduction assures the security of a ciphertext, and TLP which concerts with the root finding problem of a true polynomial $ax^n + bx^{n-1} + cx + d \equiv 0$ (% $\bar{M}$), and can withstand attacks by solving discrete logarithms assures the security of a signature. Provable security by reduction is appreciable, but not sufficient, and thus the exact security of REESSE1+ is analyzed.

It is not difficult to understand that REESSE1+ is essentially a multiproblem cryptosystem. The security of the multiproblem cryptosystem is equivalent to the complexity of what is easiest solved in all the problems. Additionally, MPP contains the four variables almost independent, and therefore, in a broad sense, REESSE1+ may be regarded as a multivariate cryptosystem. A multiproblem cryptosystem must be a multivariate cryptosystem because only multiple variables can bring multiple problems.

REESSE1+ is different from REESSE1 which is based on the subset product problem $\bar{G} = \prod_{i=1}^{n} C_i^{b_i}$ (% M), and does not have a robust signature scheme [18][19], and will make sense in untouched areas. We know that on a quantum computational model, IFP and DLP are already solved in polynomial time [20], and naturally, whether MPP, ASPP, and TLP can be solved in polynomial time on quantum computer is interesting. Besides, TLP as a primitive problem can not be converted into a discrete logarithm problem, which indicates that one can design other signature schemes over a small prime field by using TLP or its variant $y \equiv (gx)^x$ (% M).

Throughout the paper, unless otherwise specified, $n \ge 80$ is the bit-length of a plaintext block, or length of a sequence, the sign % means "modulo", $\bar{M}$ does "$M - 1$", log does a logarithm to the base 2, $\neg$ does the radix-minus-one complement of a bit, $\grave{P}$ does the maximal prime allowed in REESSE1+, $|x|$ denotes the size of a set x, $\|x\|$ denotes the order of x % M, and gcd(a, b) represents the greatest common divisor of two integers. Without ambiguity, "% M" is usually omitted in expressions.

2 A Coprime Sequence and a Lever Function

Definition 1*:* If $A_1, \ldots, A_n > 1$ are n pairwise distinct positive integers such that $\forall\, A_i, A_j$ with $A_i \ne A_j$, either gcd(A_i, A_j) = 1, or gcd(A_i, A_j) = $H \ne 1$ with $(A_i / H) \nmid A_k$ and $(A_j / H) \nmid A_k$ $\forall\, k \ne i, j \in [1, n]$, the integers are called a coprime (relatively prime) sequence, denoted by $\{A_1, \ldots, A_n\}$, and shortly $\{A_i\}$.

Property 1*:* If we randomly select $m \in [1, n]$ elements from $\{A_1, \ldots, A_n\}$ and construct a subsequence $\{A_{x_1}, \ldots, A_{x_m}\}$ (a subset as unordered), a subset product $G = \prod_{i=1}^{m} A_{x_i} = A_{x_1} \ldots A_{x_m}$ is uniquely determined, namely the mapping from G to $\{A_{x_1}, \ldots, A_{x_m}\}$ is one-to-one.

G is also called a coprime sequence product.

Proof: By contradiction.

Firstly, assume that $\forall\ A_i, A_j \in \{A_1, \ldots, A_n\}$, $\gcd(A_i, A_j) = 1$.

Because $A_1, \ldots, A_n$ are pairwise relatively prime, for arbitrary $A_j, A_k \in \{A_1, \ldots, A_n\}$, 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.

Presuppose that G is acquired from two different subsequences $\{Ax_1, \ldots, Ax_m\}$ and $\{Ay_1, \ldots, Ay_h\}$, namely

$$G = \prod_{i=1}^{m} Ax_i = Ax_1 \ldots Ax_m = \prod_{j=1}^{h} Ay_j = Ay_1 \ldots Ay_h.$$

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

Without loss of generality, let $A_z \in \{Ax_1, \ldots, Ax_m\}$ and $A_z \notin \{Ay_1, \ldots, Ay_h\}$.

In terms of the fundamental theorem of arithmetic [14], there must exist a prime number p which is the divisor of A_z.

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 $Ax_1 \ldots Ax_m$ but not the divisor of the product $Ay_1 \ldots Ay_h$. It means that the integer G has two distinct prime factorizations, which is in direct contradiction to the fundamental theorem of arithmetic.

Secondly, assume that $\exists\ A_i, A_j \in \{A_1, \ldots, A_n\}$ with $\gcd(A_i, A_j) \neq 1$.

According to definition 1, $\forall\ A_k \in \{A_1, \ldots, A_n\}$ with $k \neq i, j$, there are $(A_i / \gcd(A_i, A_j)) \nmid A_k$ and $(A_j / \gcd(A_i, A_j)) \nmid A_k$, which means at least one divisor of A_i and of A_j are not contained in any other elements.

Assume that $z = i$ or j, $A_z \in \{Ax_1, \ldots, Ax_m\}$, $A_z \notin \{Ay_1, \ldots, Ay_h\}$, $p \mid A_z$, and $p \nmid A_k\ \forall\ k \neq z \in [1, n]$, where p is a prime.

If $G = Ax_1 \ldots Ax_m = Ay_1 \ldots Ay_h$, then $p \mid (Ax_1 \ldots Ax_m)$, and $p \nmid (Ay_1 \ldots Ay_h)$, which is in direct contradiction.

Therefore, the mapping relation between G and $\{Ax_1, \ldots, Ax_m\}$ is one-to-one. □

***Property 2**:* Let $\{A_1, \ldots, A_n\}$ be a coprime sequence, and $b_1 \ldots b_n \neq 0$ be a bit string. Then the mapping from $G = \prod_{i=1}^{n} A_i^{\underline{b}_i}$ to $b_1 \ldots b_n$ is one-to-one, where $\underline{b}_i = 0$ if $b_i = 0$, 1 plus the number of successive 0-bits before b_i if $b_i = 1$, or 1 plus the number of successive 0-bits before and after b_i if b_i is the rightmost 1.

G is called an anomalous coprime sequence product.

Notice that there is an important fact: $\sum_{i=1}^{n} \underline{b}_n = n$.

Proof:

The proof is similar to that of property 1. □

Additionally, in the REESSE1+ system, the key transform is $C_i \equiv (A_i W^{\ell(i)})^{\delta}\ (\%\ M)$, where the exponent $\ell(i)$ is conceptualized.

***Definition 2**:* In a public key cryptosystem, $\ell(.)$ relevant to the key transform is called a lever function, if it has the following features:

- $\ell(.)$ is an injection from the domain $\{1, \ldots, n\}$ to the codomain $\Omega \subset \{1, \ldots, \bar{M}\}$. Let $Ł_n$ represent the collection of all injections from $[1, n]$ to $[1, \bar{M}]$, then $|Ł_n| \geq \mathrm{A}_n^n = n \times \ldots \times 1$.
- The mapping between i and $\ell(i)$ is established randomly without an analytical formula, so every time a public key is generated, $\ell(.)$ is distinct from one another, and there does not exist any dominant mapping from $\ell(.)$ to a public key.
- An attacker has to consider all the arrangements of elements in Ω 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 owner of a private key only considers the accumulative sum of elements in Ω 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 $\ell(.)$ at "a public terminal", and the small amount of calculation on $\ell(.)$ at "a private terminal".

3 Design of the REESSE1+ Public key Cryptosystem

In essence, REESSE1+ is a prototypal cryptosystem which is used for expounding some foundational definitions, concepts, ideas, thought, and methods. For its pragmatization, refer to the successor of REESSE1+.

3.1 The Key Generation Algorithm

This algorithm is employed by the owner of a key pair.

Let $\acute{p}_1, \dots, \acute{p}_n$ be the first n primes in the set $\mathbb{N}$, $\Lambda = \{2, 3, \dots, 1201\}$, and $\Omega = \{1, 3, \dots, 2n-1\}$, where Ω be an odd set.

Assume that đ, Đ, T, S are four pairwise coprime integers, where đ $\in [5, 2^{16}]$, and Đ, $T \geq 2^n$.

S1: Randomly generate coprime $A_1, \dots, A_n$ with $A_i \in \Lambda$.

S2: Find prime $M > (\max_{1 \le i \le n} A_i)^n$ making $\text{đ}\text{Đ}T \,|\, \bar{M}$, $\gcd(S, \bar{M}) = 1$,

and $\prod_{i=1}^{k} \acute{p}_i^{\bar{e}_i} \,|\, \bar{M}$, where k meets $\prod_{i=1}^{k} \bar{e}_i \geq 2^{10}$ and $\acute{p}_k \approx 2n$.

S3: Pick $W, \delta \in (1, \bar{M})$ making $\gcd(\delta, \bar{M}) = 1$, $\|\delta\| = \text{đ}\text{Đ}T$, and $\|W\| \geq 2^{n-20}$.

S4: Compute $\alpha \leftarrow \delta^{(\delta^n + \delta W^{n-1})T}$, $\beta \leftarrow \delta^{W^n T}$, $\hbar \leftarrow (W \prod_{i=1}^{n} A_i)^{-\delta S} (\alpha \delta^{-1})$ % M.

S5: Randomly produce pairwise distinct $\ell(1), \dots, \ell(n)$ with $\ell(i) \in \Omega$.

S6: Compute $C_i \leftarrow (A_i W^{\ell(i)})^{\delta}$ % M for $i = 1, \dots, n$.

At last, $(\{C_i\}, \alpha, \beta)$ is a public key, $(\{A_i\}, \{\ell(i)\}, W, \delta, \text{Đ}, \text{đ}, \hbar)$ is a private key, and S, T, M are in common.

Notice that the set Ω is not unique — $\Omega = \{\delta + 1, \delta + 2, \dots, \delta + n\}$ for example, and the primary principle for selecting Ω is that a ciphertext is decrypted as quickly as possible.

At S3, to seek δ, first let $\delta \equiv g^{\bar{M}/(\text{đ}\text{Đ}T)}$ (% M), where g is a generator by algorithm 4.80 in section 4.6 of [21], then test δ.

At S4, seeking a S-th root to $x^S \equiv c$ (% M) is referred to theorem 1 in section 3.4.

3.2 The Encryption Algorithm

Assume that $(\{C_i\}, \alpha, \beta)$ is a public key, and $b_1 \dots b_n \neq 0$ is a bit plaintext block or a symmetric key.

S1: Set $\bar{G} \leftarrow 1$, $k \leftarrow 0$, $i \leftarrow 1$.

S2: If $b_i = 0$, let $k \leftarrow k + 1$, $\underline{b}_i \leftarrow 0$;

else let $\underline{b}_i \leftarrow k + 1$, $k \leftarrow 0$, $\bar{G} \leftarrow \bar{G} C_i^{\underline{b}_i}$ % M.

S3: Let $i \leftarrow i + 1$.

If $i \leq n$, go to S2.

S4: If $b_n = 0$, let $\underline{b}_{n-k} \leftarrow \underline{b}_{n-k} + k$, $\bar{G} \leftarrow \bar{G} C_{n-k}^{\ k}$ % M.

At last, the ciphertext $\bar{G} \equiv \prod_{i=1}^{n} C_i^{\underline{b}_i}$ (% M) called an anomalous subset product is obtained.

Notice that α and β are unuseful for the encryption.

3.3 The Decryption Algorithm

Assume that $(\{A_i\}, \{\ell(i)\}, W, \delta, \text{Đ}, \text{đ}, \hbar)$ is a related private key, and $\bar{G}$ is a ciphertext.

Notice that because $\sum_{i=1}^{n} \underline{b}_i = n$ is even, $\sum_{i=1}^{n} \underline{b}_i \ell(i)$ must be even.

S1: Compute $\bar{G} \leftarrow \bar{G}^{\delta^{-1}}$ % M.

S2: Compute $\bar{G} \leftarrow \bar{G} W^{-2}$ % M.

S3: Set $b_1 \dots b_n \leftarrow 0$, $G \leftarrow \bar{G}$, $i \leftarrow 1$, $k \leftarrow 0$.

S4: If $A_i^{k+1} \,|\, G$, let $G \leftarrow G / A_i^{k+1}$, $b_i \leftarrow 1$, $k \leftarrow 0$;

else let $k \leftarrow k + 1$.

S5: Let $i \leftarrow i + 1$.

If $i \leq n$ and $G \neq 1$, go to S4.

S6: If $k \neq 0$ and $(A_{n-k})^k \,|\, G$, let $G \leftarrow G / (A_{n-k})^k$.

S7: If $G \neq 1$, go to S2; else end.

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

This algorithm can always terminate normally as long as $\bar{G}$ is a true ciphertext. In decryption, Đ and đ are unhelpful.

3.4 The Digital Signature Algorithm

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

S1: Let $H \leftarrow hash(F)$, whose binary form is $b_1 \dots b_n$.

S2: Set $\underline{k} \leftarrow \delta \sum_{i=1}^{n} b_i \ell(i)$ % $\bar{M}$, $G_0 \leftarrow (\prod_{i=1}^{n} A_i^{\neg b_i})^{\delta}$ % M.

S3: $\forall\, \bar{a} \in (1, \bar{\boldsymbol{M}})$ making $\text{đ}T \nmid \bar{a}$, $\text{đ} \nmid WQ \,\%\, \bar{\boldsymbol{M}}$, where $Q \equiv (\bar{a}\text{Đ} + WH)\delta^{-1} \,\%\, \bar{\boldsymbol{M}}$.

S4: Compute $R \leftarrow (Q(\delta\hbar)^{-1})^{1/S} G_0^{-1}$, $\bar{U} \leftarrow (RW^{k-\delta})^Q$,

$\hat{g} \leftarrow \delta^{\bar{a}\text{Đ}} \,\%\, M$, $\xi \leftarrow \sum_{i=0}^{n-1} (\delta Q)^{n-1-i}(HW)^i \,\%\, \bar{\boldsymbol{M}}$.

S5: $\forall\, r \in [1, \text{đ}2^{16}]$ making $\text{đ} \nmid (rUS + \xi) \,\%\, \bar{\boldsymbol{M}}$, where $U \equiv \bar{U}\hat{g}^r \,(\%\, M)$.

S6: If $\text{đ} \nmid ((WQ)^{n-1} + \xi + rUS) \,\%\, \bar{\boldsymbol{M}}$, go to S5.

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

It is known from S3, S4 that Q, R meet $\bar{a}\text{Đ} \equiv \delta Q - WH \,(\%\, \bar{\boldsymbol{M}})$ and $Q \equiv (RG_0)^S \delta\hbar \,(\%\, M)$.

It should be noted that owing to $\text{đ} \nmid \bar{a}$, $\gcd(\text{Đ}, \text{đ}) = 1$, and $\text{đ} \mid \bar{\boldsymbol{M}}$, there must exist $\text{đ} \nmid (\delta Q - WH)$.

According to the double congruence theorem (see section 3.6), in the signature algorithm we do not need $V \equiv (R^{-1}W^{\delta}G_1)^{QU}\delta^{\lambda} \,(\%\, M)$, where $G_1 = (\prod_{i=1}^{n} A_i^{b_i})^{\delta} \,\%\, M$, and λ satisfies

$$\lambda S \equiv ((WQ)^{n-1} + \xi + rUS)(\delta Q - HW)\,(\%\, \bar{\boldsymbol{M}}),$$

which indicates $\text{đ}\text{Đ} \mid \lambda$.

At S5, the probability of finding a fit U is roughly $1 / \text{đ}$. Since đ is a small number, which does not influence the security of REESSE1+ (see section 6.2), U can be found out at a good pace.

Let $\Delta \equiv (WQ)^{n-1} + \xi + rUS \,(\%\, \bar{\boldsymbol{M}})$.

Due to $\text{đ} \mid \bar{\boldsymbol{M}}$, if $((WQ)^{n-1} + \xi + rUS)$ contains the factor đ, it must be contained in $\Delta \,\%\, \bar{\boldsymbol{M}}$.

Besides, due to $\text{đ} \nmid S$ and $\text{đ} \nmid (WQ)^{n-1}$ (by $\text{đ} \nmid WQ$), if we want to make $\text{đ} \mid \Delta$, there must be $\text{đ} \nmid (rUS + \xi) \,\%\, \bar{\boldsymbol{M}}$.

Therefore, as long as every value of r makes rUS different, $\text{đ} \mid \Delta$ will holds after about đ attempts of r. The algorithm can also terminate normally because after r traverses the interval $[1, \text{đ}2^{16}]$, the probability of $\text{đ} \nmid \Delta$ is $(1 - 1/\text{đ})^{\text{đ}2^{16}}$, and almost zero.

At S4, we derive ξ from $\xi(\delta Q - WH) \equiv (\delta Q)^n - (WH)^n \,(\%\, \bar{\boldsymbol{M}})$. Computing R by $Q \equiv (RG_0)^S \delta\hbar \,(\%\, M)$ may resort to theorem 1, where S meets $\gcd(S, \bar{\boldsymbol{M}}) = 1$.

Theorem 1: For the congruence $x^n \equiv c \,(\%\, M)$ with M being prime, if $\gcd(n, \bar{\boldsymbol{M}}) = 1$, every c has just one n-th root modulo $\bar{\boldsymbol{M}}$. Especially, let μ satisfy $\mu n \equiv 1 \,(\%\, \bar{\boldsymbol{M}})$, then $c^{\mu} \,\%\, M$ is the n-th root.

Further, we have theorem 2 and 3.

Theorem 2: For the congruence $x^n \equiv c \,(\%\, M)$ with M being prime, if $n \mid \bar{\boldsymbol{M}}$ and $\gcd(n, \bar{\boldsymbol{M}} / n) = 1$, then when c is an n-th power residue modulo M, and μ satisfies $\mu n \equiv 1 \,(\%\, \bar{\boldsymbol{M}}/n)$, $c^{\mu} \,\%\, M$ is an n-th root.

Theorem 3: For the congruence $x^n \equiv c \,(\%\, M)$ with M being prime, if $n \nmid \bar{\boldsymbol{M}}$, let $k = \gcd(n, \bar{\boldsymbol{M}})$, and μ satisfy $\mu(n/k) \equiv 1 \,(\%\, \bar{\boldsymbol{M}} / k)$, then $x^n \equiv c \,(\%\, M)$ is equivalent to

$$x^k \equiv c^{\mu} \,(\%\, M),$$

that is, the two congruences have the same set of solutions.

The proofs of theorem 1 and 2 are referred to [22], and the proof of theorem 3 is referred to [23]. The solution which is obtained by theorem 1 and theorem 2, and may be written as a certain power of c modulo p is called the trivial solution to the congruence $x^n \equiv c \,(\%\, M)$ [23].

3.5 The Identity Verification Algorithm

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

S1: Let $H \leftarrow hash(F)$, whose binary form is $b_1 \ldots b_n$.

S2: Compute $\bar{G} \leftarrow \prod_{i=1}^{n} C_i^{b_i} \,\%\, M$.

S3: Compute $X \leftarrow (\alpha Q^{-1})^{QUT} \alpha^{Q^n} \,\%\, M$,

$Y \leftarrow (\bar{G}^Q U^{-1})^{UST} \beta^{HQ^{n-1} + H^n} \,\%\, M$.

S4: If $X \equiv Y$, the identity is valid and F intact;

else the identity is invalid or F modified.

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

The discriminant $X \equiv Y \,(\%\, M)$ at S4 is argued as follows:

It is known from section 3.1 that $\alpha \equiv \delta^{(\delta^n + \delta W^{n-1})T} \equiv \delta\hbar(W^{\delta}G_0G_1)^S \,(\%\, M)$ and $\beta \equiv \delta^{W^nT} \,(\%\, M)$.

Let $V \equiv (R^{-1}W^{\delta}G_1)^{QU}\delta^{\lambda} \,(\%\, M)$.

Considering λ meeting $\lambda S \equiv ((WQ)^{n-1} + \xi + rUS)(\delta Q - HW)\,(\%\, \bar{\boldsymbol{M}})$, let $\lambda = k\text{đ}\text{Đ}$, where k is a integer, and then

$$\begin{aligned} Q^{QU}V^S &\equiv (RG_0)^{SQU}(\delta\hbar)^{QU}(R^{-1}W^{\delta}G_1)^{QUS}\delta^{\lambda S} \\ &\equiv (W^{\delta}G_0G_1)^{QUS}(\delta\hbar)^{QU}\delta^{\lambda S} \end{aligned}$$

$$\begin{aligned}&\equiv \alpha^{QU}\,\delta^{((WQ)^{n-1}+\Sigma_{i=0}^{n-1}(\delta Q)^{n-1-i}(WH)^{i}+rUS)(\delta Q-WH)}\\ &\equiv \alpha^{QU}\,\delta^{\delta W^{n-1}Q^{n}-W^{n}HQ^{n-1}+(\delta Q)^{n}-(WH)^{n}+(\delta Q-WH)rUS}\\ &\equiv \alpha^{QU}\,\delta^{(\delta^{n}+\delta W^{n-1})Q^{n}}\,\delta^{-W^{n}(HQ^{n-1}+H^{n})}\,\delta^{\bar{a}\text{Đ}rUS}\,(\% M).\end{aligned}$$

Transposition yields

$$V^{S} \equiv (\alpha Q^{-1})^{QU}\,\delta^{(\delta^{n}+\delta W^{n-1})Q^{n}}\,\delta^{-W^{n}(HQ^{n-1}+H^{n})}\,\delta^{\bar{a}\text{Đ}rUS}\,(\% M).$$

Therefore, we have

$$\begin{aligned}V^{ST} &\equiv (\alpha Q^{-1})^{QUT}\,\delta^{(\delta^{n}+\delta W^{n-1})TQ^{n}}\,\delta^{-TW^{n}(HQ^{n-1}+H^{n})}\,\delta^{\bar{a}\text{Đ}rUST}\\ &\equiv (\alpha Q^{-1})^{QUT}\,\alpha^{Q^{n}}\,\beta^{-(HQ^{n-1}+H^{n})}\,\delta^{\bar{a}\text{Đ}rUST}\\ &\equiv X\beta^{-(HQ^{n-1}+H^{n})}\,\delta^{\bar{a}\text{Đ}rUST}\,(\% M).\end{aligned}$$

In addition,

$$\begin{aligned}U^{UT}V^{T} &\equiv (RW^{\underline{k}-\delta})^{QUT}(\delta^{\bar{a}\text{Đ}r})^{UT}(R^{-1}W^{\delta}G_1)^{QUT}\delta^{\lambda T}\\ &\equiv (W^{\underline{k}}G_1)^{QUT}\,\delta^{\bar{a}\text{Đ}rUT}\,\delta^{\lambda T}\\ &\equiv \bar{G}^{QUT}\,\delta^{\bar{a}\text{Đ}rUT}\,\delta^{k\text{đ}\text{Đ}T}\\ &\equiv \bar{G}^{QUT}\,\delta^{\bar{a}\text{Đ}rUT}\,(\% M).\end{aligned}$$

Transposition yields

$$V^{T} \equiv (\bar{G}^{Q}\,U^{-1})^{UT}\,\delta^{\bar{a}\text{Đ}rUT}\,(\% M).$$

Hence

$$V^{ST} \equiv (\bar{G}^{Q}\,U^{-1})^{UST}\,\delta^{\bar{a}\text{Đ}rUST}\,(\% M).$$

By the double congruence theorem (theorem 4), there is

$$\begin{aligned}V^{ST} &\equiv X\beta^{-(HQ^{n-1}+H^{n})}\,\delta^{\bar{a}\text{Đ}rUST}\\ &\equiv (\bar{G}^{Q}U^{-1})^{UST}\,\delta^{\bar{a}\text{Đ}rUST}\,(\% M).\end{aligned}$$

Namely, $X \equiv (\bar{G}^{Q}U^{-1})^{UST}\beta^{HQ^{n-1}+H^{n}} \equiv Y\,(\% M)$.

3.6 The Double Congruence Theorem

Theorem 4 **(*The Double Congruence Theorem*)***:* Assume that M is a prime, and that s and t satisfying $\gcd(s, t) = 1$ are two constants, then simultaneous equations

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

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

Proof: Necessity:

Assume that the simultaneous equations $x^{s} \equiv a$ (% M) and $x^{t} \equiv b$ (% M) have solutions.

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

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

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

Sufficiency:

Assume that $a^{t} \equiv b^{s}$ (% M).

In terms of the greatest common divisor theorem [14], there exists a pair of integers u and v making $u s + vt = 1$. Thus,

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

The above two equations multiplying yields

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

Furthermore, we have

$$(a^{u}\,b^{v})^{s} \equiv a^{us}b^{vs} \equiv a^{us}a^{vt} \equiv a^{us+vt} \equiv a\ (\% M),$$
$$(a^{u}\,b^{v})^{t} \equiv a^{ut}b^{vt} \equiv b^{us}b^{vt} \equiv b^{us+vt} \equiv b\ (\% M).$$

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

Uniqueness:

Let $x_0 \equiv a^{u}\,b^{v}$ (% M).

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

Then, it holds that

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

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

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

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

If neither s nor t is coprime to $\bar{M}$, may let $k = \gcd(s, \bar{M})$, $h = \gcd(t, \bar{M})$. Then we see $\gcd(s/k, \bar{M}) = 1$ and $\gcd(t/h, \bar{M}) = 1$.

Thus, there are $(x_0 x_1^{-1})^k \equiv 1$ and $(x_0 x_1^{-1})^h \equiv 1$ (% M). By theorem 3 and $\gcd(s, t) = 1$, we know $\gcd(k, h) = 1$. In terms of the group theory [24], when $\gcd(k, h) = 1$, only the element '1' belongs to two different subgroup at the 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 4. □

3.7 Characteristics of REESSE1+

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

- The security of REESSE1+ is not based on a single hardness, but on three hardnesses: MPP, ASPP, and TLP. Hence, it is a multiproblem public key cryptosystem.
- The key transform $C_i \equiv (A_i W^{\ell(i)})^{\delta}$ (% M) for $i = 1, \ldots, n$ contain $2n+2$ unknown variables, and each equation contains four almost independent variables. Hence, REESSE1+ is multivariate.
- 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 $\ell(i)$.
- The indeterminacy of $\ell(.)$ as $\delta = 1$. If 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. If C_i and A_i are determined, W and $\ell(i)$ can not be determined, and also have no one-to-one relation for $\gcd(\ell(i), \bar{M}) > 1$.
- The insufficiency of the mapping. A private key includes $\{A_i\}$, $\{\ell(i)\}$, W, δ etc, but there is only a dominant mapping from $\{A_i\}$ to $\{C_i\}$. Thus, the reversibility of the function is poor.
- Because combinations among multiple variables may bring different hardnesses, REESSE1+ is a self-improvable system while its main architecture need not be changed.

3.8 Correctness of the Decryption Algorithm

Since $(\mathbb{Z}_M^*, \cdot)$ is an Abelian, namely commutative group, $\forall \underline{k} \in [1, \bar{M}]$, there is

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

Let $b_1 \ldots b_n$ be an n-bit plaintext.

It is known from section 3.2 that $\bar{G} \equiv \prod_{i=1}^{n} C_i^{\underline{b}_i}$ (% M), where $\underline{b}_i$ means what the algorithm shows, and $C_i \equiv (A_i W^{\ell(i)})^{\delta}$ % M.

Let $G \equiv \prod_{i=1}^{n} A_i^{\underline{b}_i}$ (% M), and $\underline{k} = \sum_{i=1}^{n} \ell(i) \underline{b}_i$.

Then, ***we need to prove*** that $\bar{G}^{\delta^{-1}} (W^{-1})^{\underline{k}} \equiv G$ (% M).

Proof:

According to the key generator and encryption algorithm, there is

$$\begin{aligned} \bar{G} &\equiv \prod_{i=1}^{n} C_i^{\underline{b}_i} \equiv \prod_{i=1}^{n/2} ((A_i W^{\ell(i)})^{\delta})^{\underline{b}_i} \\ &\equiv W^{(\sum_{i=1}^{n} \underline{b}_i \ell(i))\delta} \prod_{i=1}^{n} (A_i)^{\delta \underline{b}_i} \\ &\equiv W^{\underline{k}\delta} (\prod_{i=1}^{n} A_i^{\underline{b}_i})^{\delta} \ (\%\ M). \end{aligned}$$

Further, raising either side of the above equation to the δ^{-1}-th yields

$$\begin{aligned} \bar{G}^{\delta^{-1}} &\equiv (W^{\underline{k}\delta} (\prod_{i=1}^{n} A_i^{\underline{b}_i})^{\delta})^{\delta^{-1}} \\ &\equiv W^{\underline{k}} \prod_{i=1}^{n} A_i^{\underline{b}_i} \ (\%\ M). \end{aligned}$$

Multiplying either side of the just above equation by $(W^{-1})^{\underline{k}}$ yields

$$\begin{aligned} \bar{G}^{\delta^{-1}} (W^{-1})^{\underline{k}} &\equiv W^{\underline{k}} \prod_{i=1}^{n} A_i^{\underline{b}_i} (W^{-1})^{\underline{k}} \\ &\equiv W^{\underline{k}} \prod_{i=1}^{n} A_i^{\underline{b}_i} (W^{\underline{k}})^{-1} \\ &\equiv \prod_{i=1}^{n} A_i^{\underline{b}_i} \equiv G \ (\%\ M). \end{aligned}$$

Clearly, the above process also gives a method of seeking G meantime. □

Notice that in practice, $b_1 \ldots b_n$ is unknowable in advance, so we have no way to directly compute $\underline{k}$. However, because the range of $\underline{k} \in (n, n(2n - 1))$ is very narrow, we may search $\underline{k}$ heuristically by multiplying W^{-2}, and verify whether $G = 1$ after it is divided exactly by some $A_i^{\underline{b}_i}$. It is known from section 3.3 that the original $b_1 \ldots b_n$ is acquired at the same time the condition $G = 1$ is satisfied.

3.9 Uniqueness of a Plaintext Solution to a Ciphertext

Because $\{C_1, \ldots, C_n\}$ is a non-coprime sequence, the mapping from $\prod_{i=1}^{n} C_i^{\underline{b}_i} \% M$ to $\bar{G}$ (see section 3.2) is theoretically many-to-one. It might possibly result in the nonuniqueness of a plaintext solution $b_1 \ldots b_n$ when $\bar{G}$ is being unveiled.

Suppose that the ciphertext $\bar{G}$ can be obtained from two different anomalous subset products corresponding to $b_1 \ldots b_n$ and $b'_1 \ldots b'_n$ respectively. Then,

$$\bar{G} \equiv \prod_{i=1}^{n} C_i^{\underline{b}_i} \equiv \prod_{i=1}^{n} C_i^{\underline{b}'_i} \ (\% \ M).$$

That is,

$$\prod_{i=1}^{n} (A_i W^{\ell(i)})^{\delta \underline{b}_i} \equiv \prod_{i=1}^{n} (A_i W^{\ell(i)})^{\delta \underline{b}'_i} \ (\% \ M).$$

Further, there is

$$W^{\underline{k}\delta} \prod_{i=1}^{n} (A_i)^{\delta \underline{b}_i} \equiv W^{\underline{k}'\delta} \prod_{i=1}^{n} (A_i)^{\delta \underline{b}'_i} \ (\% \ M),$$

where $\underline{k} = \sum_{i=1}^{n} \underline{b}_i \ell(i)$, and $\underline{k}' = \sum_{i=1}^{n} \underline{b}'_i \ell(i) \ \% \ \bar{M}$.

Raising either side of the above congruence to the δ^{-1}-th power yields

$$W^{\underline{k}} \prod_{i=1}^{n} A_i^{\underline{b}_i} \equiv W^{\underline{k}'} \prod_{i=1}^{n} A_i^{\underline{b}'_i} \ (\% \ M).$$

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

$$W^{\underline{k} - \underline{k}'} \equiv \prod_{i=1}^{n} A_i^{\underline{b}'_i} (\prod_{i=1}^{n} A_i^{\underline{b}_i})^{-1} \ (\% \ M).$$

Let $\theta \equiv \prod_{i=1}^{n} A_i^{\underline{b}'_i} (\prod_{i=1}^{n} A_i^{\underline{b}_i})^{-1} \ (\% \ M)$, namely $\theta \equiv W^{\underline{k} - \underline{k}'} \ (\% \ M)$.

This congruence signifies when the plaintext $b_1 \ldots b_n$ is not unique, the value of W must be relevant to θ. The contrapositive assertion equivalent to it is that if the value of W is irrelevant to θ, $b_1 \ldots b_n$ will be unique. Thus, we need to consider the probability that W 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 all the potential values. Hence, currently 80 bits are quite enough for the security of a symmetric key.

$b_1 \ldots b_n$ contains n bits which indicates $\prod_{i=1}^{n} A_i^{\underline{b}_i}$ has 2^n potential values, and thus the number of potential values of θ is at most $2^n \times 2^n$. Notice that because $A_1^{-1}, \ldots, A_n^{-1}$ are not necessarily coprime, some values of θ may possibly occur repeatedly.

Because $|\underline{k} - \underline{k}'| \leq n(2n - 1) \leq 32640 \approx 2^{15}$ with $n \leq 128$, and W has at most 2^{15} solutions to every θ, the probability that W takes a value relevant to θ is at most $2^{15} 2^{2n} / M$. When $n \geq 80$, there is $2^{15} 2^{2n} / M \leq 2^{175} / 2^{553} = 1 / 2^{378}$ which is close to zero (the product of the first 80 primes is about 2^{553}). The probability will further decrease when W is a prime since the solutions to θ lean to being composite integers in the average case.

In addition, if you please, resorting to $\sum_{i=1}^{n} \underline{b}_i = n$, may exclude some unoriginal plaintext solutions.

4 Necessity and Sufficiency of the Lever Function for Resisting Continued Fraction Attacks

To discuss the effect of the lever function $\ell(.)$ which is an injective, let $\delta = 1$, and consider a special key transform

$$C_i \equiv A_i W^{\ell(i)} \ (\% \ M) \text{ for } i = 1, \ldots, n,$$

where each $\ell(i) \in \Omega$ is pairwise distinct, and $\Omega = \{5, 7, \ldots, 19, 53, 55, \ldots\}$ is an known odd set of $2n$ elements such that ① $\forall\ e_1, e_2 \in \Omega$, $e_1 \neq e_2$; ② $\forall\ e_1, e_2, e_3 \in \Omega$, $e_1 + e_2 \neq e_3$; ③ $\forall\ e_1, e_2, e_3, e_4 \in \Omega$, $e_1 + e_2 + e_3 \neq e_4$. The above transform is called a slack transform since $\sum_{i=1}^{n} \ell(i)$ is relatively large.

The production of Ω is referred to appendix A: a program in C++ with the running time of $O(n^2)$. The maximal element in Ω is 2652, 3212, 3736, and 4260 respectively when n = 80, 96, 112, and 128.

Theorem 5: If α is an irrational number, and r/s is a rational in lowest terms such that $|\alpha - r/s| < 1/(2s^2)$, where $|\alpha - r/s|$ is an absolute value, and $r, s > 0$ are two integers, then r/s is a convergent of the simple continued fraction expansion of α [25].

The proof of theorem 5 is referred to [25].

If α is a rational number, theorem 5 also holds [25].

4.1 Necessity of the Lever Function $\ell(.)$

If a private key is insecure, a plaintext must be insecure. Therefore, the security of a private key is most foundational.

The necessity of the lever function $\ell(.)$ means that if a REESSE1+ private key is secure, $\ell(.)$ must exist in the slack transform. The equivalent contrapositive assertion is that if $\ell(.)$ does not exist or is a constant in the slack transform, the REESSE1+ private key will be insecure.

Assume that $\ell(.)$ is a unknown constant k. Then the slack transform becomes as

$$C_i \equiv A_i W^k \ (\%\ M)$$

which is equivalent to $\ell(.)$ being ineffectual or nonexistent.

Since $(\mathbb{Z}_M^*, \cdot)$ is an Abelian group [24], of course, there is

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

$\forall x \in [1, n-1]$, let

$$G_z \equiv C_x C_n^{-1}.$$

Substituting $A_x W^k$ and $A_n W^k$ respectively for C_x and C_n in G_z yields

$$G_z \equiv A_x W^k (A_n W^k)^{-1} \ (\%\ M)$$
$$A_n G_z \equiv A_x \ (\%\ M)$$
$$A_n G_z - L M = A_x,$$

where L is a positive integer.

The either side of the equation is divided by $A_n M$ gives

$$G_z / M - L / A_n = A_x / (A_n M). \tag{1}$$

Due to $M > (\max_{1\le i\le n} A_i)^n > \prod_{i=1}^{n} A_i$ and $A_i \ge 2$, there is

$$G_z / M - L / A_n < A_x / (A_n \prod_{i=1}^{n} A_i) = A_x / (A_n^2 \prod_{i=1}^{n-1} A_i) \le 1 / (2^{n-2} A_n^2),$$

that is,

$$G_z / M - L / A_n < 1 / (2^{n-2} A_n^2). \tag{2}$$

Evidently, as $n > 2$, there is

$$G_z / M - L / A_n < 1 / (2 A_n^2). \tag{2$'$}$$

In terms of theorem 5, L / A_n is a convergent of the continued fraction of G_z / M.

Thus, L / A_n and A_n may be determined by (2′) in polynomial time for the length of the continued fraction will not exceed $\lceil \log M \rceil$. Further, $W^k \equiv C_n A_n^{-1}$ (% M) may be computed. Therefore, the original coprime sequence $\{A_1, \ldots, A_n\}$ with $A_i \le \boldsymbol{Ᵽ}$ can almost be recovered.

Because W in every C_i has the same power, and the exponent of W in any $C_x C_n^{-1}$ is always zero, when $\ell(i)$ is the constant k, there does not exist the indeterministic reasoning problem. Besides, when a convergent of the continued fraction of G_z / M satisfies (2′), the subsequent some convergents also possibly satisfies (2′). If so, it will bring about the nonuniqueness of A_n.

The above analysis manifests that when the lever function $\ell(.)$ is the constant k, a related private key can be deduced from a public key, and further a related plaintext can be inferred from a ciphertext. Hence, the lever function $\ell(.)$ is necessary to the security of a private key and a plaintext block.

4.2 Ineffectualness of the Continued Fraction Attack

When the lever function $\ell(.)$ exists, we have again

$$C_i \equiv A_i W^{\ell(i)} \ (\%\ M).$$

In this case, $\ell(.)$ brings attackers at least two difficulties:

- No method by which one can directly judge whether the power of W in $C_{x_1} \ldots C_{x_m}$ is counteracted by the power of W^{-1} in $(C_{y_1} \ldots C_{y_h})^{-1}$.
- No criterion by which the presumption of an indeterministic reasoning can be verified in polynomial time.

The indeterministic reasoning based on continued fractions means that first presume that the parameter W and its inverse W^{-1} counteract each other, and then judge whether the presumption holds or not by the consequence.

According to section 4.1, first select $m \in [1, n-1]$ elements and $h \in [1, n-m]$ other elements from $\{C_i\}$. Let

$$G_x \equiv C_{x_1} \ldots C_{x_m} \ (\%\ M),$$
$$G_y \equiv C_{y_1} \ldots C_{y_h} \ (\%\ M),$$

where $C_{x_i} \ne C_{y_j}$ for $i \in [1, m]$ and $j \in [1, h]$.

Let

$$G_z \equiv G_x G_y^{-1} \ (\%\ M).$$

Since $\{\ell(1), \ldots, \ell(n)\}$ is any arbitrary arrangement of n elements of Ω, it is impossible to predicate that

G_z does not contain the factor W or W^{-1}. For a further deduction, we have to *presuppose* that W of G_x is exactly neutralized by W^{-1} of ${G_y}^{-1}$, and then,

$$G_z \equiv (A_{x_1} \ldots A_{x_m})(A_{y_1} \ldots A_{y_h})^{-1} \ (\%\ M)$$
$$G_z\,(A_{y_1} \ldots A_{y_h}) \equiv A_{x_1} \ldots A_{x_m} \ (\%\ M)$$
$$G_z\,(A_{y_1} \ldots A_{y_h}) - L\,M = A_{x_1} \ldots A_{x_m}$$
$$G_z / M - L / (A_{y_1} \ldots A_{y_h}) = (A_{x_1} \ldots A_{x_m}) / (M A_{y_1} \ldots A_{y_h}),$$

where L is a positive integer.

Denoting the product $A_{y_1} \ldots A_{y_h}$ by $\bar{A}_y$ yields

$$G_z / M - L / \bar{A}_y = (A_{x_1} \ldots A_{x_m}) / (M \bar{A}_y). \tag{3}$$

Due to $M > (\max_{1 \le i \le n} A_i)^n > \prod_{i=1}^{n} A_i$ and $A_i \ge 2$, we have

$$G_z / M - L / \bar{A}_y < 1 / (2^{n-m-h} {\bar{A}_y}^2). \tag{4}$$

Obviously, when $n > m + h$, (4) may have a variant, namely

$$G_z / M - L / \bar{A}_y < 1 / (2 {\bar{A}_y}^2). \tag{4$'$}$$

When $n = m + h$, if $M > (\max_{1 \le i \le n} A_i)^n$, (4′) still holds.

Especially, when $n > 3$, $h = 1$, $m = 2$, there exists

$$G_z / M - L / A_{y_1} < 1 / (2^{n-3} {A_{y_1}}^2) < 1 / (2 {A_{y_1}}^2). \tag{4$''$}$$

***Property 3**:* Let $h + m \le n$. If $\ell(x_1) + \ldots + \ell(x_m) = \ell(y_1) + \ldots + \ell(y_h)$, the subset product $\bar{A}_y = A_{y_1} \ldots A_{y_h}$ in (4′) will be found.

Proof:

$\ell(x_1) + \ldots + \ell(x_m) = \ell(y_1) + \ldots + \ell(y_h)$ means that a power of W in $C_{x_1} \ldots C_{x_m}$ is neutralized by a power of W^{-1} in $(C_{y_1} \ldots C_{y_h})^{-1}$, and thus (4′) holds.

In terms of theorem 5, $L / \bar{A}_y$ is inevitably a convergent of the continued fraction of G_z / M, and thus $\bar{A}_y = A_{y_1} \ldots A_{y_h}$ is found. □

Notice that (4′) is insufficient for $\ell(x_1) + \ldots + \ell(x_m) = \ell(y_1) + \ldots + \ell(y_h)$ (see property 7), and $\bar{A}_y$ is faced with nonuniqueness because there may possibly exist several convergents of the continued fraction of G_z / M which satisfy (4′).

***Property 4**:* Let $h + m \le n$. $\forall x_1, \ldots, x_m, y_1, \ldots, y_h \in [1, n]$, when $\ell(x_1) + \ldots + \ell(x_m) \ne \ell(y_1) + \ldots + \ell(y_h)$,

① there always exist

$$C_{x_1} \equiv A'_{x_1} W'^{\,\ell'(x_1)}, \ldots, C_{x_m} \equiv A'_{x_m} W'^{\,\ell'(x_m)},$$
$$C_{y_1} \equiv A'_{y_1} W'^{\,\ell'(y_1)}, \ldots, C_{y_h} \equiv A'_{y_h} W'^{\,\ell'(y_h)} \ (\%\ M),$$

such that $\ell'(x_1) + \ldots + \ell'(x_m) \equiv \ell'(y_1) + \ldots + \ell'(y_h)\ (\%\ \bar{M})$ with $A'_{y_1} \ldots A'_{y_h} \le \mathbf{Þ}^h$;

② $C_{x_1}, \ldots, C_{x_m}, C_{y_1}, \ldots, C_{y_h}$ make (4) with $A'_{y_1} \ldots A'_{y_h} \le \mathbf{Þ}^h$ hold with non-negligible probability.

Proof:

Because $A'_{y_1} \ldots A'_{y_h}$ need be scaled, the constraint $A'_{y_1} \ldots A'_{y_h} \le \mathbf{Þ}^h$ is demanded while because $A'_{x_1}, \ldots, A'_{x_m}$ need not be scaled, $A'_{x_1} \le \mathbf{Þ}, \ldots, A'_{x_m} \le \mathbf{Þ}$ are not demanded.

① Let $\bar{O}_d$ be an oracle for a discrete logarithm.

Suppose that $W' \in [1, \bar{M}]$ is a generator of $(\mathbb{Z}^*_M, \cdot)$.

Let $\mu = \ell'(y_1) + \ldots + \ell'(y_h)$. In terms of group theories, $\forall A'_{y_1}, \ldots, A'_{y_h} \in [2, \mathbf{Þ}]$ which need not be coprime, the equation

$$C_{y_1} \ldots C_{y_h} \equiv A'_{y_1} \ldots A'_{y_h} W'^{\,\mu} \ (\%\ M)$$

for μ has a solution. μ may be obtained through $\bar{O}_d$.

$\forall\ \ell'(y_1), \ldots, \ell'(y_{h-1}) \in [1, \bar{M}]$, and let $\ell'(y_h) \equiv \mu - (\ell'(y_1) + \ldots + \ell'(y_{h-1}))\ (\%\ \bar{M})$.

Similarly, $\forall\ \ell'(x_1), \ldots, \ell'(x_{m-1}) \in [1, \bar{M}]$, and let $\ell'(x_m) \equiv \mu - (\ell'(x_1) + \ldots + \ell'(x_{m-1}))\ (\%\ \bar{M})$.

Further, from $C_{x_1} \equiv A'_{x_1} W'^{\,\ell'(x_1)}, \ldots, C_{x_m} \equiv A'_{x_m} W'^{\,\ell'(x_m)}\ (\%\ M)$, we can obtain a tuple $\langle A'_{x_1}, \ldots, A'_{x_m} \rangle$, where $A'_{x_1}, \ldots, A'_{x_m} \in (1, M)$, and $\ell'(x_1) + \ldots + \ell'(x_m) \equiv \ell'(y_1) + \ldots + \ell'(y_h)\ (\%\ \bar{M})$.

Thus, property 4.1 is proven.

② Let $G_z \equiv C_{x_1} \ldots C_{x_m} (C_{y_1} \ldots C_{y_h})^{-1}\ (\%\ M)$. Then,

$$C_{x_1} \ldots C_{x_m} (C_{y_1} \ldots C_{y_h})^{-1} \equiv A'_{x_1} \ldots A'_{x_m} W'^{\,\ell'(x_1) + \ldots + \ell'(x_m)} (A'_{y_1} \ldots A'_{y_h} W'^{\,\ell'(y_1) + \ldots + \ell'(y_h)})^{-1}$$

with $\ell'(x_1) + \ldots + \ell'(x_m) \equiv \ell'(y_1) + \ldots + \ell'(y_h)\ (\%\bar{M})$.

Further, there is

$$A'_{x_1} \ldots A'_{x_m} \equiv C_{x_1} \ldots C_{x_m} (C_{y_1} \ldots C_{y_h})^{-1} A'_{y_1} \ldots A'_{y_h}.$$

The above equation manifests that the values of W' and $(\ell'(y_1) + \ldots + \ell'(y_h))$ do not influence the value of $(A'_{x_1} \ldots A'_{x_m})$.

If $A'_{y_1} \ldots A'_{y_h} \in [2^h, \mathbf{Þ}^h]$ changes, $A'_{x_1} \ldots A'_{x_m}$ also changes, where $A'_{y_1} \ldots A'_{y_h}$ is a composite integer. Thus, $\forall$

$x_1, \ldots, x_m, y_1, \ldots, y_h \in [1, n]$, the number of values of $A'x_1 \ldots A'x_m$ is roughly $(Ƥ^h - 2^h + 1)$.

Let $M = qƤ^m(A'y_1 \ldots A'y_h)2^{n-m-h}$, where q is a rational number.

According to (3),

$$G_z / M - L / (A'y_1 \ldots A'y_h) = (A'x_1 \ldots A'x_m) / (M A'y_1 \ldots A'y_h) = A'x_1 \ldots A'x_m / (qƤ^m 2^{n-m-h}(A'y_1 \ldots A'y_h)^2).$$

When $A'x_1 \ldots A'x_m \le qƤ^m$, there is

$$G_z / M - L / (A'y_1 \ldots A'y_h) \le qƤ^m / (qƤ^m 2^{n-m-h}(A'y_1 \ldots A'y_h)^2) = 1 / (2^{n-m-h}(A'y_1 \ldots A'y_h)^2),$$

which satisfies (4).

Assume that the value of $A'x_1 \ldots A'x_m$ distributes uniformly on $(1, M)$. If $A'y_1 \ldots A'y_h$ is a certain value, the probability that $A'x_1 \ldots A'x_m$ makes (4) hold on a specific $A'y_1 \ldots A'y_h$ is

$$qƤ^m / M = qƤ^m / qƤ^m(A'y_1 \ldots A'y_h)2^{n-m-h} = 1 / (A'y_1 \ldots A'y_h)2^{n-m-h}.$$

In fact, it is possible that $A'y_1 \ldots A'y_h$ take every value in the interval $[2^h, Ƥ^h]$ when $Cx_1, \ldots, Cx_m, Cy_1, \ldots, Cy_h$ are fixed. Thus, the probability that $A'x_1 \ldots A'x_m$ makes (4) hold is

$$\begin{aligned} P_{\forall x_1, \ldots, x_m, y_1, \ldots, y_h \in [1, n]} &= (1/(2^{n-m-h}))(1/2^h + 1/(2^h+1) + \ldots + 1/Ƥ^h) \\ &> (1/2^{n-m-h})(2(Ƥ^h - 2^h + 1)/(Ƥ^h + 2^h)) \\ &= (Ƥ^h - 2^h + 1)/(2^{n-m-h-1}(Ƥ^h + 2^h)) \approx 1/2^{n-m-h-1}. \end{aligned}$$

It is not negligible when $m + h$ is comparatively large.

Obviously, the larger $m + h$ is, the larger the probability is, and the smaller n is, the larger the probability is also. □

Property 4 exhibits the indeterminacy of $\ell(.)$ concretely.

***Property 5**:* Let $h + m \le n$. $\forall\, x_1, \ldots, x_m, y_1, \ldots, y_h \in [1, n]$, when $\ell(x_1) + \ldots + \ell(x_m) = \ell(y_1) + \ldots + \ell(y_h)$, the probability that another value of $\bar{A}_y$ makes (4) hold is larger than $1/2^{n-m-h-1}$.

Proof:

It is similar to the proving process of property 4.2. □

Property 5 illuminates that the nonuniqueness of value of $\bar{A}_y$, namely there may exist disturbing data of $\bar{A}_y$. The smaller $m + h$ is, the less disturbing data is.

***Property 6**:* (4) is necessary but not sufficient for W and W^{-1} neutralizing each other, namely $\ell(x_1) + \ldots + \ell(x_m) = \ell(y_1) + \ldots + \ell(y_h)$, where $x_1, \ldots, x_m, y_1, \ldots, y_h \in [1, n]$.

Proof: Necessity.

Suppose $\ell(x_1) + \ldots + \ell(x_m) = \ell(y_1) + \ldots + \ell(y_h)$.

Let $\{C_1, \ldots, C_n\}$ be a public sequence, and M be a modulus, where $C_i \equiv A_i W^{\ell(i)}$ (% M).

Let $G_x \equiv Cx_1 \ldots Cx_m$ (% M), and $G_y \equiv Cy_1 \ldots Cy_h$ (% M).

Let $G_z \equiv G_x G_y^{-1}$ (% M).

The rest is similar to the deduction of (4).

Insufficiency.

Suppose that (4) holds.

The contrapositive of the proposition that if (4) holds, $\ell(x_1) + \ldots + \ell(x_m) = \ell(y_1) + \ldots + \ell(y_h)$ is if $\ell(x_1) + \ldots + \ell(x_m) \ne \ell(y_1) + \ldots + \ell(y_h)$, (4) does not hold.

Hence, we need to prove that when $\ell(x_1) + \ldots + \ell(x_m) \ne \ell(y_1) + \ldots + \ell(y_h)$, (4) still holds.

In terms of property 4, when $\ell(x_1) + \ldots + \ell(x_m) \ne \ell(y_1) + \ldots + \ell(y_h)$, the (4) holds in non-negligible probability, which remind us that when $\{C_1, \ldots, C_n\}$ is generated, some subsequences formed as $\{Cx_1, \ldots, Cx_m\}$ and $\{Cy_1, \ldots, Cy_h\}$ which are validated to satisfy (4) with $\ell(x_1) + \ldots + \ell(x_m) \ne \ell(y_1) + \ldots + \ell(y_h)$ can always be found beforehand in tolerable time through adjusting the values of W and some elements in $\{A_i\}$ or $\{\ell(i)\}$.

Hence, the (4) is not sufficient for $\ell(x_1) + \ldots + \ell(x_m) = \ell(y_1) + \ldots + \ell(y_h)$. □

***Property 7**:* (4′) is necessary but not sufficient for W and W^{-1} neutralizing each other, namely $\ell(x_1) + \ldots + \ell(x_m) = \ell(y_1) + \ldots + \ell(y_h)$, where $x_1, \ldots, x_m, y_1, \ldots, y_h \in [1, n]$.

Proof: Because (4′) derives from (4), and property 6 holds, naturally property 7 holds. □

***Property 8**:* Let $m = 2$ and $h = 1$. $\forall\, x_1, x_2, y_1 \in [1, n]$, when $\ell(x_1) + \ell(x_2) \ne \ell(y_1)$,

① there always exist

$$Cx_1 \equiv A'x_1 W'^{\ell'(x_1)},\ Cx_1 \equiv A'x_2 W'^{\ell'(x_2)},\ Cy_1 \equiv A'y_1 W'^{\ell'(y_1)} \ (\%\ M),$$

such that $\ell'(x_1) + \ell'(x_2) \equiv \ell'(y_1)$ (% $\bar{M}$) with $A'y_1 \le Ƥ$;

② Cx_1, Cx_2, Cy_1 make (4″) with $A'y_1 \le Ƥ$ hold in all probability.

Proof:

① It is similar to the proving process of property 4.1.

② Let

$$G_z \equiv C_{x_1} C_{x_2} C_{y_1}^{-1} \equiv A'_{x_1} A'_{x_2} W'^{\,\ell'(x_1)+\ell'(x_2)} (A'_{y_1} W'^{\,\ell'(y_1)})^{-1} \ (\%\ M)$$

with $\ell'(x_1) + \ell'(x_2) \equiv \ell'(y_1)$ $(\%\ \bar{M})$.

Further, there is $A'_{x_1} A'_{x_2} \equiv C_{x_1} C_{x_2} C_{y_1}^{-1} A'_{y_1}$ $(\% M)$.

It is easily seen from the above equations that the values of W' and $\ell'(y_1)$ do not influence the value of $(A'_{x_1} A'_{x_2})$.

If $A'_{y_1} \in [2, \mathcal{P}]$ changes, $A'_{x_1} A'_{x_2}$ also changes. Thus, $\forall\, x_1, x_2, y_1 \in [1, n]$, the number of value of $A'_{x_1} A'_{x_2}$ is $\mathcal{P} - 1$.

Let $M = 2q\mathcal{P}^2 A'_{y_1}$, where q is a rational number.

According to (3),

$$G_z / M - L / A'_{y_1} = A'_{x_1} A'_{x_2} / (M A'_{y_1}) = A'_{x_1} A'_{x_2} / (2q\mathcal{P}^2 A'^{\,2}_{y_1}).$$

When $A'_{x_1} A'_{x_2} \le q\mathcal{P}^2$, there is

$$G_z / M - L / A'_{y_1} \le q\mathcal{P}^2 / (2q\mathcal{P}^2 A'^{\,2}_{y_1}) = 1 / (2A'^{\,2}_{y_1}),$$

which satisfies (4″).

Assume that the value of $A'_{x_1} A'_{x_2}$ distributes uniformly on $(1, M)$. Then, the probability that $A'_{x_1} A'_{x_2}$ makes (4″) hold is

$$\begin{aligned} P_{\forall x_1, x_2, y_1 \in [1, n]} &= (q\mathcal{P}^2 / (2q\mathcal{P}^2))(1/2 + \ldots + 1/\mathcal{P}) \\ &\ge (1/2)(2(\mathcal{P} - 1)/(\mathcal{P} + 2)) \\ &= 1 - 3/(\mathcal{P} + 2). \end{aligned}$$

Apparently, $P_{\forall x_1, x_2, y_1 \in [1, n]}$ is very large, and especially when $\mathcal{P}$ is pretty large, it is close to 1. □

According to property 8.2, for a certain C_{y_1} and $\forall\ C_{x_1}, C_{x_2} \in \{C_1, \ldots, C_n\}$, attack by (4″) will produce roughly $n^2/2$ possible values of A_{y_1}, including the repeated, while attack by (4) may filter out most of disturbing data of A_{y_1}. Due to in REESSE1+, every $A_{y_1} \le \mathcal{P} < n^2/2$, the number of different values of A_{y_1} is at most $\mathcal{P} > 2$ in terms of the pigeonhole principle, which indicates the running time of discriminating the original coprime sequence from the values of $A_1, \ldots,$ and A_n is much greater than $O(2^n)$.

4.3 Discussion of the Two Discrepant Cases

Now, we treat the cases of $h = 1$ and $h \ne 1$ distinguishingly, and analyze the time complexity of an attack by (4).

4.3.1 Case of *h* = 1

The $h = 1$ means that $\bar{A}_y = A_{y_1}$. If $\bar{A}_y$ is determined, a certain A_{y_1} might be exposed directly. A single A_{y_1} may be either prime or composite, and thus "whether A_{y_1} is prime" may not be regarded as the criterion of W and W^{-1} counteracting each other.

If take $m = 2$ and $h = 1$, by property 4, $P_{\forall x_1, x_2, y_1 \in [1, n]}$ is larger than $1/2^{n-4}$, and the number of expressions in the form G_z/M which lead (4) to holding is larger than $n^3/2^{n-4}$ when the interval $[1, n]$ is traversed separately by x_1, x_2, y_1. Notice that $P_{\forall x_1, x_2, y_1 \in [1, n]}$ is with respect to (4), but not with respect to (4′).

In fact, it is unmeaning to attack a private key by (4) with $\ell(x_1) + \ell(x_2) = \ell(y_1)$ because we already select an odd set Ω in the slack key transform, which avoids the occurrence of $\ell(x_1) + \ell(x_2) = \ell(y_1)$ $\forall\ x_1, x_2, y_1 \in [1, n]$, namely avoids the direct exposition of A_{y_1} at low attack cost.

In succession, we will validate property 6 with an example with $m = 2$ and $h = 1$.

Example 1:

Assume that the bit-length of a plaintext block is $n = 6$.

Let $\{A_i\} = \{11, 10, 3, 7, 17, 13\}$.

Let $M = 510931 > 11 \times 10 \times 3 \times 7 \times 17 \times 13$.

Stochastically select $\ell(1) = 9$, $\ell(2) = 6$, $\ell(3) = 10$, $\ell(4) = 5$, $\ell(5) = 7$, $\ell(6) = 8$, and $W = 17797$.

From $C_i \equiv A_i W^{\ell(i)}$ $(\%\ M)$, we obtain

$\{C_i\} = \{113101, 79182, 175066, 433093, 501150, 389033\}$.

Stochastically pick $x_1 = 2$, $x_2 = 6$, and $y_1 = 5$.

Notice that there is $\ell(5) \ne \ell(2) + \ell(6)$.

Compute

$$G_z \equiv C_2 C_6 C_5^{-1} \equiv 79182 \times 389033 \times 434038 \equiv 342114\ (\%\ 510931).$$

Presuppose that W in $C_2 C_6$ is just neutralized by W^{-1} in C_5^{-1}, and then

$$342114 \equiv A_2 A_6 A_5^{-1}\ (\%\ 510931).$$

According to (3),

$$342114 / 510931 - L / A_5 = A_2 A_6 / (510931 A_5).$$

It follows that the continued fraction expansion of

$$342114/510931 = 1/(1+1/(2+1/(37+1/(1+1/(2+\ldots+1/4))))).$$

Heuristically let

$$L / A_5 = 1/(1+1/2) = 2/3,$$

which indicates there is probably $A_5 = 3$. Further,

$$342114 / 510931 - 2/3 = 0.002922769 < 1/(2^3 \times 3^2) = 0.013888889,$$

which satisfies (4). Then $A_5 = 3$ is deduced, which is in direct contradiction to factual $A_5 = 17$, so it is impossible that (4) may serve as a sufficient condition.

Notice that in example 1, we observe $a_2 = 2$ and $a_3 = 37$, and it seems that there is a sharp increase from a_2 to a_3.

4.3.2 Case of $h \neq 1$

The $h \neq 1$ means $\bar{A}_y = A_{y_1} \ldots A_{y_h}$. It is well known that any composite $\bar{A}_y \neq p^k$ (p is a prime) can be factorized into some prime multiplicative factors, and many coprime sequences of the same length can be obtained from the factor set.

For instance, $\bar{A}_y = 210$ yields coprime sequences {5, 6, 7}, {6, 5, 7}, {3, 7, 10}, {10, 3, 7}, {2, 15, 7}, {3, 2, 35}, etc.

Property 4 make it clear that due to the indeterminacy of $\ell(.)$, no matter whether W and W^{-1} neutralize each other or not, in most cases, many values of $\bar{A}_y$ which may be written as the product of h coprime integers, and satisfy (4) can be found out from the convergents of the continued fraction of G_z / M when the interval $[1, n]$ is traversed respectively by $x_1, \ldots, x_m, y_1, \ldots, y_h$. Thus, "whether $\bar{A}_y$ can be written as the product of h coprime integers" may not be regarded as a criterion for verifying that W and W^{-1} neutralize each other.

Moreover, even if the t values $v_1, \ldots, v_t$ of $(A_{y_1} A_{y_2} \ldots A_{y_h})$ are obtained, where y_1 is fixed, and $y_2, \ldots, y_h$ are varied, $\gcd(v_1, \ldots, v_t)$ can not be judged to be A_{y_1} in terms of the definition of a coprime sequence.

If take $m = 2$ and $h = 2$ ($m = 3$ and $h = 1$ is unmeaning as the set Ω is odd), by property 4 and $P_{\forall x_1, x_2, y_1, y_2 \in [1, n]}$, the number of expressions in the form G_z / M which lead (4) to holding is larger than $n^4 / 2^{n-5}$ when the interval $[1, n]$ is traversed respectively by x_1, x_2, y_1, y_2.

4.3.3 Time Complexity of Continued Fraction Attack by (4) with $\ell(x_1) + \ell(x_2) = \ell(y_1) + \ell(y_2)$

We temporarily disregard the indeterminacy, and purely consider the continued fraction attack itself.

Since Ω is an odd set, $\ell(x_1) + \ell(x_2) = \ell(y_1)$ can not occur in practice. The time complexity of the continued fraction attack by (4) with $\ell(x_1) + \ell(x_2) = \ell(y_1)$ will not be discussed. Likewise, since it is stipulated that $\forall\ e_1, e_2, e_3, e_4 \in \Omega$, $e_1 + e_2 + e_3 \neq e_4$, the case of $\ell(x_1) + \ell(x_2) + \ell(x_3) = \ell(y_1)$ will not be discussed.

Firstly, the number of all possible $G_z \equiv C_{x_1} C_{x_2} (C_{y_1} C_{y_2})^{-1}$ (% M) is n^4. Obtaining G_z takes 3 modular multiplications and 1 inversion. Notice that a modular multiplication or inversion needs $O(2\log^2 M)$ bit operations.

Getting the continued fraction expansion $[0; a_1, a_2, \ldots, a_t]$ of G_z / M takes $\log M$ divisions. Getting all the convergents $c_1, c_2, \ldots, c_t$ takes $2\log M$ both multiplications and additions. The bit-lengths of all related operands may be regards as $\log M$ [25].

Hence, to find out all possible values of $L / \bar{A}_y$ by (4) and to make comparisons, an adversary will take roughly $O(2n^4 \log^2 M (3\log M + 5))$ bit operations.

Secondly, because the number of combinations of n elements from Ω is $C_{2n}^{n} > 2^n$, determining the codomain of $\ell(.)$ will take $O(2^n)$. Moreover, it is possible that for the set $\{\ell(1), \ldots, \ell(n)\}$, there exist a very few of cases where $\exists\, x_1, x_2, y_1, y_2 \in [1, n]$, $\ell(x_1) + \ell(x_2) = \ell(y_1) + \ell(y_2)$ holds. Thus, it is very hard to judge $\ell(x_1) + \ell(x_2) = \ell(y_1) + \ell(y_2)$ even though $A_{y_1} A_{y_2} \leq \grave{P}^2$ is found, and the indeterminacy involved in (4) is neglected. Further, it is infeasible to determine the values of $\ell(x_1)$, $\ell(x_2)$, $\ell(y_1)$, $\ell(y_2)$, and W.

To sum up, when the lever function $\ell(.)$ as an injective from $[1, \ldots, n]$ to Ω exists, the time complexity of the continued fraction attack is at least $O(2^n)$, namely the indeterministic reasoning by (4) is ineffectual.

4.4 Relation between the Lever Function $\ell(.)$ and a Random Oracle

An oracle is a mathematical abstraction, a theoretical black box, or a subroutine of which the running time may not be considered [21][26]. In particular, in cryptography, an oracle may be treated as a subcomponent of an adversary, and lives its own life independent of the adversary. Usually, the adversary interacts with the oracle but cannot control its behavior.

A random oracle is an oracle which answers to every query with a completely random and unpredictable value chosen uniformly from its output domain, except that for any specific query, it outputs the same value every time it receives that query if it is supposed to simulate a deterministic function [27].

Random oracles are utilized in cryptographic proofs for relpacing any irrealizable function so far which can provide the mathematical properties required by the proof. A cryprosystem or a protocol that is proven secure using such a proof is described as being secure in the random oracle model, as opposed to being secure in the standard model where the integer factorization problem, the discrete logarithm problem etc are assumed to be hard. When a random oracle is used within a security proof, it is made available to all participants, including adversaries. In practice, random oracles producing a bit-string of infinite length which can be truncated to the length desired are typically used to model cryptographic hash functions in schemes where strong randomness assumptions of a hash function′s output are needed.

In fact, it draws attention that certain artificial signature and encryption schemes are proven secure in the random oracle model, but are trivially insecure when any real function such as the hash function MD5 or SHA-1 is substituted for the random oracle [28]. Nevertheless, for any more natural protocol, a proof of security in the random oracle model gives very strong evidence that an attacker have to discover some unknown and undesirable property of the hash function used in the protocol.

A function or algorithm is randomized if its output depends not only on the input but also on some random ingredient, namely if its output is not uniquely determined by the input. Hence, to a function or algorithm, the randomness is almost equivalent to indeterminacy.

Correspondingly, the indeterminacy of $\ell(.)$ may be expounded in terms of a random oracle.

Suppose that $\bar{O}_d(y, g)$ is an oracle for solving $y \equiv g^x$ (% M) for x, and $\bar{O}_\ell$ is an oracle for solving $C_i \equiv A_i W^{\ell(i)}$ (% M) for $\ell(i)$, where M is prime, and i is from 1 to n.

Let $\check{D}$ be a database which stores records $(\{C_1, …, C_n\}, M, \{\ell(1), …, \ell(n)\})$ computed already. If the order of some C_i′s is changed, $\{C_1, …, C_n\}$ is regarded as a distinct sequence.

The structure of $\bar{O}_\ell$ is as follows:

Input: $\{C_1, …, C_n\}, M$.

Output: $\{\ell(1), …, \ell(n)\}$.

S1: If find $(\{C_1, …, C_n\}, M)$ in $\check{D}$, return related $\{\ell(1), …, \ell(n)\}$, and end.

S2: Randomly produce a coprime sequence $A_1, …, A_n$

with each $A_i \le \boldsymbol{Þ}$ and $(\max_{1\le i\le n} A_i)^n < M$.

S3: Randomly pick a generator $W \in \mathbb{Z}_M^*$.

S4: Evaluate $\ell(i)$ by calling $\bar{O}_d(C_i A_i^{-1}, W)$ for $i = 1, …, n$.

S5: Store $(\{C_1, …, C_n\}, M, \{\ell(1), …, \ell(n)\})$ to $\check{D}$.

S6: Return $\{\ell(1), …, \ell(n)\}$, and end.

Of course, $\{A_i\}$ and W as side results may be outputted.

Obviously, for the same input $(\{C_1, …, C_n\}, M)$, the output is the same, and for a different input, a related output is random and unpredictable.

Since $C_i A_i^{-1}$ is pairwise distinct, and W is a generator, $\ell(i)$ in the result is pairwise distinct. In addition, according to definition 2, every $\ell(i) \in [1, \bar{M}]$ may be outside of Ω. Thus, the result $\{\ell(1), …, \ell(n)\}$ may be regarded as a lever function though not the original lever function.

The $\bar{O}_\ell$ is perhaps strange to some people because they have never met any analogous oracle in classical cryptosystems.

The above discussion expounds soundly once more why the indeterministic reasoning, namely the continued fraction attack by (4) is ineffectual.

5 Security Analysis of the Encryption

We analyze the exact security of the key generator and encryption algorithm of the prototypal asymmetric cryptosystem REESSE1+, where $C_i \equiv (A_i W^{\ell(i)})^\delta$ (% M) and $\bar{G} \equiv \prod_{i=1}^{n} C_i^{b_i}$ (% M) with $W, \delta \in [1, \bar{M}]$, $\ell(i) \in \Omega = \{1, 3, …, 2n-1\}$, and $A_i \in \Lambda = \{2, 3, …, 1201\}$.

We know that the first n primes in the set $\mathbb{N}$ can constitute a smallest coprime sequence, and there has to be $M > (\max_{1\le i\le n} A_i)^n$ (see section 3.1). Hence, when $n = 80$, 96, 112, or 128, there is $\log M \approx 696$, 864, 1030, or 1216. In this case, IFP and DLP can almost be solved in tolerable time. In addition, when the density of a knapsack is less than 1, SSP can also be solved in tolerable time [11][26].

"In tolerable time" means that the running time of an algorithm for solving a problem is able-waited when the dominant parameter of the complexity is relatively small. For example, when $n = 80$, the running time of $O(2^{n/2})$ is tolerable, and when $\log M = 384$, $O(L_M[1/3, 1.923]) = O(2^{56})$ is also tolerable [29].

5.1 Extracting a Private Key from a Public Key Is of MPP

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, averagely the ciphertext has no direct help to inferring the private key.

In the prototypal cryptosystem, owing to $\delta \in [1, \bar{M}]$, the continued fraction attack discussed in section 4 is utterly ineffectual.

5.1.1 Interaction of the Key Transform Items

◇ Eliminating W through $\ell(x_1) + \ell(x_2) = \ell(y_1) + \ell(y_2)$

$\forall\, x_1, x_2, y_1, y_2 \in [1, n]$, assume that there is $\ell(x_1) + \ell(x_2) = \ell(y_1) + \ell(y_2)$.

Let $G_z \equiv C_{x_1} C_{x_2} (C_{y_1} C_{y_2})^{-1}$ (% M), namely

$$G_z \equiv (A_{x_1} A_{x_2} (A_{y_1} A_{y_2})^{-1})^{\delta} \ (\%\ M).$$

If adversaries divine $A_{x_1}, A_{x_2}, A_{y_1}, A_{y_2} < 720$, and compute u, v_{x_1}, v_{x_2}, v_{y_1}, v_{y_2} in the running time of at least $L_M[1/3, 1.923]$ such that

$$G_z \equiv g^u,\ A_{x_1} \equiv g^{v_{x_1}},\ A_{x_2} \equiv g^{v_{x_2}},\ A_{y_1} \equiv g^{v_{y_1}},\ A_{y_2} \equiv g^{v_{y_2}} \ (\%\ M),$$

where g is a generator of $(\mathbb{Z}_M^*, \cdot)$, then

$$u \equiv (v_{x_1} + v_{x_2} - v_{y_1} - v_{y_2})\delta \ (\%\ \bar{M}).$$

If $\gcd(v_{x_1} + v_{x_2} - v_{y_1} - v_{y_2}, \bar{M}) \mid u$, the congruence for δ has solutions. Because each of $A_{x_1}, A_{x_2}, A_{y_1}, A_{y_2}$ may traverse the interval Λ, x_1, x_2, y_1, y_2 are unfixed, and the congruence may have n solutions, the number of values of δ is about $n^5|\Lambda|^4$.

In succession, the most effectual approach seeking W is that for every i, divine A_i and $\ell(i)$, find V_i, namely the value set of W, by $C_i \equiv (A_i W^{\ell(i)})^{\delta}$ (% M), and if there exists $W_1 \in V_1, \ldots, W_n \in V_n$ being equal pairwise, the divination of δ, $\{A_i\}$, and $\{\ell(i)\}$ is thought right. Notice that to avoid seeking $\ell(i)$-th roots, may let $W = g^{\mu}$ % M.

Due to $\prod_{i=1}^{k} \acute{p}_i^{\bar{e}_i} \mid \bar{M}$ and $\acute{p}_k \approx 2n$, there exists $\ell(i) \mid \bar{M}$, and the size of every V_i is about $n|\Omega||\Lambda|$.

In summary, the running time of the above attack is at least

$$T_1 = n|\Lambda| L_M[1/3, 1.923] + (2n^5|\Lambda|^4) 2\log^2 M + (2n^5|\Lambda|^4)(n|\Omega||\Lambda|) n (2\log^2 M).$$

When $n = 80$, there is $\lceil \log M \rceil \approx 696$, and $T_1 = 2^{117} > 2^n$.

When $n = 96$, there is $\lceil \log M \rceil \approx 864$, and $T_1 = 2^{117} > 2^n$.

When $n = 112$, there is $\lceil \log M \rceil \approx 1030$, and $T_1 = 2^{124} > 2^n$.

When $n = 128$, there is $\lceil \log M \rceil \approx 1216$, and $T_1 = 2^{124} \approx 2^n$.

Therefore, T_1 is roughly exponential in n.

Clearly, the running time of attack by eliminating W through $\ell(x_1) + \ell(x_2) + \ell(x_3) = \ell(y_1)$ is the same as attack by eliminating W through $\ell(x_1) + \ell(x_2) = \ell(y_1) + \ell(y_2)$.

◇ Eliminating W through the $\|W\|$-th Power

Due to $\log M \approx 696$, 864, 1030, or 1216, $\bar{M}$ can be factorized in tolerable time. Again due to $\prod_{i=1}^{k} \acute{p}_i^{\bar{e}_i} \mid \bar{M}$ and $\prod_{i=1}^{k} \bar{e}_i \ge 2^{10}$, $\|W\|$ can be divined in the running time of about 2^{10}.

Raising either side of $C_i \equiv (A_i W^{\ell(i)})^{\delta}$ % M to the $\|W\|$-th power yields

$$C_i^{\|W\|} \equiv (A_i)^{\delta\|W\|} \ \%\ M.$$

Let $C_i \equiv g^{u_i}$ (% M), and $A_i \equiv g^{v_i}$ (% M), where g is a generator of $(\mathbb{Z}_M^*, \cdot)$. Then

$$u_i\|W\| \equiv v_i\|W\|\delta \ (\%\ \bar{M})$$

for $i = 1, \ldots, n$.

The above congruence looks to be the MH transform [8]. Actually, $\{v_1\|W\|, \ldots, v_n\|W\|\}$ is not a super increasing sequence, and there is not necessarily $\log(u_i\|W\|) = \log \bar{M}$.

Because $v_i\|W\| \in [1, \bar{M}]$ is stochastic, the inverse δ^{-1} % $\bar{M}$ not need be close to the minimum $\bar{M}/(u_i\|W\|)$, $2\bar{M}/(u_i\|W\|)$, …, or $(u_i\|W\| - 1)\bar{M}/(u_i\|W\|)$. Namely δ^{-1} may lie at any integral position of the interval $[k\bar{M}$

$/(u_i\|W\|)$, $(k+1)\bar{M}/(u_i\|W\|)]$, where $k = 0, 1, \ldots, u_i\|W\| - 1$, which illustrates the accumulation point of minima do not exist. Further observing, in this case, when i traverses the interval $[2, n]$, the number of intersections of the intervals including δ^{-1} is likely $\max_{2\le i\le n}\{u_i\|W\|\}$ which is promisingly close to $\bar{M}$. Therefore, the Shamir attack by the accumulation point of minima is fully ineffectual [9].

Even if find out δ^{-1} by the Shamir attack method, because each of v_i has $\|W\|$ solutions, the number of potential sequences $\{g^{v_1}, \ldots, g^{v_n}\}$ is up to $\|W\|^n$. Considering needing to verify whether $\{g^{v_1}, \ldots, g^{v_n}\}$ is a coprime sequence for each different sequence $\{v_1, \ldots, v_n\}$, the number of coprime sequences is in proportion to $\|W\|^n$. Hence, the initial $\{A_1, \ldots, A_n\}$ can not be determined in polynomial time. Further, the value of W can not be computed, and the values of $\|W\|$ and δ^{-1} can not be verified in polynomial time, which indicates that $C_i \equiv (A_i W^{\ell(i)})^\delta$ (% M) can also be resistant to the attack by the accumulation point of minima.

Additionally, adversaries may divine value of A_i in running time of about $|\Lambda|$, where $i \in [1, n]$, and compute δ by $u_i\|W\| \equiv v_i\|W\|\delta$ (% $\bar{M}$). However, because of $\|W\| \mid \bar{M}$, the equation will have $\|W\|$ solutions. Therefore, the running time of finding the original δ is at least

$$\begin{aligned} T_2 &= n|\Lambda|L_M[1/3, 1.923] + 2^{10}|\Lambda|\|W\| \\ &= n|\Lambda|L_M[1/3, 1.923] + 2^{10}|\Lambda|2^{n-20} \\ &\approx n|\Lambda|L_M[1/3, 1.923] + 2^n > 2^n. \end{aligned}$$

It is at least exponential in n when $80 \le n \le 128$.

Again, the equation $\alpha \equiv \hbar\delta(W\prod_{i=1}^n A_i)^{-\delta S}$ (% M) contains $\hbar$, δ, W, and $\prod_{i=1}^n A_i$, so it is impossible to separate them distinctly. The equations $\alpha \equiv \delta^{(\delta^n + \delta W^{n-1})T}$, $\beta \equiv \delta^{W^n T}$ (% M) contain δ and W, and the time of seeking them will be at least $O(2^n)$ (see section 6.2.3). If the three equations are considered simultaneously, it is also impossible to determine the four variables almost independent.

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

5.1.2 Consideration of a Certain Single C_i

Assume that there is only a solitary $C_i = (A_i W^{\ell(i)})^\delta$ % M — $i = 1$ for example, and other C_i's ($i = 2, \ldots, n$) do not exist.

Clearly, divining $A_1 \in \Lambda$ and $\ell(1) \in \Omega$, the parameters W and $\delta \in (1, \bar{M})$ can be computed. Thus, the number of solution $(A_1, \ell(1), W, \delta)$ will be up to $|\Omega||\Lambda|\bar{M}^2 > 2^n$, which manifests that the original $(A_1, \ell(1), W, \delta)$ can not be determined in subexponential time in n [30].

Seeking original A_i, W, $\ell(i)$, δ from $C_i \equiv (A_i W^{\ell(i)})^\delta$ (% M) with $A_i \in \{2, \ldots, 1201\}$ and $\ell(i) \in \{1, 3, \ldots, 2n - 1\}$ for $i = 1, \ldots, n$ is called the multivariate permutation problem (MPP).

5.2 Recovering a Plaintext from a Ciphertext and a Public Key Is of ASPP

The security of a REESSE1+ plaintext is based on the anomalous subset product problem $\bar{G} \equiv \prod_{i=1}^n C_i^{\underline{b}_i}$ (% M) which we will dissect.

5.2.1 SPP Should Be Harder than DLP

Let $\{C_1, \ldots, C_n\}$ be a public key, and then seeking a binary plaintext $b_1 \ldots b_n$ from known $\bar{G} \equiv \prod_{i=1}^n C_i^{b_i}$ (% M) is called the (modular) subset product problem, shortly SPP.

Evidently, $\prod_{i=1}^n C_i^{b_i} = LM + \bar{G}$. Owing to $L \in [1, \bar{M}]$, deriving the non-modular product $\prod_{i=1}^n C_i^{b_i}$ from $\bar{G}$ is infeasible, which means inferring $b_1 \ldots b_n$ from $\bar{G}$ is not a factorization problem.

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

$$\bar{G} \equiv C^{\sum_{i=1}^n b_i} \text{ (\% } M).$$

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 may express equivalently the sum $\sum_{i=1}^n b_i$ as $\sum_{i=1}^n b_i 2^{i-1}$, and let $z = \sum_{i=1}^n b_i 2^{i-1}$.

Correspondingly,

$$\bar{G} \equiv C^z \text{ (\% } M),$$

which is a discrete logarithm problem.

In practice, when $C_1, \ldots, C_n$ are generated, we can check $C_1, \ldots, C_n$ to make $C_1, \ldots, C_n$ pairwise distinct. Therefore, factually, SPP can not be reduced to DLP, namely SPP is generally harder than DLP in the same prime field.

Another evidence.

Presume that DLP can be solved in tolerable subexponential time.

When DLP can be solved in tolerable time, $\bar{M}$ can also be factorized [14] [21], so a generator can be found through the algorithm 4.80 in section 4.6 of [21].

Let g be a generator of $(\mathbb{Z}_M^*, \cdot)$.

Let $C_1 \equiv g^{u_1}$ (% M), …, $C_n \equiv g^{u_n}$ (% M), $\bar{G} \equiv g^v$ (% M).

Then, solving $\bar{G} \equiv \prod_{i=1}^n C_i^{b_i}$ (% M) is equivalent to solving

$$b_1 u_1 + \ldots + b_n u_n \equiv v \ (\% \bar{M}),$$

which is a subset sum problem.

It has been proved that SSP is NP-Complete (in its feasibility recognition form), and the computational version of the subset sum problem is NP-hard [4][21], which illustrates that even if DLP can be solved, $b_1 \ldots b_n$ can not be found yet in polynomial time in general. Therefore, solving $\bar{G} \equiv \prod_{i=1}^n C_i^{b_i}$ (% M) for $b_1 \ldots b_n$ is harder than solving DLP so far.

However, it should been noted that the subset sum problem will degenerate from NPC when the density and length of a sequence are comparatively small [11][31], which manifests that provable security by the polynomial time reduction is substantially relative and asymptotic.

5.2.2 ASPP Can Resist the L^3 Lattice Base Reduction

It is known from section 3.2 that the ciphertext $\bar{G} \equiv \prod_{i=1}^n C_i^{b_i}$ (% M).

Let $\{C_1, \ldots, C_n\}$ be a public key, and then seeking original $\underline{b}_1 \ldots \underline{b}_n$ from $\bar{G} \equiv \prod_{i=1}^n C_i^{b_i}$ (% M) is called the anomalous (modular) subset product problem, shortly ASPP.

Let g be a generator of $\mathbb{Z}_M^*$.

Let $C_1 \equiv g^{u_1}$ (% M), …, $C_n \equiv g^{u_n}$ (% M), $\bar{G} \equiv g^v$ (% M).

Then, seeking $\underline{b}_1 \ldots \underline{b}_n$ from $\bar{G}$ is equivalent to solving the congruence

$$u_1 \underline{b}_1 + \ldots + u_n \underline{b}_n \equiv v \ (\% \bar{M}), \tag{5}$$

where v may be substituted with $v + k\bar{M}$ with $k \in [0, n-1]$ [32]. $\{u_1, \ldots, u_n\}$ is called a compact sequence due to $\underline{b}_i \in [0, n]$, and correspondingly (5) is called the anomalous subset sum problem (ASSP).

Recall [10] and [11]. Let $\{a_1, \ldots, a_n\}$ be a positive integer sequence, $\hat{e} = \langle e_1, \ldots, e_n, 0 \rangle$ with $e_i \in [0, 1]$ be the solution vector, $s = \sum_{i=1}^n a_i e_i$, and $t = \sum_{i=1}^n a_i$.

In [10], there are two important conditions:

$$t / n \le s \le (n-1) t / n, \text{ and } \|\hat{e}\|^2 \le n / 2,$$

where $\|\hat{e}\|$ denotes the distance in l_2-Norm of the vector $\hat{e}$, which decides the threshold density < 0.6463.

In [11], there are similar

$$t / n \le s \le (n-1) t / n, \text{ and } \|\hat{e}\|^2 \le n / 4,$$

which decide the threshold density < 0.9408.

However, for (5), due to $0 \le \underline{b}_i \le n$, the similar conditions do not hold.

It is well understood that the L^3 lattice base reduction algorithm is employed in cryptanalysis to find the shortest vector or approximately shortest vectors in a lattice, and hence, if a solution to the subset sum problem has a comparatively big distance, or is not unique, it will not occur in the reduced base.

Let $\mathbb{L}$ be a lattice spanned by the vectors

$$\begin{gathered}\langle 1, 0, \ldots, 0, N u_1 \rangle, \\ \langle 0, 1, \ldots, 0, N u_2 \rangle, \\ \ldots\ldots, \\ \langle 0, 0, \ldots, 1, N u_n \rangle, \\ \langle 0, 0, \ldots, 0, N(v + k\bar{M}) \rangle\end{gathered}$$

which compose a base of the lattice, where N is a positive integer greater than $(n^2)^{1/2} = n$ (but not much greater, or else will influence speed of the L^3 reduction algorithm). Notice that because g is random, $\mathbb{L}$ is also random.

Let $\underline{D}$ be the determinant of a matrix corresponding to the lattice base. Then, by the Guassian heurisic, the expected size of the shortest vector in $\mathbb{L}$ on the base of $n+1$ dimensions lies between [15]

$$\underline{D}^{1/(n+1)}((n+1)/(2\pi e))^{1/2} \text{ and } \underline{D}^{1/(n+1)}((n+1)/(\pi e))^{1/2},$$

where $e \approx 2.7182818$.

In our case, there is $\log M / (n+1) \approx 9$, and the right expression is roughly

$$(Nk2^{9(n+1)})^{1/(n+1)}((n+1)/(\pi e))^{1/2} \approx 2^9((n+1)/2^3)^{1/2} \approx 2^7(2n)^{1/2}.$$

For (5), the largest distance of the solution vector $\langle \underline{b}_1, \ldots, \underline{b}_n, 0 \rangle$ is n, and thus the solution vector will possibly occur in the reduced base. However, whether the solution vector occurs surely or not will be influenced by a knapsack density.

To compute the density of the compact sequence, we extend $\{u_1, \ldots, u_n\}$ into

$$\{u_1, 2u_1, \ldots, nu_1, u_2, 2u_2, \ldots, nu_2, \ldots\ldots, u_n, 2u_n, \ldots, nu_n\}.$$

It is not difficult to understand that the length of the extend sequence is n^2.

The density of the compact sequence $\{u_1, \ldots, u_n\}$ is

$$D \approx n^2 / \log M.$$

When $n = 80$ and $\log M = 696$, $D \approx 9.19 > 2 > 1$.

When $n = 96$ and $\log M = 864$, $D \approx 10.66 > 2 > 1$.

When $n = 112$ and $\log M = 1030$, $D \approx 12.18 > 2 > 1$.

When $n = 128$ and $\log M = 1216$, $D \approx 13.47 > 2 > 1$.

$D > 2$ indicates that many different subsets will have the identical sum, namely the solution to (5) is not unique, and the original solution is possibly not shortest for $\underline{b}_i \in [0, n]$. Thus, it is very likely that the original solution does not occur in the reduced base which only contains $n + 1$ vectors.

Further, we can estimate the time cost of the L^3 lattice base attack.

Although SLLL, namely segment LLL in floating point arithmetic and L^2-FP are two of currently fast lattice base reduction algorithms [33][34], because floating point operation on integers greater than the modulus M with $\log M \geq 696$ can not be executed directly, and even are instable under a low precision circumstance, it is inappropriate to utilize these two algorithms to find the solution vector $\langle \underline{b}_1, \ldots, \underline{b}_n, 0\rangle$, which manifests that the only classical L^3 algorithm is appropriate.

According to [21], the running time of attack on equation (5) from ASPP by the lattice base reduction algorithm is roughly

$$\mathcal{T}_L \approx O(n L_M[1/3, 1.923] + n(n+1)^6 (\log M^2)^3)$$

on condition that N is slightly greater than n.

When $n = 80$ and $\log M = 696$, $\mathcal{T}_L \approx 2^{83}$.

When $n = 96$ and $\log M = 864$, $\mathcal{T}_L \approx 2^{83}$.

When $n = 112$ and $\log M = 1030$, $\mathcal{T}_L \approx 2^{86}$.

When $n = 128$ and $\log M = 1216$, $\mathcal{T}_L \approx 2^{86}$.

However, as is pointed out in the above, owing to $D > 9 > 2 > 1$ and $\underline{b}_i \in [0, n]$, it is almost impossible that the solution vector $\langle \underline{b}_1, \ldots, \underline{b}_n, 0\rangle$ occurs in the final reduced base, which means that attack by L^3 algorithm will be unavailing.

Besides, we also see that there exists an exhaustive search attack on the plaintext block $b_1 \ldots b_n$. Clearly, the running time of such an attack is $O(2^n)$ arithmetic steps.

Hence, the plaintext security of REESSE1+ is built on the problem $\bar{G} \equiv \prod_{i=1}^{n} C_i^{\underline{b}_i}$ (% M) which contains the trapdoor information, and means that computing an anomalous subset product from subset elements is tractable while seeking the involved elements of the set from the product is intractable.

5.3 Avoid Adaptive-chosen-ciphertext Attack

Theoretically, absolute most of public key cryptographies may probably be faced with adaptive-chosen-ciphertext attack.

During the late 1990s, Daniel Bleichenbacher demonstrated a practical adaptive-chosen-ciphertext attack on SSL servers using a form of RSA encryption [35]. Almost at the same time, The Cramer-Shoup asymmetric encryption algorithm was proposed [36]. It is the first efficient scheme proven to be secure against adaptive-chosen-ciphertext attack using standard cryptographic assumptions, which implies that not all uses of cryptographic hash functions require random oracles — some require only the property of collision resistance, and an extension of the Elgamal algorithm which is extremely malleable.

It is lucky that REESSE1+ can avoid the adaptive-chosen- ciphertext attack. In REESSE1+, a ciphertext may be produced rapidly according to the following algorithm:

Assume that $b_1 \ldots b_n$ is a plaintext block, and $\{C_1, \ldots, C_n\}$ is a public key.

S1: Set $k \leftarrow 0$, $i \leftarrow 1$.

S2: If $b_i = 0$, let $k \leftarrow k + 1$, $\underline{b}_i \leftarrow 0$;
    else let $\underline{b}_i \leftarrow k + 1$, $k \leftarrow 0$.

S3: Let $i \leftarrow i + 1$.
    If $i \leq n$, goto S2.

S4: Randomly produce $d_1 \ldots d_n \in \{0, 1\}^n$.

S5: If $\underline{ƀ}_n = 0$, set $r \leftarrow n - k$, $d_r = 1$, $\underline{ƀ}_r \leftarrow \underline{ƀ}_r + k$.

S6: Compute $\bar{G} \leftarrow \prod_{i=1}^{n} (C_{i\,d_i + (i - \underline{ƀ}_i + 1)\neg d_i})^{\underline{ƀ}_i} \% M$.

On input of an identical plaintext many times, the algorithm will return the many ciphertexts which may be different from one another. Contrariwise, it is easily understood that a ciphertext can be decrypted in polynomial time, and the output will uniquely correspond to the original plaintext.

Another approach to avoiding the adaptive-chosen-ciphertext attack is to append a stochastic fixed-length binary sequence to the terminal of every plaintext bock when it is encrypted. For example, a concrete implementation is referred to the OAEP+ scheme [37].

6 Security Analysis of the Signature

Firstly, we give a related concept.

***Definition 3**:* Let A and B be two computational problems. A is said to reduce to B in polynomial time, written as $A \le_{\mathrm{P}} B$, if there is an algorithm for solving A which calls, as a subroutine, a hypothetical algorithm for solving B, and runs in polynomial time excluding the time of the algorithm for B.

The hypothetical algorithm for solving B is called an oracle. It is easy to understand that no matter what the running time of the oracle is, it does not influence the result of the comparison.

$A \le_{\mathrm{P}} B$ means that the difficulty of A is not greater than that of B, namely the running time of the fastest algorithm for A is not greater than that of the fastest algorithm for B when all polynomial times are treated as being pairwise equivalent. Concretely speaking, if A can not be solved in polynomial or subexponential time, B can not also be solved in corresponding polynomial or subexponential time; and if B can be solved in polynomial or subexponential time, A can also be solved in corresponding polynomial or subexponential time.

***Definition 4**:* Let A and B be two computational problems. If $A \le_{\mathrm{P}} B$ and $B \le_{\mathrm{P}} A$, then A and B are said to be computationally equivalent, written as $A =_{\mathrm{P}} B$.

$A =_{\mathrm{P}} B$ means that either if A is a hardness on condition that the dominant variable approaches a large number, B is also a hardness on the same condition; or A, B both can be solved in linear or polynomial time.

Definition 3 and 4 suggest a reductive proof method called polynomial time (Turing) reduction (PTR) [21]. Provable security by PTR is substantially relative and asymptotic just as a one-way function is. Relative security implies that the security of a cryptosystem based on a problem is comparative, but not absolute. Asymptotic security implies that even if a cryptosystem based on a problem is proven to be secure, it is practically secure only on condition that the dominant parameter is large enough.

Naturally, we will meditate whether $A <_{\mathrm{P}} B$ exists or not. The definition of $A <_{\mathrm{P}} B$ may possibly be given theoretically, but the proof of $A <_{\mathrm{P}} B$ is not easy in practice.

6.1 Extracting a Related Private Key from a Signature Is of Exponential Time Complexity

Assume that p is a prime integer, and $k \mid (p - 1)$ holds. In terms of the probabilistic algorithm in section 1.6 of [38], the time complexity of finding out a random solution to $x^k \equiv c$ ($\% \, p$) is at least the maximum of $O(2^{k-1})$ and $O(p / k)$. Thus, when $k > 80$ or $p / k > 2^{80}$, this algorithm is ineffectual actually.

However, when $\gcd(k, p - 1) = 1$ or $\gcd(k, (p - 1) / k) = 1$ with $k \mid (p - 1)$, 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 there exist

$$Q \equiv (R\,G_0)^S \delta \hbar \ (\% \, M),$$

$$U \equiv (R\,W^{\underline{k}-\delta})^Q \delta^{\bar{a}\text{Đ}r} \ (\% \, M).$$

In term of section 3.4, $\delta^{\bar{a}\text{Đ}r}$ belongs to the subgroup of order $\text{đ}T$ of $\mathbb{Z}_M^*$. Because of $T \ge 2^n$, divining the value of $\delta^{\bar{a}\text{Đ}r}$ is impossible.

Let $\check{e} \equiv \delta^{\bar{a}\text{Đ}rT}$ ($\% \, M$), where $\check{e} \in \check{S}$ — the subgroup of order đ, the second congruence is equivalent to

$$U^T \equiv (R\,W^{\underline{k}-\delta})^{QT} \check{e} (\% \, M).$$

When an attacker attempts to seek $R\,G_0$ or $R\,W^{\underline{k}-\delta}$, he has to solve the two equations:

$$x^S \equiv Q \delta^{-1} \hbar^{-1} \ (\% \, M),$$

$$y^{QT} \equiv U^T \check{e}^{-1} \ (\% \, M).$$

For the first congruence, because δ, $\hbar$ are unknown, and its right is not a constant, it is impossible to solve the equation for $(R\,G_0)$. If δ is divined, the probability of hitting δ is $1 / \|\delta\| < 1 / 2^n$. Besides, it is more difficult to divine $\hbar$.

For the second congruence, there is $\|\breve{e}^{-1}\| \le \text{đ}$. Assume that đ is guessed out, and the solutions to $x^{\text{đ}} \equiv 1$ (% M) can be found out, then $\breve{e}^{-1}$ may possibly be hit.

The equation $x^{\text{đ}} \equiv 1$ (% M) may probably have the trivial root, and to its other roots, there are three approaches: ① algorithm 4.80 in section 4.6 of [21], which first finds out a generator g, then let $x \equiv g^{\text{Đ}T}$; ② the probabilistic algorithm in section 1.6 of [38], of which the running time is $O(M / \text{đ})$; ③ the Index-calculus method of seeking discrete logarithms, of which the running time is $L_M[1/3, 1.923]$.

Even if $\breve{e}^{-1}$ is known, $\gcd(Q, \bar{M}) = 1$ or $\gcd(Q, \bar{M}/Q) = 1$ holds, and $\gcd(T, \bar{M}/T) = 1$ holds, namely the trivial root to the second congruence exists, the probability that the trivial root just equals the specific $R W^{\underline{k}-\delta}$ is only $1/T \le 1/2^n$, and moreover due to the randomicity of R, it is thoroughly impossible to separate δ, W, or $\underline{k}$ from $R W^{\underline{k}-\delta}$.

Additionally, substituting R in $U^T \equiv (R W^{\underline{k}-\delta})^{QT} \breve{e}$ (% M) with $G_0^{-1}(Q(\delta\hbar)^{-1})^{S^{-1}}$ which is from $Q \equiv (R G_0)^S \delta\hbar$ (% M) gives

$$U^T \equiv (G_0^{-1}(Q(\delta\hbar)^{-1})^{S^{-1}} W^{\underline{k}-\delta})^{QT} \breve{e} \ (\% \ M),$$

namely

$$U^T \equiv (G^{-1} \bar{G}(Q(\delta\hbar)^{-1})^{S^{-1}} W^{-\delta})^{QT} \breve{e} \ (\% \ M),$$

where $G \equiv G_0 G_1$ and $\bar{G} \equiv G_1 W^{\underline{k}}$ (% M).

Thus,

$$((GW^{\delta})^{-1}(\delta\hbar)^{-S^{-1}})^{QT} \equiv U^T(\bar{G}Q^{S^{-1}})^{-QT} \breve{e}^{-1} \ (\% M).$$

Similarly, if $\breve{e}^{-1}$ is known, and $(U^T(\bar{G}Q^{S^{-1}})^{-QT} \breve{e}^{-1})^{\bar{M}/(Tk)} \equiv 1$ (% M), where $k = \gcd(Q, \bar{M})$, through the Index-calculus method, one may find out all the solutions to the equation

$$x^{QT} \equiv U^T(\bar{G}Q^{S^{-1}})^{-QT} \breve{e}^{-1} \ (\% M).$$

However, the probability that a certain found solution is no other than $(GW^{\delta})^{-1}(\delta\hbar)^{-S^{-1}}$ is less than $1/T \le 1/2^n$. Further, the running time of separating G, W, $\hbar$, or δ from $(GW^{\delta})^{-1}(\delta\hbar)^{-S^{-1}}$ is at least $O(\bar{M})$.

Therefore, the time complexity of extracting a related private key from a signature is $O(\|\delta\|) > O(2^n)$, or $O(\bar{M}) > O(2^n)$.

6.2 Faking a Digital Signature only through a Public Key Is a Hardness

According to section 3.5, the discriminant $X \equiv Y$ (% M) contains the two variables Q and U of which one may be supposed in advance by an adversary. However, seeking the other through the supposed value is faced with a problem.

6.2.1 Transcendental Logarithm Problem

Assume that $g \in \mathbb{Z}_p^*$ with p being prime is a generator, then

$$\{y \mid y \equiv g^x \ (\% \ p), x = 1, \ldots, p-1\} = \mathbb{Z}_p^* \ [24].$$

Assume that k with $\gcd(k, p-1) = 1$ is an integer, then also

$$\{y \mid y \equiv x^k \ (\% \ p), x = 1, \ldots, p-1\} = \mathbb{Z}_p^* \ [24].$$

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

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

$$\{y \mid y \equiv x^x \ (\% \ p), x = 1, \ldots, p-1\} \ne \mathbb{Z}_p^*.$$

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

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

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

The above examples illustrate that $\{y \equiv x^x \ (\% \ p) \mid x = 1, \ldots, p-1\}$ 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 nonunique, and even inexistent. Thus, x^x has extremely strong irregularity, and is essentially distinct from g^x and x^k.

It should be noted that an attempt at solving $y \equiv x^x$ (% p) for x in light of the Chinese Remainder Theorem is specious. Refer to the following example.

Observe the congruent equation $4^4 \equiv 8 \equiv 3^{12}$ (% 31), where $3 \in \mathbb{Z}_{31}^*$ is a generator.

Try to seek x which satisfies $x \equiv 12$ (% 30) and $x \equiv 3$ (% 31) at one time, and verify whether $x \equiv 4$ (%

31) or not.

In light of the Chinese Remainder Theorem [21], let $m_1 = 30$, $m_2 = 31$, $a_1 = 12$, and $a_2 = 3$. Then

$$M = 30 \times 31 = 930,$$
$$M_1 = M / m_1 = 930 / 30 = 31,$$
$$M_2 = M / m_2 = 930 / 31 = 30.$$

Compute $y_1 = 1$ such that $M_1\, y_1 \equiv 1$ (% m_1).

Compute $y_2 = 30$ such that $M_2\, y_2 \equiv 1$ (% m_2).

Thereby,

$$x = a_1 M_1\, y_1 + a_2 M_2\, y_2 = 12 \times 31 \times 1 + 3 \times 30 \times 30 = 282 \text{ (\% 930)}.$$

It is not difficult to verify

$$282^{\,288} \equiv 8 \neq 4 \text{ (\% 31), and } 282^{\,288} \equiv 504 \neq 8 \neq 4 \text{ (\% 930)}.$$

The integer 282 is an element of the group ($\mathbb{Z}^*_{930}$, ·), and the element 4 of the group ($\mathbb{Z}^*_{31}$, ·) can not be obtained from 282, which is pivotal.

***Definition 5**:* Assume that p is a prime, and $y \in \mathbb{Z}^*_p$ is known. Then solving $y \equiv x^x$ (% p) for $x \in [1, p - 1]$ is called the transcendental logarithm problem, shortly TLP.

What needs to be emphasized is that TLP is more suitable for designing signature schemes due to the non-uniqueness of its solution.

Let $\hat{H}(y = f(x))$ represent the complexity or hardness of solving the problem $y = f(x)$ for x [30].

***Property 9**:* TLP is equivalent to or harder than DLP in the same prime field. The latter comparison means that TLP can not be solved in DLP subexponential time yet on the assumption that DLP can be solved.

Proof:

① Let $g \in \mathbb{Z}^*_p$ be a generator coprime to $p - 1$, which does not lose generality since g may be selected in practice.

Assume that $y \in \mathbb{Z}^*_p$ is known, and there is $y \equiv (g x)^x$ (% p).

Raising either side of the equation to the g-th power gives

$$y^g \equiv (g x)^{g x} \text{ (\% } p\text{)}.$$

Let

$$z \equiv y^g \text{ (\% } p\text{), and } w = g x,$$

where the latter is not a congruence, then

$$z \equiv w^w \text{ (\% } p\text{)}.$$

Suppose that $\bar{O}_T(y, p, Q)$ is an oracle of solving $y \equiv x^x$ (% p) for x, where Q is the set of all the possible values of x, namely the codomain, p is a prime modulus, and $y \in [1, p - 1]$.

Its output is $x \in Q$ (each of solutions if they exist), or 0 (no solution).

Let $Q_1 = \{1, 2, \ldots, p - 1\}$, and $Q_2 = \{1g, 2g, \ldots, (p - 1)g\}$.

Clearly, by calling $\bar{O}_T(y, p, Q_1)$, $y \equiv x^x$ (% p) is solved for x.

It is easily observed that between the limited sets Q_1 and Q_2, there is a linear bijection

$$\Gamma: Q_1 \rightarrow Q_2,\ \Gamma(a) = g a,$$

which means that the set Q_1 is equivalent to the set Q_2 [39]. Hence, substituting Q_1 with Q_2 as the codomain of a function will not increase the running time of $\bar{O}_T$.

Similarly, by calling $\bar{O}_T(z, p, Q_2)$, $z \equiv w^w$ (% p) is solved for w, namely all the satisfactory values of w are obtained.

Further, $x \equiv w g^{-1}$ (% p), or $x \equiv w g^{-1}$ (% $p - 1$).

Thereby, in terms of definition 3, there is

$$\hat{H}(y \equiv (g x)^x \text{ (\% } p\text{)}) \le \hat{H}\,(y \equiv x^x \text{ (\% } p\text{)}).$$

Namely the difficulty in inverting $y \equiv (g x)^x$ (% p) is not greater than that in inverting $y \equiv x^x$ (% p).

Again, suppose that $\bar{O}_{\check{T}}(y, g, p)$ is an oracle of solving $y \equiv (g x)^x$ (% p) for x, where p is a prime modulus, and $y, g \in [1, p - 1]$.

Its output is $x \in [1, p - 1]$ (each of solutions if they exist), or 0 (no solution).

Let $g = 1$.

By calling $\bar{O}_{\check{T}}(y, 1, p)$, the solution x to $y \equiv x^x$ (% p) will be obtained.

Thereby, in terms of definition 3, there is

$$\hat{H}(y \equiv x^x \text{ (\% } p\text{)}) \le_P \hat{H}(y \equiv (g x)^x \text{ (\% } p\text{)}).$$

In terms of definition 4, we have that

$$\hat{H}(y \equiv x^x \text{ (\% } p\text{)}) =_P \hat{H}(y \equiv (g x)^x \text{ (\% } p\text{)}).$$

That is to say, the difficulty in inverting $y \equiv (gx)^x$ (% p) is equivalent to that in inverting $y \equiv x^x$ (% p).

② The congruence $y \equiv (gx)^x$ (% p) may be written as $y \equiv g^x x^x$ (% p), where g is any generator.

Change $\bar{O}_{\hat{T}}(y, g, p)$ into $\bar{O}_{\hat{T}}(y, g, p, \hat{w})$, where sw = 0 or 1. Its structure is as follows:

S1: If $\hat{w} = 1$ and x to $y \equiv g^x x^x$ (% p) inexistent, return 'No'.

S2: If $\hat{w} = 1$,

S2.1: find y_1, and compute y_2 by $y \equiv y_1 y_2$ (% p),

S2.2: compute $x < p$ by $y_1 \equiv g^x$ (% p),

S2.3: if $y_2 \neq x^x$ (% p), goto S2.1;

else

S2.4: compute $x < p$ by $y \equiv g^x$ (% p).

S3: Return x.

Clearly, by calling $\bar{O}_{\hat{T}}(y, g, p, 0)$, the solution x to $y \equiv g^x$ (% p) will be obtained.

Therefore, still in terms of definition 3, there is

$$\hat{H}(y \equiv g^x \text{ (\% } p)) \leq_P \hat{H}(y \equiv g^x x^x \text{ (\% } p)).$$

Integrating ① and ②, we have that

$$\hat{H}(y \equiv g^x \text{ (\% } p)) \leq_P \hat{H}(y \equiv g^x x^x \text{ (\% } p)) =_P \hat{H}(y \equiv x^x \text{ (\% } p)),$$

namely inverting $y \equiv x^x$ (% p) is equivalent to or harder than inverting $y \equiv g^x$ (% p) for x. □

Additionally, let $y \equiv g^t$ (% p), and $x \equiv g^u$ (% p), and then it seems that there is $g^t \equiv g^{ug^u}$ (% p).

However due to g^u (% p) $\neq g^u$ (% $p - 1$), $y \equiv x^x$ (% p) can not be expressed as $t \equiv ug^u$ (% $p - 1$).

We can also understand that in the process of x being sought from $y \equiv x^x$ (% p), it is inevitable that the middle value of x is beyond p because modular multiplication, inverse, and power operations are inevitable.

Considering the middle value of x beyond p, let

$$z_1 = x \text{ \% } p \text{ with } z_1 < p, \text{ and } z_2 = x \text{ \% } (p - 1) \text{ with } z_2 < p - 1.$$

Then there are $x = z_1 + k_1 p = z_2 + k_2(p-1)$ and $z_1 = (z_2 - k_2)$ % p, where $k_1, k_2 \geq 0$ are two integers. Further, we have $y \equiv (g(z_2 - k_2))^{z_2}$ (% p), which indicates that due to x (% p) $\neq x$ (% $p - 1$) with $x > p$, the relation between x (% $p - 1$) and x (% p) is stochastic when x changes in the interval $(1, p^p)$.

Therefore, it is reasonable that letting $v \equiv g(z_2 - k_2)$ (% p), and we obtain $y \equiv v^{z_2}$ (% p).

If v is a constant, inverting $y \equiv v^{z_2}$ (% p) is equivalent to DLP. However, v will not be a constant forever. So, it should be impossible anyway that $\hat{H}(y \equiv (gx)^x \text{ (\% } p)) =_P \hat{H}(y \equiv g^x \text{ (\% } p))$.

The above evidence inclines us believe that there is

$$\hat{H}(y \equiv g^x \text{ (\% } p)) <_P \hat{H}(y \equiv g^x x^x \text{ (\% } p)),$$

namely on the assumption that DLP can be solved through an oracle, TLP can not be solved in DLP subexponential time yet.

The famous baby-step giant-step algorithm, Pollard's rho algorithm, Pohlig-hellman algorithm, and index-calculus algorithm for discrete logarithms [21] are ineffectual on transcendental logarithms. At present, there is no better method for seeking a transcendental logarithm than the exhaustive search, and thus the time complexity of solving $x^x \equiv c$ (% p) may be expected to be $O(p) > O(2^n)$, where n is the bit-length of a message digest.

Notice that for $y \equiv x^x$ (% p), there is no determinate relation between $\|y\|$ and $\|x\|$, namely $\|y\| \geq \|x\|$ or $\|y\| < \|x\|$. Therefore, in the case of a small modulus, x in $y \equiv x^x$ (% p) is still secure.

In REESSE1+, the form of TLP is $y \equiv (gx)^x$ (% p), where g is a constant. When the bit-length of the modulus is very small ─ 80 for example, the difference between the running times of solving $y \equiv (gx)^x$ (% p) and solving $y \equiv x^x$ (% p) is valuable because gx changes with g, and has more freedom than x, which makes the relation between $\|y\|$ and $\|x\|$ be more indeterminate.

6.2.2 Faking a Signature by the Verification Algorithm Is of TLP

Assume that F is any arbitrary file, H is its hash output, and (Q, U) is a signature on F. According to the discriminant $X \equiv Y$ (% M), namely

$$(\alpha Q^{-1})^{QUT} \alpha^{Q^n} \equiv (\bar{G}^Q U^{-1})^{UST} \beta^{HQ^{n-1} + H^n} \text{ (\% } M),$$

an attacker may suppose the value of any signature variable.

If suppose the value of Q, no matter whether U exists or not, seeking U is equivalent to TLP.

Similarly, if suppose the value of U, seeking Q is also equivalent to TLP.

If the attacker hits exactly the small $đ$, raises either side of the discriminant to the $đ$-th power, and assumes $Đ \mid (\delta Q - WH)$, then there is

$$(\alpha Q^{-1})^{dQUT} \equiv (\bar{G}^{Q} U^{-1})^{dUST} \ (\% \ M).$$

Further, let

$$(\alpha Q^{-1})^{dQT} \equiv (\bar{G}^{Q} U^{-1})^{dST} \ (\% \ M). \quad (6)$$

Now, suppose that Q is known, and U is unknown. If the equation

$$U^{dT} \equiv ((\alpha^{-1} Q)^{dQT})^{S^{-1}} \bar{G}^{QdT} \ (\% \ M)$$

has the trivial solution, work U out by theorem 2; otherwise work U out by the Index-calculus method. However, Q and U must satisfy the constraint $Đ \mid (\delta Q - WH)$, which is at most with the probability $1 / Đ < 1 / 2^n$ when δ and W are unknown since continuous integral values of Q can not guarantee the integral continuity of values of $(\delta Q - WH)$.

We observe that $(\alpha^{Q^n} \beta^{-(HQ^{n-1} + H^n)})^{đ} \equiv 1$ (% M), namely the element $\alpha^{Q^n} \beta^{-(HQ^{n-1} + H^n)} \in \check{S}$ which is the subgroup of order $đ$.

Assume that $\check{e}$ is a solution to $x^{đ} \equiv 1$ (% M), and g is a generator of $\mathbb{Z}_M^*$. Evaluate u, v, q by the Index-calculus such that $g^u \equiv \alpha$ (% M), $g^v \equiv \beta$ (% M), and $g^q \equiv \check{e}$ (% M) [21]. Then

$$g^{uQ^n} g^{-v(HQ^{n-1} + H^n)} \equiv g^q \ (\% \ M),$$

namely

$$uQ^n - vHQ^{n-1} - vH^n - q \equiv 0 \ (\% \bar{M}) \quad (7)$$

which is a true polynomial in Q.

If the polynomial has solutions, and Q can be figured out, U may be evaluated according to (6). In this way, Q and U which likely meet the original discriminant can be found. However, the Index-calculus method is completely ineffective on the high degree polynomial equation, and thus one solves it only through the probabilistic algorithm in section 1.6 of [38], of which the time complexity is $O(\bar{M} / n) > O(2^n)$.

Again because there is possibly $\gcd(u, \bar{M}) > 1$, namely (7) is not necessarily the polynomial of which the coefficient of the first term is 1, and there exists $\bar{M}^{1/n} \in (2^{696/80}, 2^{1216/128}) \approx (2^{8.7}, 2^{9.5})$, the Coppersmith reduction method that finds sufficiently small integer solutions, of which the absolute values are less than $\bar{M}^{1/n}$, to a modular univariate polynomial [40] is ineffectual on (7).

We also observe that on the condition that $đ$ is guessed accurately and $\gcd(U, \bar{M}) = 1$, there is

$$((\alpha Q^{-1})^{QT} (\bar{G}^{-Q} U)^{ST})^{đ} \equiv 1, \text{ or } ((\alpha Q^{-1})^{Q} (\bar{G}^{-Q} U)^{S})^{đT} \equiv 1 \ (\% \ M),$$

which implies that the element $(\alpha Q^{-1})^{QT} (\bar{G}^{-Q} U)^{ST} \in \check{S}$. Thereby, if gather many enough signature pairs (Q, U), all the elements of $\check{S}$ can be picked out. However, The analysis in section 6.1 shows that even if all the elements of the subgroup of order $đ$ can be found out, it does not influence the security of a REESSE1+ signature. Further, through gathering more enough signature pairs or following the Index-calculus method, all the elements of the subgroup of order $đT$ can be figured out and enumerated in the running time of $O(Đ) > O(2^n)$, or described with a general expression in the running time of $L_M[1/3, 1.923]$.

6.2.3 Faking a Signature by the Signature Algorithm Is of Exponential Time Complexity

Owing to $Q \equiv (R G_0)^S \delta \hbar$ (% M), $U^T \equiv (R W^{k-\delta})^{QT} \check{e}$ (% M), and $V \equiv (R^{-1} W^{\delta} G_1)^{QU} \delta^{\lambda}$ (% M), an adversary may attempt the following attack approach.

Let

$$Q \equiv a^S \delta \hbar \ (\% \ M), \ U^T \equiv b^{QT} \check{e} \ (\% \ M), \ V \equiv c^{QU} \delta^{\lambda} \ (\% \ M),$$

where λ meets

$$\lambda S \equiv ((WQ)^{n-1} + \xi + rUS)(\delta Q - HW) (\% \bar{M}).$$

Then, there are $a c \equiv (\alpha \delta^{-1} \hbar^{-1})^{1/S}$, and $b c \equiv \bar{G}$ (% M).

Firstly, if $\check{e}$ is hit, (Q, U) is a known signature, and $(U^T \check{e}^{-1})^{\bar{M}/(Tk)} \equiv 1$ (% M) holds, where $k = \gcd(Q, \bar{M})$, then resorting to the Index-calculus method, a solution b to the equation $b^{QT} \equiv U^T \check{e}^{-1}$ (% M) can be found, or the equation has the trivial root. Further, c can be figured from $b c \equiv \bar{G}$ (% M). However, it is impossible to find δ from $V \equiv c^{QU} \delta^{\lambda}$ (% M), and of course, it is impossible to find a from $a c \equiv (\alpha \delta^{-1} \hbar^{-1})^{1/S}$ (% M).

Secondly, when δ, $\hbar$ are found, if suppose a value of a, then c, b can be figured out. Further, when $Đ$ is factorized from $\bar{M}$, W is found, and r is guessed, the adversary may compute the values of Q and U which make

$$Đ \mid (\delta Q - WH) \ (\% \bar{M}),$$

$$đ \mid ((WQ)^{n-1} + \xi + rUS) \ (\% \bar{M}).$$

To seek δ and W from a known pivotal and clear clue, the attacker has to try to solve the simultaneous equations

$$\begin{cases} \alpha \equiv \delta^{(\delta^n + \delta W^{n-1})T} \ (\%\ M) \\ \beta \equiv \delta^{W^n T} \ (\%\ M). \end{cases}$$

Obviously, the first equation is at least equivalent to TLP. The second equation contains two variables, and belongs to nondeterministic problems. Raising either side of the first to the W-th power yields

$$\alpha^W \equiv \delta^{(\delta^n W + \delta W^n)T} \equiv \delta^{\delta^n W T} \beta^{\delta} \ (\%\ M),$$

which is still very complicated, and the problem is not simplified.

Let g be a generator of the group $(\mathbb{Z}_M^*, \cdot)$. By the Index-calculus for discrete logarithms [21], evaluate u, v, and x such that $g^u \equiv \alpha$, $g^v \equiv \beta$, and $g^x \equiv \delta\,(\%\ M)$ (Notice that this does not means $g^x \equiv \delta\,(\%\ \bar{M})$). Then, we obtain

$$\begin{cases} u \equiv x(\delta^n + \delta W^{n-1})T \ (\%\ \bar{M}) \\ v \equiv xTW^n \ (\%\ \bar{M}). \end{cases}$$

If $\gcd(xT, \bar{M}) \mid v$ and $\gcd(n, \phi(\bar{M})) = 1$, there exists the trivial root to $xTW^n \equiv v\ (\%\ \bar{M})$ (see theorem 1). However, even if W and x are known, seeking δ of a large value from the true polynomial $u \equiv x(\delta^n + \delta W^{n-1})T(\%\ \bar{M})$ is very arduous. If you are afraid of the Coppersmith reduction [40], in practice, the exponent n may be substituted with a larger integer.

In conformity with the key generation algorithm, there is $W \equiv (\prod_{i=1}^n A_i)^{-1}(\alpha\,\delta^{-1}\,\hbar^{-1})^{(S\,\delta)^{-1}}$ $(\%\ M)$. However, because $\prod_{i=1}^n A_i$, $\hbar$ are unknown, and W is at least the δ^{-1}-th power of $(\alpha\,\delta^{-1}\,\hbar^{-1})$, the substitution of W will not make the simultaneous equations reduced.

6.3 Faking a Signature through Known Signatures with a Public Key Is 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 $\bar{G}'$. Then, if any arbitrary (Q', U') satisfies

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

it is a signature fraud on F'.

Clearly, an adversary is allowed to utilize the known values of Q and U separately.

If let $Q' = Q$, Q' does not necessarily satisfy $Đ \mid (\delta Q' - WH')$, and computing U' is equivalent to TLP.

If let $U' = U$, no matter whether the discriminant has solutions or not, seeking Q' is also equivalent to TLP.

If the two signatures (Q_1, U_1) and (Q_2, U_2) on the files F_1 and F_2 are obtained, due to $Đ \mid (\delta Q_1 - WH_1)$ and $Đ \mid (\delta Q_2 - WH_2)$, we see that

$$Đ \mid (\delta(Q_1 + Q_2) - W(H_1 + H_2)).$$

Let $Q' = Q_1 + Q_2$, $H' = H_1 + H_2$, then $Đ \mid (\delta Q' - WH')$. However, inferring F' from H' is intractable in terms of the property of hash functions, and finding a fit U' from

$$U'^{T\bar{d}} \equiv ((\alpha^{-1} Q')^{Q' T S^{-1}} \bar{G}'^{Q' T})^{d} \ (\%\ M)$$

is also intractable since U' has $T\bar{d}$ values.

If many of the pair (Q, U) are known, because Q is random, Q and U interrelate through a transcendental logarithm, and the value of U changes intensely between 1 and M, there is no polynomial function or statistic regularity among different (Q, U), which means that they are unhelpful to solving TLP, but yet they is helpful to finding elements of the subgroup of order $\bar{d}$ or $\bar{d}T$ as is pointed out in section 6.1.

Thus, forging another signature through known signatures with a public key is of TLP or hash-hard.

6.4 Adaptive-chosen-message Attack Is Faced with Indistinguishability

Conforming to section 3.4, Q satisfies $Đ \mid \delta Q - WH$, namely $Q \equiv (\bar{a}Đ - WH)\delta^{-1}$ $(\%\ \bar{M})$, where $\bar{a}$ is random, and satisfies $\bar{d}T \nmid \bar{a}$.

The randomness of $\bar{a}$ leads Q to be random while U is interrelated with Q in a transcendental logarithm, where $Q \equiv (R\,G_0)^S \delta\hbar\ (\%\ M)$, and $U \equiv (R\,W^{k-\delta})^Q \delta^{\bar{a}Đ r}\ (\%\ M)$.

Hence, for an identical file F, there will be many different signatures on it. That is, the signature (Q, U) owns indistinguishability.

In terms of [27], the signature (Q, U) on F is secure against adaptive-chosen-message attack.

6.5 Chosen-signature Attack Is Faced with RSP and SPP

It is well understood from the discriminant that

$$\bar{G}^{QUST}\beta^{HQ^{n-1}+H^n} \equiv (\alpha Q^{-1})^{QUT}\alpha^{Q^n}U^{UST}\ (\%\ M). \quad (8)$$

Assume that the values of Q and U are chosen in advance, and an adversary attempts to figure out H and the corresponding file or message F.

Let $\bar{G} = f(H) = \prod_{i=1}^{n} C_i^{b_i}\ \%\ M$, where $H = b_1 \ldots b_n = \sum_{i=1}^{n} b_i 2^{i-1}$, then (8) is an equation in b_1, …, and b_n.

Moreover, let g be a generator of $(\mathbb{Z}_M^*, \cdot)$, and in terms of the Index-calculus method, work out $q_1, \ldots, q_n$, v, u, w such that

$$g^{q_1} \equiv C_1\ (\%\ M),\ \ldots,\ g^{q_n} \equiv C_n\ (\%\ M),\ g^v \equiv \beta\ (\%\ M),$$
$$g^u \equiv (\alpha Q^{-1})^{QUT}\alpha^{Q^n}U^{UST}\ (\%\ M),\ w \equiv QUST\ (\%\ \bar{M}).$$

Then, there is

$$(q_1 b_1 + \ldots + q_n b_n)w + v(Q^{n-1}\textstyle\sum_{i=1}^{n} b_i 2^{i-1} + (\sum_{i=1}^{n} b_i 2^{i-1})^n) \equiv u (\%\ \bar{M}),$$

which is the root seeking problem (RSP) of a high degree multivariate polynomial.

On the other hand, if there exists the inverse function $H = f^{-1}(\bar{G})$, namely H in (8) is substituted with $\bar{G}$, then evaluating $\bar{G}$ from (8) is the combination of DLP and RSP. Even if $\bar{G}$ is found out, due to $\bar{G} \equiv \prod_{i=1}^{n} C_i^{b_i}$ (% M), evaluating H from $\bar{G}$ is a subset product problem (see section 5.2.1).

Notice that in the verification algorithm, $\bar{G} \equiv \prod_{i=1}^{n} C_i^{b_i}$ (% M) is of SPP, and in the decryption algorithm, $\bar{G} \equiv \prod_{i=1}^{n} C_i^{b_i}$ (% M) is of ASPP (see section 3.2).

7 Conclusion

Because REESSE1+ is only a prototypal cryptosystem which is used for explaining some concepts, ideas, and methods, the space and time complexities of the five algorithms are not analyzed in the paper.

A REESSE1+ private key contains $2n + 5$ variables, but does not contain quadratic polynomials; thus REESSE1+ is a multivariate cryptosystem different from TTM and TTS.

In REESSE1+, not only numerical calculation ability but also logic judgement ability of a computer is utilized, and thus, the reversibility of the functions is relatively poorer.

MPP which contains indeterminacy is a compound problem integrating IFP and DLP, and ASPP is also a compound problem integrating IFP, DLP, and ASSP which can resist the L^3 lattice base reduction. TLP is a primitive problem. Evidences in the paper show that MPP, ASPP, and TLP are harder than DLP in the same prime field $\mathbb{GF}(M)$, namely MPP, ASPP, and TLP can not be solved in DLP subexponential time, which makes us believe that the polynomial-time algorithms for solving MPP, ASPP, and TLP do not exist on a quantum computational model while the polynomial-time algorithms for solving IFP and DLP exist [20]. Due to indeterminacy, even when $\log M \approx 80$, solving MPP is infeasible yet.

Among univariate functions, the transcendental logarithm problem $y \equiv x^x$ (% M) or $y \equiv (gx)^x$ (% M) and the root seeking problem of a true polynomial — $ax^n + bx^{n-1} + cx + d \equiv 0$ (% $\bar{M}$) with $n \ge 80$ for example are two primitive problems which can withstand any attacks except for brute forces so far even if $\log M \approx 80$. Notice that $\log M \approx 80$ indicates that the constraint $M > (\max_{1\le i\le n} A_i)^n$ is removed from the key generator, and REESSE1+ is only used for digital signature.

At present, the REESSE1+ cryptosystem is constructed in a multiplicative group $\mathbb{Z}_M^*$.

Suppose that M is a prime, then $\mathbb{Z}_M$ is a finite field with general addition and multiplication, and $\mathbb{Z}_M[x]$ is a Euclidean domain over $\mathbb{Z}_M$, namely a principle ideal domain and a uniquely factorial domain [24]. Additionally, we assume that $P(x) \in \mathbb{Z}_M[x]$ is an irreducible polynomial, of which the coefficient of the first term is the integer 1, then $\mathbb{Z}_M[x]\ /\ P(x)$ constitute a congruent Abelian group. Therefore, it is feasible to transplant REESSE1+ to the group $\mathbb{Z}_M[x]\ /\ P(x)$ from the group $\mathbb{Z}_M^*$.

From the dialectical viewpoint, it is impossible that a public key cryptosystem possesses all merits because some merits are possibly restrained by some others. Along with the development of CPU techniques and quantum computations, what people are more concerned about are the securities of cryptosystems, but not the lengths of parameters.

Clearly, it is worthy to be researched further how to decrease the length of a REESSE1+ modulus and to increase the speed of a REESSE1+ decryption.

Appendix A

What follows in rows is a program in MS Visual C++ for producing the set Q, namely Ω in section 4, and it contains the several nested loops.

```
/* Find an odd set Q such that for arbitrary e1, e2, e3, e4 belonging to Q, e1 + e2 + e3 != e4.
   Of course, there are e1 != e2 and e1 + e2 != e3. */
Void CkeyManagementPage::OnBtnGetAnOddSetQ( )
// CkeyManagementPage is the name of a class.
{
  DWORD i, j, max_s_2, max_s_3;
  DWORD *Q, e, g, n = 128;
  // n = 80, 96, 112, 128.
  BYTE     B[256 * 256];
  char       str[10];
  // Initialization
  Q = (DWORD *)calloc(2 * n, sizeof(DWORD));
  // The initial value of each element of Q is zero.
  Q[0] = 5; Q[1] = 7; Q[2] = 9;
  for (i = 0; i < 256 * 256; i++) B[i] = 0;
  B[12] = 2; B[14] = 2; B[16] = 2;
  /* That an element of B with an even index equals 2 means
     that the number denoted by the index itself is the sum of
     some two elements in Q. */
  B[21] = 3;
  /* That an element of B with an odd index equals 3 means
     that the number denoted by the index itself is the sum of
     some three elements in Q. */
  i = 2;
  max_s_2 = 16;
  // The maximum sum of two elements in Q.
  max_s_3 = 21;
  // The maximum sum of three elements in Q.
  while (i < 2 * n – 1) {
    e = Q[i] + 2;
    while (B[e] == 3) e += 2;
    i++;
    Q[i] = e;
    for (j = 12; j <= max_s_2; j += 2) {
      g = j + Q[i];
      if ((B[j] == 2) && (B[g] != 3)) {
        B[g] = 3;
        max_s_3 = (g > max_s_3) ? g : max_s_3;
      }
    }
    for (j = 0; j <= i – 1; j++) {
      g = Q[j] + Q[i];
      if (B[g] != 2) {
        B[g] = 2;
        max_s_2 = (g > max_s_2) ? g : max_s_2;
      }
    }
  }
  _ultoa(Q[2 * n – 1], str, 10);
  MessageBox(str, NULL, MK_OK);
  _ultoa(max_s_2, str, 10);
  MessageBox(str, NULL, MK_OK);
  _ultoa(max_s_3, str, 10);
  MessageBox(str, NULL, MK_OK);
  free(Q);
}
```

Appendix B — Offering a Reward

This paper shows that any effectual attack on REESSE1+ will be reduced to the solution of MPP, ASPP, TLP, and a true polynomial modulo a composite number.

Assume that M is a prime. As $n \ge 80$ and $\log M \ge 80$, it is well known that it is infeasible to find a large root to $ax^n + bx^{n-1} + cx + d \equiv 0$ (% $\bar{M}$) with $a \notin \{0, 1\}$, $b + c \ne 0$, and $d \ne 0$ [40].

According to the pragmatized aim, let $n = 80, 96, 112$, or 128 with $\log M = 384, 464, 544$, or 640, or with $\log M = 80, 96, 112$, or 128 as REESSE1+ is downsized to the lightweight.

Assume that $(\{A_i\}, \{\ell(i)\}, W, \delta, M)$ is a private key, and $(\{C_i\}, M)$ is a public key, where $W, \delta \in (1, \bar{M})$, $A_i \in \{2, 3, \ldots, 1201\}$, and $\ell(i) \in \{1, 3, \ldots, 2n - 1\}$ for $i = 1, \ldots, n$.

The authors promise solemnly that

① anyone who can extract the private key definitely from

$$C_i \equiv (A_i W^{\ell(i)})^{\delta} \ (\%\ M)$$

in DLP subexponential time will gain a reward of USD 100000, or USD 10000 as $\log M = 80, 96, 112$, or 128;

② anyone who can recover the plaintext $b_1 \ldots b_n$ definitely from the ciphertext

$$\bar{G} \equiv \prod_{i=1}^{n} C_i^{\underline{b}_i} \ (\%\ M)$$

in DLP subexponential time will gain a reward of USD 100000, or USD 10000 as $\log M = 80, 96, 112$, or 128, where $\underline{b}_i = 0$ if $b_i = 0$, 1 plus the number of successive 0-bits before b_i if $b_i = 1$, or 1 plus the number of successive 0-bits before and after b_i if b_i is the rightmost 1;

③ anyone who can find a large answer $x \in (1, \bar{M})$ definitely to

$$y \equiv (gx)^x \ (\%\ M)$$

with known $g, y \in (1, \bar{M})$ in DLP subexponential time will gain a reward of USD 100000, or USD 10000 as $\log M = 80, 96, 112$, or 128.

Of course, any solution must give a formal process, can be verified with our examples, and is subject to a CPU speed with regard to n and $\log M$.

Acknowledgment

The authors would like to thank the Academicians Jiren Cai, Changxiang Shen, Zhongyi Zhou, Zhengyao Wei, Andrew C. Yao, Binxing Fang, and Xicheng Lu for their important guidance, suggestions, and helps.

Also would like to thank the Professors Dingyi Pei, Dengguo Feng, Zejun Qiu, Jie Wang, Ronald. L. Rivest, Moti Yung, Alfred J. Menezes, Dingzhu Du, Xuejia Lai, Yongfei Han, Mulan Liu, Huanguo Zhang, Dake He, Maozhi Xu, Yixian Yang, Jianfeng Ma, Xiaoyun Wang, Kefei Chen, Yupu Hu, Qibin Zhai, Haiwen Ou, Chao Li, Wenbao Han, Dongqing Xie, Guoqiang Bai, Dongdai Lin, Lei Hu, Rongquan Feng, Ping Luo, Lusheng Chen, Zhiying Wang, and Quanyuan Wu for their important advice, suggestions, and corrections.

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*, vol. 21, no. 2, pp. 120-126, Feb. 1978.

[2] T. ElGamal, “A Public-key Cryptosystem and a Signature Scheme Based on Discrete Logarithms,” *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469-472, Jul. 1985.

[3] W. Diffie and M. E. Hellman, “New Directions in Cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, Nov. 1976.

[4] O. Goldreich, *Foundations of Cryptography: Basic Tools*. Cambridge, UK: Cambridge University Press, 2001, ch.1-2.

[5] A. C. Yao, “Theory and Applications of Trapdoor Functions,” in *Proc. the 23rd Annual Symposium on the Foundations of Computer Science*, IEEE, 1982, pp. 80-91.

[6] S. Vadhan, *Computational Complexity* [online]. http://eecs.harvard.edu/~salil/papers/encyc.pdf, Jul. 2004.

[7] M. Bellare and P. Rogaway, “The Exact Security of Digital Signatures — How to Sign with RSA and Rabin,” in *Proc. Advance in Cryptology: Eurocrypt ’96*, Springer-Verlag, 1996, pp. 399-416.

[8] R. C. Merkle and M. E. Hellman, “Hiding information and Signatures in Trapdoor Knapsacks,” *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 525-530, Sep. 1978.

[9] A. Shamir, “A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem,” in *Proc. the 23th IEEE Symposium on the Foundations of Computer Science*, 1982, pp. 145-152.

[10] E. F. Brickell, “Solving Low Density Knapsacks,” in *Proc. Advance in Cryptology: CRYPTO ’83*, Plenum Press, 1984, pp. 25-37.

[11] M. J. Coster, A. Joux, and B. A. LaMacchia etc., “Improved Low-Density Subset Sum Algorithms,” *Computational Complexity*, vol. 2, issue 2, pp. 111-128, Dec. 1992.

[12] C. P. Schnorr and M. Euchner, “Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems,” *Mathematical Programming*: Series A and B, vol. 66, issue 2, pp. 181-199, Sep. 1994.

[13] I. F. Blake, G. Seroussi, and N. P. Smart, *Elliptic Curves in Cryptography*. Cambridge, UK: Cambridge University Press, 1999, ch. 1, 6.

[14] S. Y. Yan, *Number Theory for Computing* (2nd ed.). New York: Springer- Verlag, 2002, ch. 1, 3.
[15] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A Ring-based Public Key Cryptosystem," in *Proc. the Algorithmic Number Theory Symposium - ANTS III*, Springer-Verlag, 1998, pp. 267-288.
[16] T. Moh, "An Application of Algebraic Geometry to Encryption: Tame Transformation Method," *Revista Matemática Iberoamericana*, vol. 19, no. 2, pp. 667-685, May 2003.
[17] J. M. Chen and B. Y. Yang, "A More Secure and Efficacious TTS Signature Scheme," in *Proc. 6th Int. Conference on Information Security & Cryptology* (ICISC '03), Springer-Verlag, 2003, pp. 320-338.
[18] S. Su, "The REESSE1 Public-key Cryptosystem," *Computer Engineering & Science* (in Chinese), vol. 25, no.5, pp.13-16, Sep. 2003.
[19] S. Liu, F. Zhang, and K. Chen, "*Cryptanalysis of REESSE1 Digital Signature Algorithm*," in *Proc. CCICS 2005*, Xi'an, China, May 2005, pp. 200-205.
[20] P. W. Shor, "Polynomial-time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM J. on Computing*, vol. 26, issue 5, pp.1484-1509, Oct. 1997.
[21] A. J. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. London, UK: CRC Press, 1997, ch. 2, 3, 8.
[22] P. Garrett, *Making, Breaking Codes: An Introduction to Cryptology*. New Jersey: Prentice-Hall, 2001, ch. 12.
[23] S. Su and S. Lü, "To Solve the High Degree Congruence x ^ n ≡ a (mod p) in GF(p)," in *Proc. of Int. Conference on Computational Intelligence and Security*, IEEE Computer Society, Dec. 2007, pp.672-676.
[24] T. W. Hungerford, *Algebra*. New York: Springer-Verlag, 1998, ch. 1-3.
[25] K. H. Rosen, *Elementary Number Theory and Its Applications* (5th ed.). Boston: Addison-Wesley, 2005, ch. 12.
[26] D. Z. Du and K. Ko, *Theory of Computational Complexity*. New York: John Wiley & Sons, 2000, ch.3-4.
[27] M. Bellare and P. Rogaway, "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols," in *Proc.* the 1st *ACM Conference on Computer and Communications Security*, New York: ACM Press, 1993, pp. 62-73.
[28] R. Canetti, O. Goldreich, and S. Halevi, "The Random Oracle Methodology Revisited, " in *Proc. the 30th Annual ACM Symposium on Theory of Computing*, New York: ACM Press, 1998, pp. 209-218.
[29] D. R. Stinson, *Cryptography: Theory and Practice* (2nd ed.). London, UK: CRC Press, 2002, ch. 6.
[30] M. Davis, *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions*. Mineola: Dover Publications, 2004, ch. 2-4.
[31] H. Ritter, "Breaking Knapsack Cryptosystems by Max-Norm Enume- ration," in *Proc. 1st International Conference of the Theory and Application of Cryptology - Pragocrypt '96*, 1996, pp. 480-492.
[32] V. Niemi, "A New Trapdoor in Knapsacks," in *Proc. Advances in Cryptology: EUROCRYPT '90*, Springer-Verlag, 1991, pp. 405-411.
[33] C. P. Schnorr, "Fast LLL-Type Lattice Reduction," *Information and Computation*, vol. 204, no. 1, pp. 1-25, 2006.
[34] P. Q. Nguên and D. Stehlé, "Floating-Point LLL Revisited," in *Proc. Advances in Cryptology: EUROCRYPT '05*, Springer-Verlag, 2005, pp. 215-233.
[35] D. Bleichenbacher, "Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1," in *Proc. Advance in Cryptology: Crypto '98*, Springer-Verlag, 1998, pp. 1-12.
[36] R. Cramer and V. Shoup, "A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack," in *Proc. Advance in Cryptology: Crypto '98*, Springer-Verlag, 1998, pp. 13-25.
[37] V. Shoup, "OAEP Reconsidered," in *Proc. Advance in Cryptology: Crypto '01*, Springer-Verlag, 2001, pp. 239-259.
[38] H. Cohen, *A Course in Computational Algebraic Number Theory*. New York: Springer-Verlag, 2000, ch. 1, 3.
[39] R. R. Stoll, *Set Theory and Logic*. Mineola: Dover Publications, 1979, ch. 2-3.
[40] D. Coppersmith, "Small Solutions to Polynomial Equations and Low Exponent RSA Vulnerabilities," *Journal of Cryptology*, vol. 10, no. 4, pp. 223-260, Nov. 1997.

Shenghui Su received a bachelor degree in computer science from National University of Defense Technology, a master degree from Peking University, and a Ph.D. degree from University of Science and Technology Beijing. He was given the title of a senior programmer by Ministry of Electronic Industry China in 1994, and of a professor by Academic and Technologic Committee Beijing in 2004. He is a senior member of China Computer Federation, and of China Association for Cryptologic Research. He was long engaged in design of algorithms and development of software. Since 2000, he has been indulged in designing the new cryptosystems. He is currently with college of computer science, Beijing University of Technology. His research area covers computational complexity, cryptographic algorithms, and trusted systems.

Shuwang Lü received a bachelor degree in electronics from University of Science and Technology China. He is a senior member of China Association for Cryptologic Research, and sits on the academic committee of SKLOIS. Since 1980, he has mainly been making researches on cipher algorithms and cipher chips. He presided over the projects supported by the national plan of high technology development, and of fundamental science researches, and was the chief designer of the SMS4 symmetric cryptosystem, an industrial standard. He is with the School of Graduate, Chinese Academy of Sciences, also with University of Science and Technology China, and with USTB as a professor and a Ph.D. advisor. Nowadays, his research interests are still focused on cryptographic algorithms and knowledge security.