Design and Analysis of the REESSE1+
Public Key Cryptosystem v2.21

Shenghui Su and Shuwang Lü

1 College of Computers, Beijing University of Technology, Beijing 100124, P.R.China
2 Graduate School, Chinese Academy of Sciences, Beijing 100039, P.R.China

Abstract: In this paper, the authors give the definitions of a coprime sequence and a lever function, and describe the five algorithms and six characteristics of a prototypal public key cryptosystem which is used for encryption and signature, and based on three new problems and one existent problem: the multivariate permutation problem (MPP), the anomalous subset product problem (ASPP), the transcendental logarithm problem (TLP), and the polynomial root finding problem (PRFP). Prove by reduction that MPP, ASPP, and TLP are computationally at least equivalent to the discrete logarithm problem (DLP) in the same prime field, and meanwhile find some evidence which inclines people to believe that the new problems are harder than DLP each, namely unsolvable in DLP subexponential time. Demonstrate the correctness of the decryption and the verification, deduce the probability of a plaintext solution being nonunique is nearly zero, and analyze the exact securities of the cryptosystem against recovering a plaintext from a ciphertext, extracting a private key from a public key or a signature, and forging a signature through known signatures, public keys, and messages on the assumption that IFP, DLP, and LSSP can be solved. Studies manifest that the running times of effectual attack tasks are greater than or equal to $O(2^n)$ so far when $n = 80, 96, 112$, or $128$ with $\lg M = 696, 864, 1030$, or $1216$. As viewed from utility, it should be researched further how to decrease the length of a modulus and to increase the speed of the decryption.

Keywords: Public key cryptosystem; Coprime sequence; Lever function; Bit shadow; Digital Signature; Double congruence theorem; Transcendental logarithm problem; Provably Security; Polynomial time Turing 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 utilization of a cryptosystem feasible and breaking of the cryptosystem infeasible in polynomial time [6]. Taking RSA based on the integer factorization problem (IFP) as an example, when the bit-length of a RSA modulus reaches 1024, attack is infeasible but 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 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 cannot prevent a related private key from being extracted from a public key. For instance, in the MH knapsack cryptosystem [8], the subset sum problem (SSP) which contains the trapdoor information cannot prevent a MH private key from being extracted from $c_i = a_i W \pmod{M}$ by the method of accumulating points of minima [9], and moreover, when a related knapsack density is less than 1, the low-density SSP (LSSP) 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 the elevation of a computer’s CPU speed, the security dominant parameter of a cryptosystem becomes larger and larger. For instance, the bit-length of modulus of the ElGamal cryptosystem based on the discrete logarithm problem (DLP) is already up to 1024. We think that there are currently four manners of decreasing the bit-lengths of security dominant parameters and meantime increasing the one-wayness of related trapdoor functions.

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 generic elliptic curve discrete logarithms in time being subexponential 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 whose security still relies on the two large prime factors [14] for example.

The second manner is to design cryptosystems over polynomial rings — the NTRU cryptosystem for example. The shortest vector problem (SVP) is the security bedrock of NTRU since it is impossible to seek an NTRU secret polynomial or an NTRU plaintext polynomial through the $L^3$ lattice base reduction on condition that two special parameters $c_b$ 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 discovering or constructing new one-way computational problems which should be harder than IFP, DLP, or LSSP in light of the found evidence. Some threads of the fourth manner are given in this paper.
The paper has five novelties: ⋄ gives the definitions and properties of a coprime sequence, a lever function, and a bit shadow; ⋄ offers three new computational problems—the multivariate permutation problem (MPP) $C_1 = (A, \mathcal{W}^{[1]} \delta) \% M$, the anomalous subset product problem (ASPP) $\mathcal{G} = \prod_{i=1}^{n} C_i \% M$, and the transcendental logarithm problem (TLP) $y = x^r \% M$, proves that the three problems are computationally at least equivalent to DLP in the same prime field, and finds some evidence which inclines people to believe that the three problems are separately harder than DLP; ⋄ over a prime field, designs the five algorithms of a prototypical public key cryptosystem called REESSE1+; ⋄ analyzes the security of REESSE1+; ⋄ proposes and proves the double congruence theorem.

MPP which owns the indeterminacy assures the security of a private key and the signature algorithm. ASPP as a trapdoor function which can resist the L3 lattice base of a private key and the signature algorithm. ASPP as a primitive problem cannot be converted naturally, whether MPP, ASPP, and TLP can be solved in multivariate cryptosystem because only multiple variables must exist a prime divisor which does the opposite. Additionally, assume that $A_1, A_2, A_3$ are pairwise distinct positive integers such that $\forall A_1, A_2 (i \neq j)$, either $\gcd(A_1, A_2) = 1$ or $\gcd(A_1, A_2) = F \neq 1$ with $(A_1 / F) \% A_i$ and $(A_2 / F) \% A_i \forall k \neq i$, $j \in [1, n]$, these integers are called a coprime sequence, denoted by $\{A_1, \ldots, A_n\}$, and shortly $\{A\}$.

Notice that the elements of a coprime sequence are not necessarily pairwise coprime, but a sequence whose elements are pairwise coprime is a coprime sequence.

Property 1: Let $\{A_1, \ldots, A_n\}$ be a coprime sequence. If randomly select $m \in [1, n]$ elements from $\{A_1, \ldots, A_n\}$, and construct a subsequence $\{A_{n_1}, \ldots, A_{n_m}\}$ also called a subset, then the subset product $G = \prod_{i=1}^{m} A_i = A_{n_1} \cdots A_{n_m}$ is uniquely determined, namely the mapping from $\{A_{n_1}, \ldots, A_{n_m}\}$ to $G$ is one-to-one.

Proof: By contradiction. Presuppose that $G$ is acquired from two different subsequences $\{A_{n_1}, \ldots, A_{n_m}\}$ and $\{A_1, \ldots, A_s\}$, namely

$$G = \prod_{i=1}^{m} A_i = A_{n_1} \cdots A_{n_m} = \prod_{i=1}^{s} A_i = A_1 \cdots A_s.$$

Since the two subsequences are unequal, there must exist a certain element $A_m$ which does not belong to the two subsequences at one time. Without loss of generality, let $A_s \in \{A_{n_1}, \ldots, A_{n_m}\}$ and $A_m \notin \{A_1, \ldots, A_s\}$.

By the fundamental theorem of arithmetic [14], there must exist a prime $q$ which is a divisor of $A_m$.

Firstly, assume that $\forall A_i, A_j \in \{A_1, \ldots, A_s\}$, $\gcd(A_i, A_j) = 1$, namely $A_1, \ldots, A_s$ are pairwise coprime.

Then, there does not exist a common prime divisor between any two elements, which manifests that the prime divisors of every element do not belong to any other elements.

Thus, $q$ must be a divisor of $\prod_{i=1}^{m} A_i$, but not a divisor of $\prod_{i=1}^{s} A_i$, which means that the integer $G$ has two distinct prime factorizations, and is in direct contradiction to the fundamental theorem of arithmetic.

Secondly, assume that $\exists A_{n_h}, A_s \in \{A_1, \ldots, A_s\}$ with $\gcd(A_{n_h}, A_s) \neq 1$.

According to Definition 1, $\forall A_i \in \{A_1, \ldots, A_s\}$ with $r \neq s$, $i$, there are

$$(A_i / \gcd(A_i, A_s)) \% A_i 	ext{ and } (A_i / \gcd(A_i, A_s)) \% A_s,$$

which means that both at least one divisor of $A_i$ and at least one divisor of $A_s$ are not contained in any other elements.

Let $z = s$ or $t$, and $q$ be a prime divisor of $A_i$, with $q \% A_i \forall r \neq z \in [1, n]$.

Notice that from the above assignment, we know $A_z \in \{A_{n_1}, \ldots, A_{n_m}\}$ and $A_s \notin \{A_{n_1}, \ldots, A_{n_m}\}$.

Then, there are $q \% \prod_{i=1}^{m} A_i = G$ and $q \% \prod_{i=1}^{s} A_i = G$, which is in direct contradiction.

In sum, the mapping between $G$ and $\{A_{n_1}, \ldots, A_{n_m}\}$ is one-to-one.

Definition 2: Let $b_1, \ldots, b_h \neq 0$ be a bit string. Then $b_i$ with $i \in [1, n]$ is called a bit shadow if it is produced by such a rule: $b_i$ equals 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 $b_i$ if $b_i = 1$.
before and after $b_j$ if $b_j$ is the rightmost 1-bit.

**Fact 1:** Let $b_1, \ldots, b_n \neq 0$ be a bit string. Then there is $\sum_{i=1}^n b_i = n$.

**Proof:** According to Definition 2, every bit of $b_1, \ldots, b_n$ is considered into $\sum_{i=1}^n b_i$, where $b_1, \ldots, b_n$ are 1-bit shadows in string $b_1, \ldots, b_n$, and there is $\sum_{i=1}^n b_i = n$.

On the other hand, there is $\sum_{i=1}^n b_i = 0$, where $b_1, \ldots, b_n$ are 0-bit shadows.

In total, there is $\sum_{i=1}^n b_i = n$. □

**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 $b_1, \ldots, b_n$ to $G = \prod_{i=1}^n A_i^b_i$ is one-to-one.

**Proof:** Let $b_1, \ldots, b_n$ and $b_1', \ldots, b_n'$ be two different bit strings, and separately correspond to $b_1, \ldots, b_n$ and $b_1', \ldots, b_n'$. If $b_1, \ldots, b_n = b_1', \ldots, b_n'$ it is not difficult to understand $b_1, \ldots, b_n = b_1', \ldots, b_n'$. So, the mapping from $b_1, \ldots, b_n$ to $b_1, \ldots, b_n$ is one-to-one.

Additionally, since $A_i^b_i$ is not equal to 1 and contains the same prime factors as those of $A_i$, it is known from the proof of Property 1 that the mapping from $b_1, \ldots, b_n$ to $G = \prod_{i=1}^n A_i^b_i$ is one-to-one.

Therefore, the mapping from $b_1, \ldots, b_n$ to $G = \prod_{i=1}^n A_i^b_i$ is one-to-one. □

**Definition 3:** The secret $\ell(i)$ in the key transform of a public key cryptosystem is called a lever function, if it has the following features:

1. $\ell(i)$ is an injection from the domain $\{1, \ldots, n\}$ to the codomain $\Omega \subset \{1, \ldots, M\}$;
2. the mapping between $i$ and $\ell(i)$ is established randomly without an analytical expression;
3. an attacker has to be faced with all the permutations of elements in $\Omega$ when extracting a related private key from a public key;
4. the owner of a private key only needs to consider the accumulative sum of elements in $\Omega$ when recovering a related plaintext from a ciphertext.

Feature 2 and 3 make it clear that if $n$ is large enough, it is feasible for an attacker to search all the permutations of elements in $\Omega$ exhaustively while decryption is feasible in time being polynomial in $n$. Thus, the amount of calculation on $\ell(i)$ at ‘a public terminal’ is large, and the amount of calculation on $\ell(i)$ at ‘a private terminal’ is small.

Concretely to REESSE1+, $\ell(i)$ in the transform $C_i = (A_i, W_i^\ell(\ell(i)))^0(y)$ (% M) for $i = 1, \ldots, n$ is on the position of an exponent.

**Property 3 (Indeterminacy of $\ell(i)$):** Let $\delta = 1$ and $C_i = A_i, W_i^\ell(\ell(i))$ (% M) for $i = 1, \ldots, n$, where $\ell(i) \in \Omega \subset \{1, \ldots, n\}$ and $A_i \in A = \{2, \ldots, P\}$. Then for each $W_i \in [1, \ldots, M]$ with $\|W_i\| \neq \Omega$, and $\forall x, y, z \in [1, \ldots, n]$ with $x = x, y$,

- when $\ell(x) + \ell(y) = \ell(z)$, there is $\ell(x) + \|W_i\| + \ell(y) + \|W_i\| \neq \ell(z) + \|W_i\|$ (% M);
- when $\ell(x) + \ell(y) \neq \ell(z)$, there always exist $C_i = A_i, W_i^\ell(\ell(i))$, $C_i = A_i, W_i^\ell(\ell(i))$, and $C_i = A_i, W_i^\ell(\ell(i))$ (% M)

such that $\ell(x) + \ell(y) = \ell(z)$ (% M) with $A_i \leq D$.

**Proof:**

- It is easy to understand that $W_i^\ell(x) \neq W_i^\ell(x)' \neq \|W_i\|$, $W_i^\ell(y) = W_i^\ell(y) \neq \|W_i\|$, $W_i^\ell(z) = W_i^\ell(z) \neq \|W_i\|$, $\|W_i\| \neq \|W_i\|$.

Due to $\|W_i\| \neq \Omega$, $2\|W_i\| \neq \|W_i\|$, and $\ell(x) + \ell(y) = \ell(z)$, it follows that $\ell(x) + \|W_i\| + \ell(y) + \|W_i\| \neq \ell(z) + \|W_i\|$ (% M).

However, it should be noted that when $\|W_i\| = \Omega$, there is $\ell(x) + \|W_i\| + \ell(y) + \|W_i\| = \ell(z) + \|W_i\|$ (% M).

- Let $\delta$ be an oracle on seeking a discrete logarithm from DLP.

Suppose that $W' \in [1, \ldots, M]$ is a generator of $(\mathbb{Z}_M, \cdot)$. In light of group theories, $\forall A_i, A_i \in [2, \ldots, P]$, the congruence

$$C_i = A_i, W_i^\ell(\ell(i)) \neq \|M\|$$

has a solution. Then, $\ell(i)$ may be taken through $\delta$.

Thus, for $C_i = A_i, W_i^\ell(\ell(i))$ (% M) and $C_i = A_i, W_i^\ell(\ell(i))$ (% M), we can obtain many distinct pairs $(A_i, A_i)$, where $A_i, A_i \in [1, \ldots, M]$, and $\ell(x) + \ell(y) = \ell(z)$ (% M). □

According to Definition 2,  every bit of $\delta$ is on the position of an exponent.

- Notice that letting $\Omega = \{5, \ldots, n + 4\}$, namely every $\ell(i)$ makes seeking $W'$ from $W_i^\ell(\ell(i)) = A_i, C_i$ (% M) face an unsolvable Galois group when $A_i$ is guessed [22], and especially when $\Omega$ is any subset containing $n$ elements of $\{1, \ldots, M\}$, Property 3 still holds.

Assume that $\ell(x) + \ell(y) = \ell(z)$, and let $G' = C_i, C_i, C_i^{-1}$ (% M). Then

$$G' / M - L / A_i = (A_i, A_i) / (M, A_i)$$

where $L$ is a positive integer.

Due to $M > \prod_{i=1}^n A_i, A_i \geq 2$, and $n > 3$, there is

$$G' / M - L / A_i < 1 / (2^{n-3} A_i^2) < 1 / (2 A_i^2)$$

By Theorem 12.19 in Section 12.3 of [23], $L / A_i$ is a convergent of the continued fraction expansion of $G' / M$.

Property 3 illuminates that $\ell(x) + \ell(y) = \ell(z)$ is not necessary for the above discriminant. Therefore, a continued fraction attack on $C_i = A_i, W_i^\ell(\ell(i))$ (% M) according to the discriminant will be ineffectual as long as $A$ and $\Omega$ are fitly selected [24]. However, the robust $A$ and $\Omega$ will make the decryption of a ciphertext get slow. Hence, $C_i = A_i, W_i^\ell(\ell(i))$ (% M) is not an efficient key transform.

3 Design of the REESSE1+ Public key Cryptosystem

In essence, REESSE1+ is a prototypical cryptosystem which is used to expound some foundational concepts, ideas, and methods.

3.1 The Key Generation Algorithm

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

Let $p_1, \ldots, p_n$ be the first $n$ primes in the set $\mathbb{P}$, $A = \{2, 3, \ldots, 1201\}$, and $\Omega = \{5, 7, \ldots, 2n + 3\}$. Assume that $d, D$, $T, S$ are four pairwise coprime integers, where $d \in \{5, 2^{10}\}$, $T \geq 2^{10}$, and $D$ contains a prime not less than $2^9$.

S1: Randomly generate a coprime sequence $\{A_1, \ldots, A_n\}$ with $A_i \in A$.

S2: Find a prime $M > \prod_{i=1}^n A_i^{x_i}$ making $(dDT) / M$ gcd$(S, M) = 1$ and $\prod_{i=1}^n p_i / M$, where $k$ meets $\prod_{i=1}^n e_i \geq 2^{10}$ and $p_i \neq 2 n$.

S3: Pick $W, \delta \in (1, M)$ making gcd$(\delta, M) = 1$, $\|\delta\| = dDT$, and $\|\delta\| \geq 2^{10}$.

S4: Compute $\alpha \leftarrow 2^{\delta+2^{10}}$, $\beta \leftarrow 2^{\delta+\ell(\delta, M)}$. % M,
h ← (W\prod_{i=1}^{n} A_i) \beta (α^\delta) \% M.

S5: Randomly produce pairwise distinct l(1), ..., l(n) ∈ Ω

S6: Compute Ci ← (A_iW^{l(i)})\beta \% M for i = 1, ..., n.

At last, regard (|Ci|, α, β) as a public key, (|A_i|, l(i), W, δ, D, h, d) as a private key, and (S, T, M) as being in common.

Notice that if REESSE1+ is pragmatized, we suggest that the set Ω = {+/−5, +/−7, ..., +/−(2n + 3)}, where every sign +/− means that + or − is selected, and unknown to the public, which may bring more indeterminacy, and that gcd(W, D) > 1, which may avoid the existence of W^{-1} \% D no matter what the value of D is.

At S3, to seek δ, first let δ = g^{W(\delta D)}(\% M), where g is a generator by Algorithm 4.80 in Section 4.6 of [25], then test δ. At S4, seeking a S-th root to x^3 = c (\% M) is referred to Theorem 1 in Section 3.4.

Considering M > (\text{max}_{i \in A_i} A_i)^n and the fact that the first n primes in the set |A_i| can constitute a smallest coprime sequence, we can estimate lgM ≈ 696, 864, 1030, or 1216 when n = 80, 96, 112, or 128.

**Definition 4:** Let {C_1, ..., C_n} be a non-coprime sequence, and M be a prime. Seeking the original {A_1, ..., A_n} with A_i ∈ A, l(1), ..., l(n) with l(i) ∈ Ω, W, δ from C_i = (A_iW^{l(i)})\beta (\% M) for i = 1, ..., n is called the multivariate permutation problem, shortly MPP.

### 3.2 The Encryption Algorithm

Assume that {\{C_i\}, α, β} is a public key, and b_1...b_n \neq 0 is a plaintext block or a symmetric key.

S1: Set δ = 1, k = 0, i = 1.

S2: If \( b_i = 0 \) do k ← k + 1, b_i ← 0; else do b_i ← k + 1, k ← 0, δ ← δC_i^\beta (\% M).

S3: Let i ← i + 1.

If i ≤ n, go to S2.

S4: If b_0 = 0, do k ← k + 1, b_i ← (C_i^\beta) (\% M).

So, the ciphertext δ = |C_i|^\beta (\% M) is obtained.

Notice that α and β are useless for the encryption.

**Definition 5:** Let {C_1, ..., C_n} be a non-coprime sequence, and M be a prime. Seeking the original b_1...b_n from δ_i = (|C_i|^\beta) \% M is called the (modular) subset product problem, shortly SPP.

**Definition 6:** Let {C_1, ..., C_n} be a non-coprime sequence, and M be a prime. Seeking the original b_1...b_n, namely b_1...b_n from δ = |C_i|^\beta (\% M) is called the anomalous subset product problem, shortly ASPP.

### 3.3 The Decryption Algorithm

Assume that {\{A_i\}, l(i), W, δ, D, d, h} is a related private key, and δ is a ciphertext.

Notice that because \( \sum_{i=1}^{n} b_i = n \) is even, \( \sum_{i=1}^{n} b_i l(i) \) must be even.

S1: Compute δ ← δ^\beta (\% M).

S2: Compute δ ← δW^2 (\% M).

S3: Set b_1...b_k ← 0, G ← δ, i ← 1, k ← 0.

S4: If A_i^{l(i)+1} \% G, do G ← G / A_i^{l(i)+1}, b_i ← 1, k ← 0; else let k ← k + 1.

S5: Let i ← i + 1.

If i ≤ n and G \neq 1, go to S4.

S6: If k \neq 0 and (A_i^{l(i)+1} \% G), do G ← G / (A_i^{l(i)+1}).

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

So, the original plaintext block or symmetric key b_1...b_n is recovered.

Only if δ is a true ciphertext, can this algorithm terminate normally. In decryption, \( l(i), D, d, \) and h are unhelpful.

### 3.4 The Digital Signature Algorithm

Assume that {\{A_i\}, l(i), W, δ, D, d, h} is a private key, F is a file or message to be signed, and hash is a one-way compression function.

S1: Let H ← hash(F), whose binary form is b_1b_2b_3...

S2: Set k ← δS_\sum_{i=1}^{n} b_i (\% M), G_0 ← (|C_i|^\beta) \% M.

S3: \forall u \in \Omega (\% M) making (\delta D) \% M, \( G \equiv (\delta D + W)H \delta \% M \).

S4: Compute \( R ← \left( Q(\delta h) \right)^{\delta} G_0 \), \( U ← (R^{W+\delta}D) \% M \), \( \delta_{\% M} \), \( \bar{ξ} = \sum_{i=0}^{n} (\delta_{\% M}^{i+1}) (\% M) \).

S5: \forall r \in \{1, 2^{216}\} making \( \delta_{\% M} \), \( U = U \delta \% M \).

S6: If \( \delta(U^{W+\delta}+ξ + rUS) \% M \), goto S5; else end.

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

It is known from S3, S4 that \( Q, R = \delta D = \frac{Q}{δ} = \frac{\delta D}{δ} = \frac{δD}{δ} \% M \).

According to the double congruence theorem (see Section 3.6), in the signature algorithm we do not need \( V = (R^{W+\delta}D)G_{1\% M} \% M \), where \( G_1 = (|\prod_{i=1}^{n} |C_i|^\beta) \% M \), and λ satisfies \( λS = ((WQ)^{\% λ} + ξ + rUS) (δQ \% M) \), which indicates \( (δD) \% λ \).

At S5, the probability of finding a fit U is roughly 1/d. Because λ is a small number, U can be found out at a good pace. The small d, however, does not influence the security of REESSE1+ (see Section 6.3).

Let \( A = (WQ)^{\% λ} + ξ + rUS \% M \).

Due to d | M, if (WQ)^{\% λ} + ξ + rUS contains the factor d, it must be contained in A \% M.

Besides, due to d | A and d | (WQ)^{\% λ} (according to d | (WQ)), if we want to make d | A, there must be d | (rUS + ξ) \% M.

Therefore, as long as every value of r makes rUS different, d | A will holds after about d attempts of r. The algorithm can also terminate normally because after r traverses the interval [1, d^{216}], the probability of d | A is (1 - 1/d\% d^{216}), and almost zero.

At S4, we derive \( ξ \) from \( ξ(δOQ = WH) = (δQ)^{\% M} \).

Computing R by \( Q = (RQ)\% δ (δQ \% M) \) may resort to Theorem 1, where S meets gcd(S, M) = 1.

**Theorem 1:** For the congruence \( x^k = c (\% M) \) with M prime, if gcd(k, M) = 1, every c has just one k-th root modulo M. Especially, let μ satisfy \( μk = 1 (\% M) \), then e^\% M is the k-th root.

Further, we have Theorem 2 and 3.

**Theorem 2:** For the congruence \( x^k = c (\% M) \) with M prime, if k ∣ M and gcd(k, M/k) = 1, then when c is an k-th
power residue modulo $M$, and $\mu$ satisfies $\mu k = 1 \ (\text{mod} \ M)$, $c^2 \equiv M$ is an $k$-th root.

**Theorem 3:** For the congruence $x^t = c \ (\text{mod} \ M)$ with $M$ prime, if $k \mid M$, let $h = \gcd(k,M)$, $m = k/h$, and $\mu$ satisfy $\mu m = 1 \ (\text{mod} \ h)$, then $x^t = c \ (\text{mod} \ M)$ is equivalent to $x^h = c^\mu \ (\text{mod} \ M)$,

that is, the two congruences have the same set of solutions. Furthermore, the sufficient and necessary condition for either congruence to have solutions is $c^{\phi(h)} \equiv 1 \ (\text{mod} \ M)$.

For the proofs of Theorem 1 and 2, refer to [26], and for the proof of Theorem 3, refer to [27]. The solution which is obtained in terms of Theorem 1 or 2, may be written as a certain power of $c$ modulo $M$ is called the trivial solution to the congruence $x^t = c \ (\text{mod} \ M)$ [27].

### 3.5 The Identity Verification Algorithm

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

**S1:** Let $H = \text{hash}(F)$, whose binary form is $b_1 \ldots b_n$.

**S2:** Compute $\hat{\sigma} \leftarrow \prod_{i=1}^n C_i \ (\text{mod} \ M)$.

**S3:** Compute $X \leftarrow (\hat{\sigma}Q^U)^{-1}M \ (\text{mod} \ M)$, $Y \leftarrow (\hat{\sigma}Q^U)^{-1}H \ (\text{mod} \ M)$.

**S4:** If $X = Y$, the identity is valid and $F$ intact; else the identity is invalid or $F$ modified.

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

**Definition 7:** Let $M$ be a prime. Seeking $x \in \{1, M\}$ from $y = x^t \ (\text{mod} \ M)$ is called the transcendental logarithm problem, shortly TLP.

In what follows, we argue the discriminant $X = Y \ (\text{mod} \ M)$ at $S4$.

It is known from Section 3.1 that

$$\begin{align*}
\alpha &= \delta^{(\alpha + W_1 \cdot \cdot \cdot - t)} \delta h(WG_1G_2) \\
\beta &= \delta^{W_1} \ (\text{mod} \ M).
\end{align*}$$

Let $V = (R^T W_1 G_2) \delta^\lambda \ (\text{mod} \ M)$.

Since $\lambda$ meets $\lambda S = ((WQ)^{1} + \frac{\delta}{\text{deg}} + rUS)(\delta Q - HW)(\text{mod} \ M)$, let $\lambda = k \alpha D$, where $k$ is a $k$-th root, and then

$$\begin{align*}
Q^U V &= (R G_2 G_3) \delta^\lambda \delta \delta S \\
&= (W G_1 G_2) \delta \delta S \\
&=\alpha \delta^{W_1 \text{deg} + W_1 + \cdot \cdot \cdot + t} \delta^{W_1 \text{deg} + W_1 + \cdot \cdot \cdot + t}
\end{align*}$$

Transposition yields

$$V^t = (\alpha Q^U) \delta^{W_1 \text{deg} + W_1 + \cdot \cdot \cdot + t} \delta^{W_1 \text{deg} + W_1 + \cdot \cdot \cdot + t} \delta^{W_1 \text{deg} + W_1 + \cdot \cdot \cdot + t} \delta^{W_1 \text{deg} + W_1 + \cdot \cdot \cdot + t} \ (\text{mod} \ M).$$

Hence

$$V^t = \left(\alpha \delta^{W_1 \text{deg} + W_1 + \cdot \cdot \cdot + t} \delta^{W_1 \text{deg} + W_1 + \cdot \cdot \cdot + t} \delta^{W_1 \text{deg} + W_1 + \cdot \cdot \cdot + t} \delta^{W_1 \text{deg} + W_1 + \cdot \cdot \cdot + t} \ (\text{mod} \ M).$$

**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

$$\begin{align*}
x^t &= a \ (\text{mod} \ M) \\
x^s &= b \ (\text{mod} \ M)
\end{align*}$$

have the unique solution if and only if \(a' = b' \ (\text{mod} \ M)\).

**Proof:**

Necessity:

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

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

Further, $x_0^s = a' \ (\text{mod} \ M)$ and $x_0^t = b' \ (\text{mod} \ M)$ can be obtained.

Therefore, $x_0^{st} = a'b' \ (\text{mod} \ M)$.

Sufficiency:

Assume that $a' = b' \ (\text{mod} \ M)$.

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

$$\begin{align*}
x^u &= a' \ (\text{mod} \ M) \\
x^v &= b' \ (\text{mod} \ M)
\end{align*}$$

The above two equations multiplying yields $x^{us+vt} = x^t = a' b' \ (\text{mod} \ M)$.

Furthermore, we have

$$\begin{align*}
(a'b')^t &= a'^tb'^t = a''a'^{t+v} = a^{t+v} \ (\text{mod} \ M) \\
(a'b')^s &= a'^sb'^s = b''b'^{s+v} = b^{s+v} \ (\text{mod} \ M)
\end{align*}$$

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

Uniqueness:

Let $x_0 = a' b' \ (\text{mod} \ M)$.

Assume that another value $x_1$ meets the equations $x'^t = a \ (\text{mod} \ M)$ and $x'^s = b \ (\text{mod} \ M)$ at one time.

Then, it holds that

$$x_1^t = a \ (\text{mod} \ M)$$

By comparison, we have $x_1^t = x_0^t$ and $x_1^s = x_0^s$.

Transposing gives

$$(x_0 x_1^s)^t = 1 \ (\text{mod} \ M)$$

If at least one between $s$ and $t$ is relatively prime to $\mathcal{M}$, by Theorem 1, there must be $x_0 x_1^{-1} = 1 \ (\text{mod} \ M)$, namely $x_0 = x_1 \ (\text{mod} \ M)$.

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

Thus, there are $(x_0 x_1^{-1})^t = 1$ and $(x_0 x_1^{-1})^s = 1 \ (\text{mod} \ M)$.

By Theorem 3 and $\gcd(s, t) = 1$, we know $\gcd(k, h) = 1$. In terms of the group theory [22], when $\gcd(k, h) = 1$, only the element ‘1’ belongs to two different subgroup at the same time. Therefore, $x_0 x_1^{-1} = 1$, namely $x_1 = x_0$, and $x_0$ bears uniqueness.

To sum up, we prove Theorem 4. \qed
3.7 Characteristics of REESSE1+

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

- The security of REESSE1+ is not based on a single problem, but on the four problems: MPP, ASPP, TLP, and PRFP. Hence, it is a multiproblem public key cryptosystem.
- The key transform $Ci = (A,W^{i(t)})^δ(M)$ for $i = 1, \ldots, n$ contains $2n+2$ unknown variables, and each equation contains four almost independent variables. Hence, REESSE1+ is multivariate.
- If any of $A_i$, $W_i$ and $(t(i))$ is determined, the relation between the two remainders is still nonlinear, and thus there is a very complicated nonlinear relations among $A_i$, $W_i$, and $(t(i))$.
- The indeterminacy of $t(.)$ with $δ = 1$. If $C_i$ and $W_i$ are determined, $A_i$ and $t(i)$ cannot be determined, and even have no one-to-one relation when $W_i$ is a non-generator. If $C_i$ and $A_i$ are determined, $W_i$ and $t(i)$ cannot be determined, and also have no one-to-one relation for $\gcd(t(i),M) > 1$.
- The insufficiency of the mapping. A private key includes $\{A_i\}$, $\{t(i)\}$, $W_i$, $δ$ etc. but there is only a dominant mapping from $\{A_i\}$ to $\{C_i\}$, and thus the invertibility of the transform function is poor.
- Because combinations among multiple variables may bring different hardnesses, REESSE1+ is a self-improvable cryptosystem while its main architecture remains unchanged.

3.8 Correctness of the Decryption Algorithm

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

$W_k(W^{-1})^k = W_k W^k = 1 (\% M)$

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

It is known from Section 3.2 that $G = \prod_{i=1}^{n} C_i^h (\% M)$, where $b_i$ means what the algorithm shows, and $C_i = (A_i,W_i^{t(i)})^h (\% M)$.

Let $G = \prod_{i=1}^{n} A_i^h (\% M)$, and $k = \sum_{i=1}^{n} t(i) b_i$.

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

Proof:

According to the key generator and the encryption algorithm, there is

$G = \prod_{i=1}^{n} C_i^h = \prod_{i=1}^{n} ((A_i,W_i^{t(i)})^h)^h$

$= W^\sum_{i=1}^{n} t(i) b_i (A_i,W_i^{t(i)})^h$

$= W^\sum_{i=1}^{n} t(i) b_i A_i^h (\% M)$

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

$G^{\delta^{-1}} = (W^\delta) (\prod_{i=1}^{n} A_i^h)^{\delta^{-1}}$

$= W^\delta \prod_{i=1}^{n} A_i^h (\% M)$

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

$G^{\delta^{-1}} = (W^\delta \prod_{i=1}^{n} A_i^h)(W^{-1})^k$

$= W^\delta \prod_{i=1}^{n} A_i^h (W^{-1})^k$

$= \prod_{i=1}^{n} A_i^h G (\% M)$

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 $k$. However, because the range of $k \in (5n, (2n + 3))$ is very narrow, we may search $k$ heuristically by multiplying $W^\delta$, and verify whether $G = 1$ after it is divided exactly by some $A_i^h$. 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 $\prod_{i=1}^{n} C_i^h (\% M)$ to $G$ (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 $G$ is being unveiled.

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

$G = \prod_{i=1}^{n} C_i^h = \prod_{i=1}^{n} C_i^h (\% M)$

That is,

$\prod_{i=1}^{n} (A_i,W_i^{t(i)})^h = \prod_{i=1}^{n} (A_i,W_i^{t(i)})^h (\% M)$

Further, there is

$W^\delta \prod_{i=1}^{n} (A_i)^h = W^\delta \prod_{i=1}^{n} (A_i)^h (\% M)$

where $k = \sum_{i=1}^{n} t(i) b_i (\% M)$,

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

$W^\delta \prod_{i=1}^{n} A_i^h = W^\delta \prod_{i=1}^{n} A_i^h (\% M)$

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

$W^\delta k = \prod_{i=1}^{n} A_i^h (\prod_{i=1}^{n} A_i^h)^{-1} (\% M)$

Let $\theta = \prod_{i=1}^{n} A_i^h (\prod_{i=1}^{n} A_i^h)^{-1} (\% M)$, namely $\theta = W^\delta k (\% M)$.

The above congruence signifies that when the plaintext $b_1 \ldots b_n$ is not unique, the value of $W$ must be relevant to $\theta$. The contrapositive assertion equivalent to it is that if the value of $W$ is irrelevant to $\theta$, $b_1 \ldots b_n$ will be unique. Thus, we need to consider the probability that $W$ takes a value relevant to $\theta$.

If an adversary tries to attack an 80-bit symmetric key through 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 $\prod_{i=1}^{n} A_i^h$ has $2^n$ potential values, and thus the number of potential values of $\theta$ is at most $2^2 \times 2^n$. Notice that because $A_{i+1}, \ldots, A_{i+1}$ are not necessarily coprime, some values of $\theta$ may possibly occur repeatedly.

Because $|k - k| \leq m(2n + 3) - 5n \leq 32512 \approx 2^{15}$ with $n \leq 128$, and $W$ has at most $2^{25}$ solutions to every $\theta$, the probability that $W$ takes a value relevant to $\theta$ is at most $2^{12} 2^n/M$. When $n \geq 80$, there is

$2^{15} 2^{20} / M < 2^{12} 2^n / 2^{20} = 1 / 2^{21}$

which is close to zero. The probability will further decrease when $W$ is a prime since the solutions to $\theta$ lean toward being composite integers averagely.
In addition, if please, resorting to $\sum_{i=1}^{n} b_i = n$, you may exclude some unoriginal plaintext solutions.

4 Security Analysis of the Key Transform

We analyze the exact security of the REESSE1+ key transform $C_i = (A, W^{\ell_0_i})^i$ (% M) for $i = 1, \ldots, n$, where $W$, $\delta \in [1, \mathcal{M}]$, $\ell(i) \in \mathcal{O} = \{5, 7, \ldots, 2n + 3\}$, and $A_i \in A = \{2, 3, \ldots, 1201\}$.

We know that when $n = 80, 96, 112, \text{ or } 128$, there is $\lceil \lg M \rceil \approx 696, 864, 1030, \text{ or } 1216$. In this case, IFP and DLP can almost be solved in tolerable time, and LSVP with $D \approx n / \lceil \lg M \rceil$ can also be solved in tolerable time [11][28]. In addition, because the root finding problem (RFP) $y = x^\delta$ (% M) may be converted into a linear congruence through a discrete logarithm, RFP can also be solved in tolerable time when DLP can be solved in tolerable time.

‘Tolerable time’ indicates that the running time of an algorithm for solving a problem may be accepted by a user when the time dominant parameter is relatively small. For example, when $n = 80$, the time $O(2^{60})$ is tolerable, and when $\lceil \lg M \rceil = 384, O(L_M[1/3, 1, 923]) = 2^{56}$ is also tolerable [29].

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

**Definition 8**: Let $A$ and $B$ be two computational problems. $A$ is said to reduce to $B$, written as $A \leq^r_T B$, 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 $B$ [25][28].

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.

$A \leq^r_T B$ 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 $A$ cannot be solved in polynomial or subexponential time, $B$ cannot 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 9**: Let $A$ and $B$ be two computational problems. If $A \leq^r_T B$ and $B \leq^r_T A$, then $A$ and $B$ are said to be computationally equivalent, written as $A =^r_T B$ [25][28].

$A =^r_T B$ means that either if $A$ is a hardness of a certain complexity on condition that the dominant variable approaches a large number, $B$ is also a hardness of the same complexity on the identical condition; or $A, B$ both can be solved in linear or polynomial time.

Definition 8 and 9 suggest a reductive proof method called polynomial time Turing reduction (PTR) [25]. Provably secure 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 enquire whether $A <^r_T B$ exists or not. The definition of $A <^r_T B$ may possibly be given theoretically, but the proof of $A <^r_T B$ is not easy in practice.

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

4.1 MPP Is at Least Equivalent to DLP

Definition 4 refers to $C_i = (A, W^{\ell_0_i})^i$ (% M) for $i = 1, \ldots, n$ as MPP. It has the following property.

**Property 4**: MPP is computationally at least equivalent to DLP in the same prime field.

**Proof**: Firstly, systematically consider $C_i = (A, W^{\ell_0_i})^i$ (% M) for $i = 1, \ldots, n$.

Assume that $g_i = A_iW^{\ell_0_i}$ (% M) for each $i$ is a constant.

Let $g_i = g^\delta_i$ (% M), and $z_i = \delta x_i$ (% $\mathcal{M}$), where $g \in \mathbb{Z}_M^*$ be a generator.

Then, there is $C_i = g_i^\delta_i$ (% M) for $i = 1, \ldots, n$.

Again let $\delta x_i = z_i$ (% $\mathcal{M}$). Then $C_i = g_i^\delta_i$ (% M) for $i = 1, \ldots, n$.

The above expression corresponds to the fact that in the ElGamal cryptosystem with many users sharing a modulus and a generator, user 1 acquires the private key $z_1$ and the public key $C_1$, ..., user $n$ acquires the private key $z_n$ and the public key $C_n$. It is well known that in this case, attack of adversaries is still faced with DLP, namely seeking $z_i$ from $C_i = g_i^\delta_i$ (% M) for $i = 1, \ldots, n$ is equivalent to DLP [25].

Thus, when every $g_i$ is weakened to a constant, seeking $\delta$ from $C_i = g_i^\delta_i$ (% M) for $i = 1, \ldots, n$ is equivalent to DLP, which indicates that when every $g_i$ is not a constant, seeking $g_i$ and $\delta$ from $C_i = g_i^\delta_i$ (% M) for $i = 1, \ldots, n$ is at least equivalent to DLP.

Secondly, singly consider a certain $C_i$, where the subscript $i$ is designated.

Assume that $\tilde{O}_a(C_i, M, B)$ is an oracle on solving $C_i = g_i^\delta_i$ (% M) for $g_i$ and $\delta$, where $i$ is in $\{1, \ldots, n\}$, and $B$ is a constraint on $g_i$ such that the original $g_i$ can be found.

Let $y = g^\delta_i$ (% M) be of DLP. Then, by calling $\tilde{O}_a(y, M, g)$, $x$ can be obtained.

According to Definition 8, there is $H(y = g^\delta_i$ (% M) $\leq^r_H H(C_i = g_i^\delta_i$ (% M)), which means that when only a certain $g_i$ is known, seeking $g_i$ and $\delta$ from $C_i = g_i^\delta_i$ (% M) is at least equivalent to DLP.

Integrally, seeking the original $\{A_i\}$, $\{\ell(i)\}$, $W$, and $\delta$ from $C_i = (A_iW^{\ell_0})^\delta_i$ (% M) for $i = 1, \ldots, n$ is computationally at least equivalent to DLP in the same prime field.

Further, the following analysis will incline people to believe that MPP is harder than DLP.

4.2 Attacks by Interaction of the Key Transform Items

Every $\ell(i) \in \{5, 7, \ldots, 2n + 3\}$ and every $A_i \in \{2, 3, \ldots, 1201\}$ are the thinness of $C_i = (A_iW^{\ell_0})^\delta_i$ (% M). Naturally
adversaries will adopt combinational attack measures around the thinness.

4.2.1 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 $G_i = C_i(x_1, x_2, y_1, y_2)$, namely

$$G_i = (A_i, A_i, A_i, A_i, A_i, A_i, A_i) \mod (\% M),$$

If the adversaries divide the values of $A_i, A_i, A_i, A_i$, and $A_i$, and compute $u = v_i \mod V_j, v_i, v_j, v_i$ in time of at least $L_0[1/3, 1.923]$ such that

$$g_i = g^{\alpha_i}, A_i = g^{\beta_i}, y_i = g^{\gamma_i}, y_i = g^{\gamma_i}, (% M),$$

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

$$u = (v_i + v_i - v_j - v_j) \mod (\% M).$$

If gcd$(v_i, v_i - v_j - v_j, \% M) = u$, the congruence in $\delta$ has solutions. Because each of $A_i, A_i, A_i, A_i, A_i$ may traverse the interval $A_i$, the subscripts $x_1, x_2, y_1, y_2$ are unfixed, and the congruence may have $n$ solutions, the number of potential values of $\delta$ is about $n^2[A_i]$. In succession, the most effectual approach seeking $W$ is that for every $i$, divide $A_i$ and $(\% i)$, find $V_i$ by $C_i = (A_i, W(i)^{\delta}) \mod (\% M)$, namely the value set of $W$, and if there exists $W_1 \in V_1, \ldots, W_n \in V_n$, being equal pairwise, the divination of $\delta, \{A_i\}, \{(\% i)\}$ is thought right. Notice that to avoid seeking $(\% i)$-th roots, may let $W = g^{\delta} \% M$.

Due to $\prod_{i=1}^n p_i^{\epsilon_i} \mod M$, where $k$ meets $p_k = 2n$, there is $(\% i) \mod M$, and the size of every $V_i$ is about $n[A_i]$. In summary, the running time of the above attack is at least

$$F = n[A_i]L_0[1/3, 1.923] + (2n)A_i[\% A_i]2[\log_2 M] + 2n[\% A_i] = n[A_i]L_0[1/3, 1.923] + 2^{10} > 2^n.$$

When $n = 80$ with $[\log_2 M] \approx 696$, $F = 2^{110} > 2^n$.

When $n = 96$ with $[\log_2 M] \approx 864$, $F = 2^{115} > 2^n$.

When $n = 112$ with $[\log_2 M] \approx 1030$, $F = 2^{125} > 2^n$.

When $n = 128$ with $[\log_2 M] \approx 1216$, $F = 2^{129} > 2^n$.

Therefore, $F$ is not less than a quantity of time exponential in $n$.

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

4.2.2 Eliminating $W$ through the $[\% M]$-th Power

Due to $[\log_2 M] \approx 696, 864, 1030, or 1216$, $M$ can be factored in tolerable time. Again due to $\prod_{i=1}^n p_i^{\epsilon_i} \mod M$ and $\prod_{i=1}^n V_i \mod M$ where $k$ meets $p_k = 2n$, $[\% M]$ can be divided in the running time of about $2^n$.

Raising either side of $C_i = (A_i, W(i)^{\delta}) \% M$ to the $[\% M]$-th power yields

$$C_i^{[\% M]} = (A_i)^{[\% M]} \% M.$$

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

$$u_i[\% M] = v_i[\% M] \mod (\% M)$$

for $i = 1, \ldots, n$. Notice that $v_i \neq v_i \% M$.

The above congruence looks to be the MH transform [8]. Actually, $[\% v_1, ..., v_n] \% M$ is not a super increasing sequence, and moreover there is not necessarily $[\log(u_i[\% M])] = [\log_2 M]$.

Because $v_i[\% M] \in \{1, M\}$ is stochastic, the inverse $\delta^{-1} \% M$ not need be close to the minimum

$$[\% M/(u_i[\% M]), 2[\% M/(u_i[\% M]), \ldots, (u_i[\% M] - 1)M/(u_i[\% M])]$$

Namely $\delta^{-1}$ may lie at any integral position of the interval

$$k\% M/(u_i[\% M]), (k + 1)\% M/(u_i[\% M]),$$

where $k = 0, 1, \ldots, u_i[\% M] - 1$, which illustrates the accumulation points 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 $\delta^{-1}$ is likely $\max_{2 \leq i \leq n} u_i[\% M]$ which is promisingly close to $M$. Therefore, the Shamir attack by the accumulation point of minima is fully ineffectual [9].

Even if find out $\delta^{-1}$ by the Shamir attack method, because each of $v_i$ has $[\% M]$ solutions, the number of potential sequences $\{g^{v_1}, \ldots, g^{v_n}\}$ is up to $[\% M]^n$. Because of 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 $[\% M]^n$. Hence, the initial $\{A_i, \ldots, A_i\}$ cannot be determined in polynomial time. Further, the value of $V$ cannot be computed, and the values of $[\% M]$ and $\delta^{-1}$ cannot be verified in polynomial time, which indicates that MPP can also be resistant to the attack by the accumulation point of minima.

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

$$F = n[A_i]L_0[1/3, 1.923] + 2^{10}[\% M][\% M] = n[A_i]L_0[1/3, 1.923] + 2^{10}[\% M][\% M] \approx n[A_i]L_0[1/3, 1.923] + 2^n > 2^n.$$

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

Again, it is infeasible to separate $h, \delta, W$, and $\prod_{i=1}^n A_i$ distinctly from $\alpha = h^0(W \prod_{i=1}^n A_i)^{-\delta} \% M$, and the time complexity of seeking $W$ from $\alpha = h^0(W \prod_{i=1}^n A_i)^{-\delta} \% M$ and $\beta = \delta^{-1} \% M$ will be at least $O(2^n)$ (see Section 6.3.3), so even if the three equations are considered simultaneously, it is also impossible to determine the values of the four variables almost independent.

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

4.3 Attack by a Certain Single $C_i$

Assume that there is only a solitary $C_i = (A_i, W(i)^{\delta}) \% M - i = 1$ for example, and other $C_i$'s $(i = 2, \ldots, n)$ are unknown for attackers.

Through divining $A_i \in \Lambda$ and $(\% i) \in \Omega$, the parameters $W$ and $\delta \in (1, \% M)$ can be computed. Thus, the number of solution $(A_i, (\% i), W, \delta)$ will be up to $[\log_2 M][\% M] > 2^n$, which manifests that the original $(A_i, (\% i), W, \delta)$ cannot be determined in time being subexponential in $n$ [30].

Evidently, if $g_i = A_i W(i)^{\delta} \% M$ is a constant, solving $C_i = g_i^{v_i} \% M$ for $i$ is equivalent to DLP. Factually, $g_i$ is not a constant. At present, seeking the original $g_i, \delta$ will take at least $O(M) > O(2^n)$ steps.
5 Security Analysis of the Encryption Algorithm

The security of the encryption algorithm is namely the security of a REESSE1+ ciphertext.

5.1 ASPP is at Least Equivalent to DLP

Definition 5 refers to $\tilde{G} = \prod_{i=1}^{n} C_{i}^b$ (% M) as SPP. It has the following property.

Property 5: SPP is computationally at least equivalent to DLP in the same prime field.

Proof: Let $\tilde{G}_1 = \prod_{i=1}^{n} C_{i}^b$ (M) where $b_1 \ldots b_n$ is a plaintext block.

Especially, define $\tilde{G}_1 = \prod_{i=1}^{n} C_{i}^{2^{r-1}} = \prod_{i=1}^{n} C_{i}^{(2^{\nu})^b}$ (M) where $C_1 = \ldots = C_n = C$. Obviously, $\prod_{i=1}^{n} C_{i}^b = L \cdot M + \tilde{G}_1$. Owing to $L \in [1, M]$, deriving the non-modular product $\prod_{i=1}^{n} C_{i}^b$ from $\tilde{G}_1$ is infeasible, which means inferring $b_1 \ldots b_n$ from $\tilde{G}_1$ is not a factorization problem.

Assume that $\tilde{G}_1$ is an oracle on solving $\tilde{G}_1 = \prod_{i=1}^{n} C_{i}^b$ (% M) for $b_1 \ldots b_n$.

Let $y = g^z$ (M) be of DLP, where $g$ is a generator of $(\mathbb{Z}_m^*, \cdot)$, and the binary form of $y$ is $\tilde{b}_1 \ldots \tilde{b}_n$, namely $y = \prod_{i=1}^{n} (g^{2^{r-1}})^{\tilde{b}_i}$. Then, by calling $\tilde{G}_1(y, g^{-1}, \ldots, g, M)$, namely $b_1 \ldots b_n$ can be found.

By Definition 8, there is $H(y, g^{2^{r-1}}) \leq \tilde{G}_1$. Property 8 refers to $\tilde{G}_1 = \prod_{i=1}^{n} C_{i}^b$ (% M) as SPP. It has a similar property.

Property 6: ASPP is computationally at least equivalent to DLP in the same prime field.

Proof: Assume that $\tilde{G}_1$ is an oracle on solving $\tilde{G} = \prod_{i=1}^{n} C_{i}^b$ (% M) for $b_1 \ldots b_n$, where $b_1 \ldots b_n$ is the bit shadow string of $b_1 \ldots b_n$.

Especially, define $\tilde{G} = \prod_{i=1}^{n} C_{i}^{n-1} = \prod_{i=1}^{n} (C_{i}^{2^{r-1}})^b$ (% M) with the stipulation $b_i < n$ (namely that $b_1 \ldots b_n$ contains at least two nonzero bits) when $C_1 = \ldots = C_n = C$.

Let $\tilde{G}_2 = \prod_{i=1}^{n} C_{i}^b$ (% M) be of SPP.

Due to $0 \leq b_1 \leq b_i$, by calling $\tilde{G}_2(\tilde{G}_1, C_1, \ldots, C_n, M)$, namely $b_1 \ldots b_n$ can be found.

By Definition 8, there is $H(\tilde{G}_1; \tilde{G} = \prod_{i=1}^{n} C_{i}^b$ (% M)) s.t. $H(\tilde{G} = \prod_{i=1}^{n} C_{i}^b$ (% M)).

Further by transitivity, there is $H(y, g^{2^{r-1}}) \leq \tilde{G}_2$. Property 9 refers to $\tilde{G} = \prod_{i=1}^{n} C_{i}^b$ (% M) as SPP. It has a similar property.

It is very interesting whether ASPP is harder than DLP or not. We can find some positive evidence.

Piece 1 of evidence: Have $C_1 = \ldots = C_n = C$, then we have

$$\tilde{G} = \prod_{i=1}^{n} C_{i}^{n-1}$$

which may be written as

$$\tilde{G} = C_{i}^{n-1}$$

which is a form of DLP.

Correspondingly,

$$\tilde{G} = C_{i}^{n}$$

which is a form of DLP.

However, when $C_1, \ldots, C_n$ are generated, we can check $C_1, \ldots, C_n$ to prevent any two elements in $\{C_1, \ldots, C_n\}$ from being equal. Therefore, in practice, ASPP cannot be reduced to DLP in any time.

Piece 2 of evidence: Assume that DLP can be solved in tolerable subexponential time.

When DLP can be solved in tolerable time, SPP can also be factorized [14][25], so a generator can be found through the algorithm 4.80 in Section 4.6 of [25].

Let $C_1 = g^v$ (% M), $\ldots, C_n = g^v$ (% M), $\tilde{G} = g^v$ (% M), where $g$ is a generator of $(\mathbb{Z}_m^*, \cdot)$.

Then, solving $\tilde{G} = \prod_{i=1}^{n} C_{i}^b$ (% M) for $b_1 \ldots b_n$ is equivalent to solving

$$b_1 u_1 + \ldots + b_n u_n = v \ (% M)$$

which is called the anomalous subset sum problem (ASSP) due to $b_1 \geq b_n$ and computationally at least equivalent to SPP.

It has been proven that SPP is NP-complete in its feasibility recognition form, and the computational version of SPP with the sufficiently large length and density is NP-hard [4][25]. Hence, solving ASSP is at least NP-hard. Additionally, the density relevant to ASSP is far greater than 1 because of $b_1 \geq b_1$ (see Appendix). This indicates that the $L^2$ lattice base attack on ASSP will be ineffectual (see Section 5.2). These two points illustrate that even although DLP can be solved, $b_1 \ldots b_n$ cannot be found yet in polynomial time.

The above two pieces of evidence incline us to believe that ASSP is harder than DLP.

By the way, LSSP will degenerate to a polynomial time problem from NPC [11][31].

5.2 ASPP Can Resist $L^3$ Lattice Base Reduction

It is known from Section 3.2 that the ciphertext $\tilde{G} = \prod_{i=1}^{n} C_{i}^b$ (% M).

Still let $C_1 = g^v$ (% M), $\ldots, C_n = g^v$ (% M), $\tilde{G} = g^v$ (% M), where $g$ is a generator of $(\mathbb{Z}_m^*, \cdot)$ randomly selected.

Then, seeking $b_1 \ldots b_n$ from $\tilde{G}$ is equivalent to solving the congruence

$$u_1 b_1 + \ldots + u_n b_n = v \ (% M)$$

where $v$ may be substituted with $v + k \mathcal{M}$ with $k \in [0, n-1]$ [32]. $u_1, \ldots, u_n$ is called a compact sequence due to $b_1 \in [0, n]$ and $n > 1$.

Recall [10] and [11]. Let $\{a_1, \ldots, a_m\}$ be a positive integer sequence, $\hat{e} = \{\hat{e}_1, \ldots, \hat{e}_0\}$ with $\hat{e}_i \in [0, 1]$ be the solution vector, $s = \sum_{i=1}^{n} a_i \hat{e}_i$ and $t = \sum_{i=1}^{n} a_i$.

In [10], there are two important conditions:

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

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

\[
\frac{t}{n} \leq s \leq \frac{n(n-1)}{2}, \quad \text{and} \quad |\mathcal{G}|^2 \leq n/4,
\]

which decide the threshold density \( < 0.9408 \).

However, for (1), due to \( 0 \leq b_i \leq n \), the similar conditions do not hold.

It is well understood that the \( L^3 \) lattice base reduction algorithm is employed in crytpanalysis to find the shortest vector or approximately shortest vectors in a lattice, and hence, if a solution to SSP has a comparatively big distance, or is not unique, it will not occur in the reduced base. Notice that because \( \mathcal{L} \) is random.

Let \( \mathcal{L} \) be a lattice spanned by the vectors

\[
(1, 0, \ldots, 0, Nu_1),
(0, 1, \ldots, 0, Nu_2),
\vdots
(0, 0, \ldots, 0, \mathcal{N}N_n),
(0, 0, \ldots, 0, N(v+k \mathcal{N})),
\]

which compose a basis 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 \( \mathcal{L} \), is also random.

Let \( \mathcal{Q} \) be the determinant relevant to a matrix corresponding to the lattice base. Then, by the Guassian heuristic, the expected size of the shortest vector in \( \mathcal{L} \) of \( n + 1 \) dimensions lies between [15]

\[
\mathcal{Q}^{(n+1)/(2n\pi)}(n+1)/(2\pi)^{1/2} \quad \text{and} \quad \mathcal{Q}^{(n+1)/(n+1)}(n+1)/(2\pi)^{1/2},
\]

where \( \varepsilon = 2.7182818 \).

In our case, there is \( |\mathcal{L}|^2 \sim (n+1) \approx 9.19 > 2 > 1. \)

Thus, it is very likely that the original solution vector will not occur in the reduced base.

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

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

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

\[
D = n^2/|\mathcal{L}|^2 \quad \text{(see Appendix B)}.
\]

When \( n = 80 \) with \( |\mathcal{L}|^2 = 696, D \approx 9.19 > 2 > 1. \)

When \( n = 96 \) with \( |\mathcal{L}|^2 = 864, D \approx 10.66 > 2 > 1. \)

When \( n = 112 \) with \( |\mathcal{L}|^2 = 1030, D \approx 12.18 > 2 > 1. \)

When \( n = 128 \) with \( |\mathcal{L}|^2 = 1216, D \approx 13.47 > 2 > 1. \)

\( D > 2 \) indicates that a great many different subsets will have the identical sum, namely the solution to (1) is not unique, and the original solution is possibly not shortest for \( b_i \in [0, n] \). Thus, it is very likely that the original solution does not occur in the reduced base only containing \( n + 1 \) 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 \( \lg M \geq 696 \) cannot 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 \( (b_1, \ldots, b_n, 0) \), which manifests that the only classical \( L^3 \) algorithm is appropriate.

According to [25], the running time of attack on equation (1) from ASP by the lattice base reduction algorithm is roughly

\[
T \approx O(n|\mathcal{L}|^2(1/3, 1.923) + n(n+1)^2(|\mathcal{L}|^2)^3)
\]

on condition that \( N \) is slightly greater than \( n \).

When \( n = 80 \) with \( |\mathcal{L}|^2 = 696, T \approx 2^{16}. \)

When \( n = 96 \) with \( |\mathcal{L}|^2 = 864, T \approx 2^{20}. \)

When \( n = 112 \) with \( |\mathcal{L}|^2 = 1030, T \approx 2^{32}. \)

When \( n = 128 \) with \( |\mathcal{L}|^2 = 1216, T \approx 2^{36}. \)

However, as is pointed out in the above, owing to \( |D| > 2 > 1 \) and \( b_i \in [0, n] \), it is almost impossible that the solution vector \((b_1, \ldots, b_n, 0)\) occurs in the final reduced base, which means that attack by the \( L^3 \) 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 security of a REESSE1+ plaintext is built on the problem \( \mathcal{G} = \prod_{i=1}^s C_i^b \% M \) which contains the trapdoor information, and means that computing an anomalous subset product from subset elements is tractable while seeking the subset elements from the product is intractable.

5.3 Avoid Adaptive-chosen-ciphertext Attack

Theoretically, most of public key cryptosystems 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 the RSA encryption [35]. Almost at the same time, The Cramer-Shoup asymmetric encryption algorithm which is extremely malleable, and an extension of the Elgamal algorithm was proposed [36]. It is the first efficient scheme proven to be secure against an 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.

5.3.1 Different Ciphertexts of the Identical Plaintext

It is lucky that REESSE1+ can avoid the adaptive-chosen-ciphertext attack when a REESSE1+ ciphertext is produced according to the following algorithm:

Paralleling Section 3.2, 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 \( k \leftarrow k + 1, b_i \leftarrow 0; \) else let \( b_i \leftarrow k + 1, k \leftarrow 0 \).

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

S4: Randomly produce \( s_1 \ldots s_n \in [0, 1]^n \).

S5: If \( b_a = 0 \), set \( d \leftarrow n - k, s_d = 1, b_d \leftarrow b_d + k; \) else \( s_a = 1 \).
S6: Compute $\vec{c} \leftarrow \prod_{i=1}^{n} (C_{i} G^{(i-1) \cdot \vec{e} \cdot \vec{h}}) \% M$ with $C_n = 1$. Clearly, when the identical plaintext is inputted many times, the above algorithm will return a ciphertext different from one another every time. It is easily understood that contrarily, a ciphertext can almost uniquely be decrypted in polynomial time in terms of Section 3.3 and 3.9. Paralleling Section 3.3, we design a related decryption algorithm of which the running time is equivalent to that of the algorithm in Section 3.3.

Assume that $(\{A_i\}, \{\ell(i)\}, W, \delta, D, d, h)$ is a related private key, and $\mathcal{G}$ is a ciphertext. Notice that because $\sum_{i=1}^{n} b_i \ell(i)$ must be even.

S1: Compute $\vec{c} \leftarrow \mathcal{G}^{\vec{e} \cdot \vec{h}} \% M$.
S2: Compute $\mathcal{G} \leftarrow \mathcal{G} W^{-2} \% M$.
S3: Set $b_1 \ldots b_{n} \ell(i), k \leftarrow 0, G \leftarrow \mathcal{G}, i \leftarrow 1$.
S4: If $A_i^{\vec{e}+1} | G$, let $\vec{e} \leftarrow \vec{e} + 1$, goto S4.
S5: If $\vec{e} \equiv 0$

1. Let $k \leftarrow k + 1, i \leftarrow i + 1$; else compute $G \leftarrow G / A_i^{\vec{e}}$;
2. If $k > 0$ or $i + \vec{e} \equiv 1 \% n$, let $b_i \leftarrow 1$; else $b_{i+\vec{e} \cdot \vec{h}} \leftarrow 1$;
3. If $k = 0$, let $i \leftarrow i + \vec{e}$; else $i \leftarrow i + 1$;
4. If $k + 1 \equiv \vec{e} \% n$, let $i \leftarrow i + 1$;
5. Set $\vec{e}, k \leftarrow 0$.
S6: If $i \leq n$ and $G \equiv 1$, goto S4.
S7: If $G \equiv 1$, goto S2; else end.

In this way, the original plaintext block $b_1 \ldots b_n$ is recovered although $s_1 \ldots s_n$ is introduced in encryption. Besides, in decryption, $\ell(i)$, $D$, $d$, and $h$ are unhelpful.

5.3.2 Appending of a Stochastic Binary String

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

6 Security Analysis of the Digital Signature

The security of the REESSE1+ signature includes three aspects: a private key cannot be inferred from a signature, a signature cannot be forged through known signatures, public keys, and algorithms, and a message cannot be forged through a known or chosen signature.

6.1 Unforgeability of a Signature in the Random Oracle Model

Because REESSE1+ is a multiproblem cryptosystem, and MPP preventing a forged signature from being obtained from the signature algorithm is different from TLP or PRFP preventing a forged signature from being obtained from the verification algorithm, the proof in the random oracle model (RO model) given below is incomplete, and only offers another piece of evidence for the security of the REESSE1+ signature.

6.1.1 What Is 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 [25][38]. 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 [38].

Random oracles are utilized in cryptographic proofs for replacing any irrealizable function so far which can provide the mathematical properties required by the proof. A cryptosystem or a protocol that is proven secure using such a proof is described as being secure in the RO model, as opposed to being secure in the standard model where IFP, DLP 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 RO model, but are trivially insecure when any real function such as the hash function MD5 or SHA-1 is substituted for the random oracle [39]. Nevertheless, for any more natural protocol, a proof of security in the RO model gives very strong evidence that an attacker have to discover some unknown and undesirable property of the hash function used in the protocol.

6.1.2 The Forking Lemma

Lemma 1 (the forking lemma): Let $A$ be a probabilistic polynomial time Turing machine, given only the public data as input. If $A$ can find, with non-negligible probability, a valid signature $(\sigma_1, \sigma_2, \ldots, \sigma_{2n})$, then, with non-negligible probability, a replay of this machine, with the same random tape and a different oracle, outputs two valid signatures $(\sigma_1, \sigma_2, \ldots, \sigma_{2n})$ and $(\sigma_1, \sigma_2, \ldots, \sigma_{2n})$ such that $b \neq b'$ [40].

In [40], the forking lemma is specified in terms of an adversary that attacks a digital signature scheme instantiated in the RO model. D. Pointcheval and J. Stern show that if an adversary can forge a signature with non-negligible probability, then there is a non-negligible probability that the same adversary with the same random tape can create a second forgery in an attack with a different random oracle. The forking lemma was later generalized by M. Bellare and G. Neven [41], and has been used to prove the security of a variety of digital signature schemes and other cryptographic constructions based on random oracles [42].

The forking lemma is applicable to such a type of signature scheme where a signer must perform the following steps to sign the message $\sigma$: 

> http://arxiv.org/pdf/cs/0702046 (revised Dec 2012) <
randomly produce a promise \( \sigma_1 \) in a large set;
2. compute \( h = \text{hash}(\sigma_1, q) \);
3. compute \( \sigma_2 \) by using \( \sigma_1 \) and \( h \).

The signature output is a triple \((\sigma_1, h, \sigma_2)\).

For example, the Schnorr signature scheme [43] may be proven secure in the RO model according to the forking lemma [40][42].

Assume that solving \( y = g^r \) \((\% M)\) for \( x \) is of DLP, where \( g \) is a generator of \( \mathbb{Z}_M \) with \( M \) prime, and the input to an adversary \( A \) is the public key \( y \).

In terms of the forking lemma, the adversary can obtain two different signatures with non-negligible probability: \((q, \sigma_1 = g^r \) \((\% M), h, \sigma_2 = (xh + r) \) \((\% M)\), and \((q, \sigma_1' = g^r' \) \((\% M), h', \sigma_2' = (xh' + r') \) \((\% M)\),

where \( h = \text{hash}(\sigma_1, q) \) is an oracle query of the adversary \( A \) during his first play, and \( h' = \text{hash}(\sigma_1', q) \) is an oracle query of the adversary \( A \) during his second play.

The target of the adversary \( A \) is to extract the private key \( x \) form the signature.

Since there is \( \sigma_1 = \sigma_1' \), there is \( r = r' \). Hence, we have \((h - h')x = (\sigma_2 - \sigma_2') \) \((\% M)\).

Because there always exists \( x \), the above linear congruence in \( x \) has solutions, and \( x \) can be found in polynomial time, namely DLP can be solved in polynomial time, which is in direct contradiction to the standard assumption. Therefore, it is infeasible to forge a Schnorr signature.

6.1.3 Proof of Unforgeability of a REESSE1+ Signature

In Section 4.1, we prove that MPP is computationally at least equivalent to DLP, and additionally at present, a subexponential time algorithm for solving MPP is not found. Therefore, we may have a superstandard assumption comparable to the standard assumption.

**Assumption 1:** MPP cannot be solved in subexponential time.

Firstly, we adapt the REESSE1+ signature algorithm to the modality outputting a triple signature.

Assume that \((\{A_i\}, l(i))\), \(W, \tilde{D}, d, h\) is a private, and \(F\) is a file or message to be signed.

Define \(\text{hash}(F, \tilde{a}) \) as \(\text{hash}(F, \tilde{a}) = (\tilde{a}D + \text{hash}(F))W^{-\delta} \) \((\% M)\).

**S1:** Let \( H \leftarrow \text{hash}(F)\), whose binary form is \( b_1 ... b_n\).

**S2:** Randomly take \( \tilde{a} \in [1, M]\), compute \( Q = \text{hash}(F, \tilde{a})\).

If \( \text{d}(\text{d}T) \not\mid \tilde{a} \) or \( \text{d}(\text{d}W) \% M \), goto S2.

**S3:** Set \( \delta = \sum_{i=1}^{n}b_i(l(i)) \% M, G_0 = \left\{\sum_{i=1}^{n}b_i(l(i)) \% M\right\}.

**S4:** Compute \( R \leftarrow (Q(\delta h^{-1})^{1, S_i}G_0^{-1}, \tilde{U} \leftarrow (RW^{-\delta}G_0^{1, S_i}hW^{\delta} \% M), \tilde{G} \leftarrow \delta^{\alpha^{-1}} \% M, \tilde{Z} \leftarrow \sum_{i=0}^{\delta}(Q\delta h^{-1})(\% H/W) \% M)\).

**S5:** \( \forall r \in [1, 2^{10}] \), making \( d \mid (rUS + \tilde{Z}) \% M, \text{ where } U = \Upsilon \tilde{G} \% M)\).

**S6:** If \( d \not\mid (\text{d}W^{-1} + rUS + \tilde{Z}) \% M \), goto S6, else end.

On executing this algorithm which is essentially equivalent to that in Section 3.4, one can obtain the signature \((\tilde{a}, Q, U)\) that is sent to a receiver together with \(F\).

Secondly, probe the unforgeability of a signature in the RO model.

**Proof:**

Let \( \tilde{a} \) be a promise produced by the signer. In terms of the forking lemma, during the two plays of the adversary \( A \), two valid signatures \((\tilde{a}, Q, U)\) and \((\tilde{a}', Q', U')\) on the identical file \( F \) can be obtained. Then, it is known from the above signature algorithm that

\[
\delta Q = \delta D + WH \% M, \quad \delta Q' = \delta D' + WH' \% M, \quad U = (RWW^{-\delta}G_0^{1, S_i}hW^{\delta} \% M). \quad U' = (RWW^{-\delta}G_0^{1, S_i}hW^{\delta} \% M), \]

where \( \tilde{a} = \tilde{a}', H = \text{hash}(F, \tilde{a}), \text{and } H' = \text{hash}(F, \tilde{a}')\).

Because there always exist \( W \), it may be obtained from the two formulas with \( H \) and \( H' \) that

\[
W = (Q - Q')(H - H')^{-1} \% M. \tag{2}
\]

Similarly, it may be obtained from the two formulas with \( U \) and \( U' \) that

\[
(RWW^{-\delta}G_0^{1, S_i}hW^{\delta})^{-1} = UL^{-1} \% M. \tag{2}
\]

Multiplying either side of the above equation by \( G_0^{1, S_i}Q \) gives

\[
(RWW^{-\delta}G_0^{1, S_i}hW^{\delta})^{-1} = UL^{-1}G_0^{1, S_i} \% M. \tag{2}
\]

Because there exists \( \tilde{a} \), \( \tilde{a}' \) may be obtained in subexponential time through a heuristic approach, namely \( \delta = \tilde{a}' \% M \). In addition, by using \( C_i = (A_i W^{\delta})^{1, S_i} \% M, A_i \in \{2, 3, ..., 1201\}, l(i) \in \{5, 7, ..., 2n + 1\}, \) and \( \{A_1, ..., A_n\} \) and \( \{l(1), ..., l(n)\} \) in expected subexponential time.

Further, according to \( \tilde{U} = (RWW^{-\delta}G_0^{1, S_i}hW^{\delta} \% M) \), \( h \) may
be found. In sum, the private key \((A_i, \{\ell(i)\}, W, \delta, D, d, h)\) is found, namely MPP is also solved in subexponential time, which is in direct contradiction to assumption 1. Therefore, it is infeasible in subexponential time that the adversary forge a REESSE1+ promise signature \((\delta, Q, U)\).

In practice, the promise \(\delta\) is unpublicized, and therefore, it is at least the same infeasible in subexponential time that \(\hat{A}\) forges a REESSE1+ signature \((Q, U)\).

Notice that the REESSE1+ verification algorithm is built on the hardnesses different from MPP, so this proof is incomplete for the unforgeability of a REESSE1+ signature, and further security analysis is necessary.

### 6.2 Extracting a Private Key from a Signature Is of TLP or the Indeterminate Problem

To analyze exact securities, we attend to the solution of \(x^k = c \mod M\) with \(M\) prime called RFP for a while. In some cases, \(x^k = c \mod M\) has the trivial root which can be found in terms of Theorem 1 or 2. At present, there should be three methods of solving \(x^k = c \mod M\): \(\delta\) the algorithm through discrete logarithms [25] whose running time is \(L(1, 1, 1.923)\); \(\delta\) the probabilistic algorithm [44] whose running time is the larger of \(O(2^{k-1})\) and \(O(M/k)\); \(\delta\) the algorithm offered by [45] whose running time is \(O(k^{1/2} \lg M)\). Obviously, when \(k\) is comparatively small, method 3 is most efficient.

In our analysis, we assume that RFP can be solved in tolerable time.

It is known from Section 3.4 that there exist

\[
Q = (RG_0)^\delta h \mod M, \quad U = (RW^{\delta})^{\delta^2 \delta^k} \mod M.
\]

Firstly, solving the two congruences separately.

According to Section 3.4, \(\delta^{2 \delta^k} \mod M\) belongs to the subgroup of order \(d\). Because of \(T \geq 2^k\), dividing the value of \(\delta^{2 \delta^k}\) \(\mod M\) is impossible.

Let \(\tilde{e} = \delta^{2 \delta^k} \mod M\) be an element of the subgroup of order \(d\), then latter may be written as

\[
U^T = (RW^{\delta})^{\delta^2 \delta^k} \tilde{e} \mod M.
\]

When an attacker attempts to seek \(RG_0\) or \(RW^{\delta}\), he has to solve the equation

\[
x^k = Q \delta^h \mod M, \quad (4)
\]

or

\[
x^T \tilde{e}^T = U^T \tilde{e} \mod M, \quad (5)
\]

We see that the number of unknown variables in the two equations is greater than 2, and thus solving (4) and (5) is of the indeterminate problem.

For (4), because \(\delta, h\) are unknown, and the right of (4) is not a constant, it is impossible to solve this equation for \(RG_0\). If \(\delta\) is divided, the probability of hitting \(\delta = 1\) is \(\ll 1/2^k\). Similarly it is impossible to divide \(h\) owing to \(h \in (1, M)\).

For (5), there is \(|\tilde{e}| \leq d\). Assume that \(d\) is guessed out, and all the solutions to \(x^k = 1 \mod M\) can be found out, then \(\tilde{e}^T\) may possibly be figured.

However, even if \(\tilde{e}^T\) is known, and the trivial root to (5) exists, the probability that the trivial root just equals the specific \(RW^{\delta}\) is only \(1/T \leq 1/2^k\), and moreover due to the randomness of \(R\), it is thoroughly impossible to separate \(\delta, W, \delta^k\) from \(RW^{\delta}\).

Secondly, solving the two congruences systematically.

Substituting \(R\) in \(U^T = (RW^{\delta})^{\delta^2 \delta^k} \tilde{e} \mod M\) with

\[
Q^{-1} (Q \delta^h)^{x^k} \mod M
\]

derived from \(Q = (RG_0)^\delta \mod M\) gives

\[
U^T = (Q^{-1} (Q \delta^h)^{x^k})^{\delta^2 \delta^k} \tilde{e} \mod M,
\]

namely

\[
U^T = (G^{-1} (Q \delta^h)^{x^k} W^{\delta})^{\delta^2 \delta^k} \tilde{e} \mod M,
\]

where \(G = G_0 G_1 \mod M\). Since \(W\) is a function of \(\delta = \delta^y \mod M, y = W^\delta \mod M\) is of TLP.

Thus,

\[
((GW^\delta)^{-1} (\delta)^{-1})^{\delta^2 \delta^k} \tilde{e} = U^T (\delta)^{-1} \tilde{e} \mod M.
\]

Similarly, if \(\tilde{e}^T\) is known, and \((U^T (\delta)^{-1} \tilde{e})^{\delta^2 \delta^k} \tilde{e} \mod M\) gives

\[
U^T = (G^{-1} (Q \delta^h)^{x^k} W) \tilde{e} \mod M,
\]

where \(G = G_0 G_1 \mod M\). Since \(W\) is a function of \(\delta = \delta^y \mod M, y = W^\delta \mod M\) is of TLP.

Thus,

\[
((GW^\delta)^{-1} (\delta)^{-1})^{\delta^2 \delta^k} \tilde{e} = U^T (\delta)^{-1} \tilde{e} \mod M.
\]

Furthermore, mapping from \(\tilde{e}\) to \(\tilde{e}^T\) may possibly be figured.

### 6.3 Forging a Digital Signature only from a Public Key Is a Hardness

According to Section 3.5, the discriminant \(X = Y \mod M\) contains the two variables \(Q\) and \(U\) of which one may be supposed in advance by an adversary. However, seeking the other by the supposed value is faced with TLP.

#### 6.3.1 TLP Is at Least Equivalent to DLP

We observe \(y = g^x \mod M\) referred to as TLP by Definition 7.

Assume that \(g \in \mathbb{Z}_M\) is a generator, where \(M\) is prime, then

\[
\{y \mid y = g^x \mod M, x = 1, \ldots, M\} = \mathbb{Z}_M^* [22].
\]

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

\[
\{y \mid y = x^k \mod M, x = 1, \ldots, M\} = \mathbb{Z}_M^* [22].
\]

Therefore, \(\forall x \in \{1, \ldots, M\}, y = g^x \mod M\) or \(y = x^k \mod M\) with \(\gcd(k, M) = 1\) is a self-isomorph of the group \(\mathbb{Z}_M^*\).

However, for the \(x\) operation, \(\{y \mid y = x^k \mod M, x = 1, \ldots, M\} = \mathbb{Z}_M^*\) does not hold, that is,

\[
\{y \mid y = x^k \mod M, x = 1, \ldots, M\} \neq \mathbb{Z}_M^*.
\]

For example, when \(M = 11\), \(\{y \mid y = x^6 \mod M, x = 1, \ldots, M\} = \{1, 3, 4, 5, 6\}\), where \(3^6 = 6^8 = 8^5 = 5 \mod 11\).

When \(M = 13\), \(\{y \mid y = x^9 \mod M, x = 1, \ldots, M\} = \{1, 3, 4, 5, 6, 9, 12\}\), where \(7^9 = 11^{12} = 6 \mod 13\), and \(1^8 = 2^{12} = 9^9 = 12^{12} = 1 \mod 13\).

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

The above examples illustrate that \(\{y \mid y = x^k \mod M, x = 1, \ldots, M\}\) 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 nonexistent. Thus, \(x^k\) has extremely strong irregularity, and is essentially distinct from \(g^x\) and \(x^k\).

It should be noted that an attempt at solving \(y = x^k \mod M\)
for $x$ in light of the Chinese Remainder Theorem is spurious. Refer to the following example.

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

Try to seek $x$ which satisfies $x = 12 \pmod{30}$ and $x = 3 \pmod{31}$ at one time, and verify whether $x = 4 \pmod{31}$ or not.

In light of the Chinese Remainder Theorem [25], 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 $M_1 y_1 = 1 \pmod{m_1}$. Compute $y_2 = 30$ such that $M_2 y_2 = 1 \pmod{m_2}$.

Thereby,

$$x = a_1 M_1 y_1 + a_2 M_2 y_2 = 12 \times 31 \times 1 + 3 \times 30 \times 30 = 282 \pmod{930}.$$

It is not difficult to verify

$$282^{288} \equiv 8 \not\equiv 4 \pmod{31},$$

and

$$282^{288} = 504 \equiv 4 \pmod{930}.$$

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

So, these examples manifest that TLP seems to be harder than DLP.

**Property 7**: TLP is computationally at least equivalent to DLP in the same prime field.

**Proof**:

Let $g \in \mathbb{Z}_{M}^*$ be a generator coprime to $\mathcal{M}$, which does not lose generality since $g$ may be selected in practice.

Assume that $y \in \mathbb{Z}_{M}$ is known, and there is

$$y = (g x)^y \pmod{M}.$$ 

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

$$y^w = (g x)^{y^w} \pmod{M}.$$ 

Let $z = y^w \pmod{M}$, and $w = g x$, where the latter is not a congruence, and then

$$z = w^w \pmod{M}.$$ 

Suppose that $\hat{O}(y, M, X)$ is an oracle on solving $y = x^y \pmod{M}$ for $x$, where $X$ is the set of all the possible values of $x$, and $y \in [1, \mathcal{M}]$.

Its output is $x \in X$ (each of solutions), or 0 (no solution).

Let $X_1 = \{1, 2, \ldots, \mathcal{M}\}$, and $X_2 = \{1 g, 2 g, \ldots, \mathcal{M} g\}$.

Clearly, by calling $\hat{O}(y, M, X_1), y = x^y \pmod{M}$ is solved for $x$.

It is easily observed that between the finite sets $X_1$ and $X_2$, there is a linear bijection

$$\Gamma: X_1 \rightarrow X_2, \Gamma(a) = g a,$$

which means that the set $X_1$ is equivalent to the set $X_2$ [46]. Hence, substituting $X_1$ with $X_2$ as the codomain of a function will not increase the running time of $\hat{O}$.

Similarly, by calling $\hat{O}(z, M, X_2), z = w^w \pmod{M}$ is solved for $w$, namely all the satisfactory values of $w$ are obtained.

According to $w = g x$ and $z = w^w \pmod{M}$, there is $x = w g^{-1} \pmod{M}$, or $x = w g^{-1} \pmod{\mathcal{M}}$.

Hence, in terms of Definition 8, there is

$$\hat{H}(y = (g x)^y \pmod{M}) \leq T \hat{H}(y = x^y \pmod{M}),$$

namely the difficulty in inverting $y = (g x)^y \pmod{M}$ is not greater than that in inverting $y = x^y \pmod{M}$.

On the other hand, suppose that $\hat{O}(y, g, M)$ is an oracle on solving $y = (g x)^y \pmod{M}$ for $x$, where $y, g \in [1, \mathcal{M}]$.

Its output is $x \in [1, \mathcal{M}]$ (each of solutions), or 0 (no solution).

Let $g = 1$.

By calling $\hat{O}(y, 1, M)$, the solution $x$ to $y = x^y \pmod{M}$ will be obtained.

Hence, in terms of Definition 8, there is

$$\hat{H}(y = x^y \pmod{M}) \leq T \hat{H}(y = (g x)^y \pmod{M}).$$

Combinatorially, in terms of Definition 9, we have that

$$\hat{H}(y = g^y \pmod{M}) \leq T \hat{H}(y = (g x)^y \pmod{M}),$$

namely the difficulty in inverting $y = (g x)^y \pmod{M}$ is equivalent to that in inverting $y = x^y \pmod{M}$.

The congruence $y = (g x)^y \pmod{M}$ may be written as $y = g^y \pmod{M}$, where $g$ is any element.

Change $\hat{O}(y, g, M)$ into $\hat{O}(y, g, M, \hat{w})$, where $\hat{w} = 0$ or 1.

Its structure is as follows:

S1: If $\hat{w} = 1$ and $x \to y = g^x \pmod{M}$ inexistent, return ‘No’.

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

S2.1: find $y_1$, and compute $y_2$ by $y = y_1 y_2 \pmod{M}$;

S2.2: compute $x < M$ by $y_1 = g^x \pmod{M}$;

S2.3: if $y_2 \not\equiv x \pmod{M}$, goto S2.1; else

S2.4: compute $x < M$ by $y = g^x \pmod{M}$.

S3: Return $x$.

Clearly, by calling $\hat{O}(y, g, M, 0)$, the solution $x$ to $y = g^y \pmod{M}$ will be obtained.

Hence, still in terms of Definition 8, there is

$$\hat{H}(y = g^y \pmod{M}) \leq T \hat{H}(y = x^y \pmod{M}).$$

Integrating $\oplus$ and $\otimes$, we have that

$$\hat{H}(y = g^y \pmod{M}) \leq T \hat{H}(y = g^x \pmod{M}) = T \hat{H}(y = x^y \pmod{M}),$$

namely inverting $y = x^y \pmod{M}$ is at least equivalent inverting $y = g^y \pmod{M}$ for $x$ in complexity.

In [47], we have a similar result by the asymptotic granularity reduction (AGR).

Further discussion.

Let $y = g^x \pmod{M}$, and $x = g^w \pmod{M}$, and then it seems that there is $g^w = g^{g^x} \pmod{M}$.

However due to $g^w \pmod{M} \neq g^w \pmod{\mathcal{M}}, y = x^y \pmod{M}$ cannot be expressed as $v = u^g \pmod{\mathcal{M}}$.

We can also understand that in the process of $x$ being sought from $y = x^y \pmod{M}$, it is inevitable that the middle value of $x$ is beyond $M$ because modular multiplication, inverse, and power operations are inevitable.

Considering the middle value of $x$ beyond $M$, let $z_1 = x \pmod{M}$, $z_2 = x \pmod{\mathcal{M}}$.

Then there are $x = z_1 + k_1 M = z_2 + k_2 \mathcal{M}$ and $z_1 = (z_2 - k_1) \pmod{M}$, where $k_1, k_2 \geq 0$ are two integers. Ahead, we have $y = (g (z_2 - k_1))^z \pmod{M}$. This formula indicates that due to $x \pmod{M} = x \pmod{\mathcal{M}}$ with $x > M$ and $k_2$ unfixable, the relation between $x \pmod{\mathcal{M}}$ and $x \pmod{M}$ is stochastic when $x$ changes in the interval $(1, M^2)$, which illuminates that it is reasonable to let $q = g (z_2 - k_1) \pmod{\mathcal{M}}$, namely $y = q^z \pmod{\mathcal{M}}$.

If $q$ is a constant, inverting $y = q^z \pmod{\mathcal{M}}$ is equivalent to DLP. However, $q$ will not be a constant forever. Therefore, it should be impossible anyway that $\hat{H}(y = (g x)^y \pmod{M}) \leq T \hat{H}(y = g^y \pmod{M})$.

The above evidence inclines us to believe that TLP is
harder than DLP, namely on the assumption that DLP can be solved through an oracle, TLP cannot 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 [25] are ineffectual on transcendental logarithms. At present, there is no better method for solving TLP than exhaustive search, and thus the running time of solving $x^y = y \; (\% \; M)$ may be expected to be $O(M) > O(2^n)$.

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

In $\text{REEESSE1}^+$, the form of the TLP is $y = (cx)^y \; (\% \; M)$ with $c$ known. When the bit-length of a modulus is very small — 80 for example, the difference between the running times of solving $y = (cx)^y \; (\% \; M)$ and $y = x^y \; (\% \; M)$ is valuable, where $cx$ changes with $c$, and has more freedom than a single $x$, which makes the relation between $|y|$ and $|x|$ be more indeterminate.

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

6.3.2 Forging a Signature from the Verification Algorithm Is of TLP or PRFP

Assume that $H$ is the output of hash on input of a file $F$, and $(Q, U)$ is a signature on $F$. According to the discriminant $X = Y \; (\% \; M)$, namely

\[(\alpha Q^\delta)^{O_{UT}} = (\delta^Q U_{\alpha}^{1, \delta_{ST}}) \; (\% \; M),\]

an adversary may seek the value of any signature variable by supposing the value of the other variable. If the value of $Q$, no matter whether $U$ exists or not, seeking $U$ is equivalent to TLP.

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

\[\text{① Faced with RFP and the tight constraint} \]

If the adversary hits exactly the small $d$, raises either side of the discriminant to the $d$-th power, and assumes that $D \; (\% \; WH)$, then there is

\[\alpha Q^\delta)^{O_{UT}} = (\delta^Q U_{\alpha}^{1, \delta_{ST}}) \; (\% \; M).\]

Further, let

\[\alpha Q^\delta)^{O_{UT}} = (\delta^Q U_{\alpha}^{1, \delta_{ST}}) \; (\% \; M).\]  

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

\[U^{\delta_{ST}} = ((\alpha Q^\delta)^{O_{UT}} S^{-1}) \; (\% \; M)\]

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 $D \; (\% \; WH)$. The probability of satisfaction is at most $1 / D < 1 / 2^n$ when $\delta$ and $W$ are unknown. Notice that successive values of $Q$ cannot guarantee the integral solution of $D \; (\% \; WH)$, and $(Q, U) = ((\alpha Q^\delta)^{O_{UT}} S^{-1}) \; (\% \; M)$ does not satisfy the discriminant.

\[\text{② Faced with PRFP} \]

We observe that $(\alpha Q^\delta)^{O_{UT}} S^{-1} \; (\% \; M)$ is an element of the subgroup of order $d$.

Assume that $\bar{e}$ is a solution to $x^d = 1 \; (\% \; M)$, and $g$ is a generator of $\mathbb{Z}_M^*$. Evaluate $u, v, q$ by the Index-calculus such that $g^v = \alpha \; (\% \; M), g^u = \beta \; (\% \; M),$ and $g^q = \bar{e} \; (\% \; M)$.

Then

\[g^\bar{e} = g^{-\langle \delta (Q^\delta)^{O_{UT}} S^{-1} \rangle} = g^\alpha \; (\% \; M),\]

namely

\[uQ^\delta - vH (Q^\delta)^{O_{UT}} - q = 0 \; (\% \; M)\]

which is a modular polynomial equation in $Q$.

If this polynomial equation has solutions, and $Q$ can be figured out, $U$ may be evaluated according to (6). In this wise, $Q$ and $U$ which likely meet the discriminant can be found.

Solving (7) for $Q$ is of PRFP for which a generic subexponential algorithm is not found so far, and in terms of AGR [47], PRFP is believed to be harder than RFP. Besides, because gcd$(u, \mathbb{M}) > 1$ is fully possible, namely (7) is not necessarily a polynomial of which the first term coefficient is 1, and there exists $\mathbb{M}^{Q^\delta} \in (2^{280/80}, 2^{1216/128}) \in (2^{8}, 2^{5}, 8)$, attack (7) is ineffectual by the Coppersmith reduction which can find sufficiently small solutions, whose absolute values are less than $\mathbb{M}^{1 / 8}$, to a modular univariate polynomial [48] if such solutions exist. Again, if the adversary solves (7) through the probabilistic algorithm in Section 1.6 of [44], the time complexity will be $O(\mathbb{M} / n) > O(2^n)$.

③ Unmalicious subgroup of order $d$

We observe from (6) that if $d$ is guessed accurately, and gcd$(U, \mathbb{M}) = 1$, holds there is

\[\langle \alpha Q^\delta \rangle^{O_{UT}} (\delta^Q U_{\alpha}^{1, \delta_{ST}}) = 1 \; (\% \; M),\]

which implies that the element $(\alpha Q^\delta)^{O_{UT}} (\delta^Q U_{\alpha}^{1, \delta_{ST}})$ belongs to the subgroup of order $d$. Therefore, if gather many enough signatures $(Q, U)$, all the elements of the subgroup are likely found out. However, the analysis in Section 6.2 shows that even if this case occurs, it does not influence the security of a $\text{REEESSE1}^+$ signature.

Further, through gathering more enough signatures or following the Index-calculus method, all the elements of the subgroup of order $d$ are likely found out. They can be described with a general expression in the time $L_M[1 / 3, 1.923]$, but picking out a specific element will take the time $O(dT) > O(2^n)$ since we must try all the elements one by one.

6.3.3 Forging a Signature from the Signature Algorithm Is of TLP or PRFP

Owing to $Q = (RG_0)^{\delta h} \; (\% \; M), U^T = (RW^{\delta h})^{\delta e} \; (\% \; M)$, and $V = (R^{-1} W^T G_i^{\delta}) \; (\% \; M)$ (see Section 3.4), an adversary may attempt the following attack approach.

Let

\[Q = a^\delta h \; (\% \; M), U^T = b^\delta e \; (\% \; M), V = c^{\delta} \; (\% \; M)\]

where $\lambda S = ((WQ)^{\delta_1} + \xi + rUS)(\delta Q - HW)(\% \; M)$.

Correspondingly, there are $ac = (\alpha \delta^{-1} h)^{\delta_1}$, and $bc = \delta_1 \; (\% \; M)$.

If $\bar{e}$ is hit, $(Q, U)$ is a known signature, and $(U^T \bar{e})^{\delta h}^{\delta_1} (\% \; M) = 1 \; (\% \; M)$ holds, a solution $b$ to the equation $b^\delta = U^T \bar{e} \; (\% \; M)$ can be found through either the Index-calculus method or the trivial root method. Further, $c$ can be figured from $bc = \delta_1 \; (\% \; M)$. However, it is impossible to find $a$ from $ac = (\alpha \delta^{-1} h)^{\delta_1} \; (\% \; M)$ since $\delta, h$ cannot be obtained from $Q, U,$ and $V$.

If $\delta, h$ are found, then $c, b$ can be evaluated on assuming a
value of $a$. Further, if $D$ is extracted from $\mathcal{M}$, $W$ is figured, and $r$ is guessed, the adversary may compute the values of $Q$ and $U$ which satisfy

$$D \mid (\delta Q - WH) \pmod{\mathcal{M}} \text{ and } d \mid ((WQ)^{\theta-1} + \xi + r U S) \pmod{\mathcal{M}}.$$  

If $\delta$, $W$ can be found, $h$ may possibly be computed according to $\alpha = \delta h (W^G_G S, y)^{\delta} \pmod{\mathcal{M}}$.

The above analysis shows that the acquiring $\delta$ and $W$ is the key to the problem. To seek $\delta$ and $W$ from a known clear clue, the adversary has to try to solve the simultaneous equations

$$\begin{aligned}
\alpha & = \delta^{W+1} \pmod{\mathcal{M}} \\
\beta & = \delta^{W+1} \pmod{\mathcal{M}}.
\end{aligned}$$

Obviously, the first equation is computationally at least equivalent to TLP. The second equation contains two variables, and belongs to the indeterminate problem. Raising the two values of $\delta$ and $W$ to the power of $\theta$ yields

$$\alpha^\theta = \delta^{(\theta + \theta W - 1)^\theta} \pmod{\mathcal{M}},$$

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

Let $g$ be a generator of the group $(\mathbb{Z}/n\mathbb{Z})^\times$. By the Index-calculus for discrete logarithms [25], evaluate $u$, $v$, and $x$ such that $g^x = \alpha$, $g^y = \beta$, and $g^z = \delta \pmod{\mathcal{M}}$ (notice that the latter does not mean $g^z = \delta \pmod{\mathcal{M}}$). Then, we have

$$\begin{aligned}
u & = xT(\delta^\theta + \delta W^{\theta - 1}) \pmod{\mathcal{M}} \\
v & = xT w^{-1} \pmod{\mathcal{M}}.
\end{aligned}$$

If $x$ is guessed, the two values of $W$ may possibly be obtained according to the above two equations. Nevertheless, the two values of $W$ are not necessarily equal to each other, which indicates that the value of $x$ is not right.

If the inverse of $W \pmod{\mathcal{M}}$ exists, there is $xT = vW^{-1} \pmod{\mathcal{M}}$. Hence, we have

$$\begin{aligned}
u & = vW^{-1}(\delta^\theta + \delta W^{\theta - 1}) \\
v & = v(\delta W^{\theta - 1} + \delta W^{\theta - 1}) \pmod{\mathcal{M}} \\
v & = v(\delta^\theta + v) \pmod{\mathcal{M}},
\end{aligned}$$

where $y = \delta W^{\theta - 1} \pmod{\mathcal{M}}$. Notice that $y$ is of great value to the adversary if it can be found, for the adversary may obtain $Q = y^{-1} H \pmod{\mathcal{M}}$ as he lets $\delta Q - WH = 0 \pmod{\mathcal{M}}$ according to $D \mid (\delta Q - WH)$.

The above congruence is of PRPF which is very intractable, and believed to be harder than RFP [47]. Additionally, because gcd($v, \mathcal{M}$) $> 1$ is fully possible, and there exists $\mathcal{M}^{1/n} \in (2^{100/80}, 2^{121/128}) = (2^{8.7}, 2^{9.5})$, attack on the above congruence is ineffectual by the Coppersmith reduction [48]. If you are afraid of the Coppersmith reduction in practice, the exponent $n$ independent of the block length may be substituted with a larger integer.

If $\mathcal{M}$ is factorized, $D$ is obtained, and $W^{\theta - 1} \pmod{\mathcal{M}}$ exists, there is $u = v(\delta^\theta + y) \pmod{\mathcal{M}}$, where $D$ contains a prime not less than $2^t$. When the Coppersmith reduction is useless, the adversary may attempt the exhaustive search. Nevertheless, its running time is close to $D \geq 2^t$.

Again it is known from the key generation algorithm that there is $W = (\prod_{i=1}^{n} A_i)^{-1} (\alpha \delta^{-1} h^{-1})^{\delta(b - 1)} \pmod{\mathcal{M}}$. However, such a substitution of $W$ will not make the simultaneous equations reduced since $\prod_{i=1}^{n} A_i$ is unknown, and $W(\prod_{i=1}^{n} A_i)$ is the $(S\delta)^{-1}$-th power of the unknown quantity $(\alpha \delta^{-1} h^{-1})$.

### 6.4 Forging a Signature from Known Signatures with a Public Key Is of TLP or Hash

Given a file $F$ and a signature $(Q, U)$ on it, and assume that there exists another file $F'$ with the related $H$ and $G_{i'}$. If any arbitrary $(Q', U')$ satisfies 

$$(\alpha Q'^{-1})^{\delta U'} \pmod{\mathcal{M}} = (\delta G_{i'}^{-1} U'^{-1})^{\beta H_{i'}^{\theta + 1} + H' \pmod{\mathcal{M}},}$$

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 $D \mid (\delta Q' - WH)$, 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 $D \mid (\delta Q_1 - WH_1)$ and $D \mid (\delta Q_2 - WH_2)$, we see that 

$$D \mid (\delta Q_1 - Q_2) - WH(1 + H_2)).$$

Let $Q' = Q_1 + Q_2$. If $H' = H_1 + H_2$, then $D \mid (\delta Q' - WH'F')$. However, inferring $F'$ from $H'$ is intractable according to the properties of hash functions. In addition, finding a fit $U'$ from 

$$U'^{td} = ((\alpha^{-1} Q')^{\theta^{-1}} G_{i'} Q')^{\gamma} \pmod{\mathcal{M}}$$

is also intractable since $U'$ has $T d$ values.

If many of the pair $(Q, U)$ are gathered, because $Q$ is random, $Q$ and $U$ interrelate through a transcendental logarithm, and the value of $U$ varies intensely between 1 and $M$, there is no polynomial function or statistic regularity among different $(Q, U)$'s, which indicates that they are unhelpful in solving TLP, but yet helpful in finding out the elements of the subgroup of order $d$ or $dT$ as is pointed out in Section 6.2.

Thus, forging a signature through known signatures with a public key is of TLP or the hash$^{-1}(H)$.

### 6.5 Adaptive-chosen-message Attack Is Faced with Indistinguishability

In accordance with Section 3.4, $Q$ satisfies $D \mid (\delta Q - WH)$, namely $Q = (\bar{a} D - WH)^{-1} \pmod{\mathcal{M}}$, where $\bar{a}$ is a random integer, and meets $(dT) \bar{a}$.

The randomness of $\bar{a}$ leads $Q$ to be random while $U$ is interrelated with $Q$ in a transcendental logarithm, where $Q = (RG_{i'})^i \delta h \pmod{\mathcal{M}}$, and $U = (R W_{\theta - 1})^i \delta h \pmod{\mathcal{M}}$.

Hence, for the identical file $F$, there will be many different signatures on it, which manifests that the signature $(Q, U)$ owns indistinguishability.

In terms of [38], the signature $(Q, U)$ on $F$ is secure against adaptive-chosen-message attack.

### 6.6 Chosen-signature Attack Is Faced with PRFP

It is well understood from the discriminant that 

$$Q_{i'}^{\theta U' \pmod{\mathcal{M}}} = (\alpha Q^{-1})^{\theta U' \pmod{\mathcal{M}}} \pmod{\mathcal{M}},$$

(8)  

If the values of $Q$ and $U$ are chosen in advance, an adversary may attempt to figure out $H$ and the corresponding file or message $F$.

Let $G_{i'} = h (H) = \prod_{i=1}^{n} C_i^{h_i} \pmod{\mathcal{M}}$, where $H = b_1, ..., b_n = \sum_{i=1}^{n} h_i 2^{n-i}$, then (8) is an equation in $b_1, ..., b_n$.

Furthermore, let $g$ be a generator of $(\mathbb{Z}/n\mathbb{Z})^\times$, and through the Index-calculus method, work out $q_1, ..., q_m, v, u, w$ such that
\[g^h = C_1 \mod M, \ldots, g^e = C_n \mod M, \quad g^{w} = \beta \mod M, \quad g^{\alpha^{-1}Q^{\delta}} U^{\alpha^{\delta}} \mod M, \quad w = \text{QUST} \mod M.\]

Then, there is
\[(q_1 b_1 + \ldots + q_n b_n)^{w+1} \cdot v(Q^{\alpha^{-1}} \sum b_i 2^{\alpha^{-i}} + (\sum b_i 2^{\alpha^{-i}})) = u \mod M,\]

which is of the multivariate polynomial root finding problem (Multivariate PRFP). Clearly, even though \(H\) is found, it is infeasible to infer a \(F\) from \(H\).

On the other hand, if there exists the inverse function \(H = f^{-1}(\mathcal{G})\), namely \(H\) in (8) is substituted with \(\mathcal{G}\), evaluating \(\mathcal{G}\) from (8) is the combination of RFP, DLP and PRFP.

7 Conclusion

REESSE1+ is only a prototypal cryptosystem which is used for explaining some concepts, ideas, and methods, so 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 the numerical calculation ability but also the logic judgement ability of a computer is utilized; thus the reversibility of the functions is relatively poor.

MPP which contains indeterminacy is a composite problem integrating IFP with DLP. ASPP is also a composite problem integrating IFP, DLP with ASSP, where ASSP can resist the L3 lattice base reduction. TLP is a primitive problem which may be regarded as consisting of two variables. PRFP is also a primitive problem which contains both addition and multiplication — \(ax^a + bx^{a-1} + cx + d = 0 \mod M\) with \(n \geq 80\), \(a \neq 0, 1, d \neq 0\), and \(|b| + |c| \neq 0\) for example. So far, a generic subexponential algorithm for solving MPP, ASPP, TLP, or PRFP is not found. Due to indeterminacy, even as \(\log M \approx 80\), solving MPP, ASPP, TLP, or PRFP for the original answer is infeasible in subexponential time yet. Notice that \(\log M \approx 80\) indicates that the constraint \(M > (\max_{0 \leq i < k} A_i)^9\) is removed from the key generator, and REESSE1+ is only used for digital signature.

Some evidence given in the paper inclines people to believe that MPP, ASPP, and TLP are harder than DLP in the same prime field \(\mathbb{Z}_M\) each, and the evidence given in [47] inclines people to believe that PRFP is harder than RFP, which makes people be interested in it whether or not there exists a polynomial time algorithm for solving MPP, ASPP, TLP, or PRFP in the quantum computational model [21].

At present, the REESSE1+ cryptosystem is constructed in a prime field \(\mathbb{Z}_M\), namely \(\mathbb{Z}(M)\).

Suppose that \(M\) is still a prime number. 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 [22]. Additionally, we suppose 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 \(\mathbb{Z}_M[x]/P(x)\) constitutes a polynomial ring including a congruent Abelian group. Therefore, it is feasible to transplant REESSE1+ to the ring \(\mathbb{Z}_M[x]/P(x)\) from the prime field \(\mathbb{Z}_M\).

From the dialectical viewpoint, it is impossible to see that a public key cryptosystem possesses all the merits because some merits are possibly restrained by 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, as viewed from utility, it should be researched further how to decrease the length of a REESSE1+ modulus and to increase the speed of a REESSE1+ decryption.

Acknowledgment

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

The authors also would like to thank the Professors Dingyi Pei, Jie Wang, Ronald L. Rivest, Moti Yung, Dingzhu Du, Mulan Liu, Huanguo Zhang, Dengguo Feng, Yixian Yang, Miaozi Xu, Qibin Zhai, Hanliang Xu, Xuejia Lai, Yongfei Han, Kefei Chen, Yupu Hu, Rongquan Feng, Ping Luo, Donghai Lin, Jinfeng Ma, Lei Hu, Lusheng Chen, Xiao Chen, Wenbao Han, Lequan Min, Bogang Lin, Xiaolin Zheng, Hong Zhu, Renji Tao, Bingru Yang, Zhuying Wang, Quanyuan Wu, and Zhichang Qi for their important counsel, suggestions, and corrections.

References

[1] R. L. Rivest, A. Shamir, and M. L. Adleman, A Method for Obtaining Digital Signatures and Public-key Cryptosystems, Communications of the ACM, vol. 21, no. 2, 1978, pp. 120-126.
[2] T. ElGamal, A Public-key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, IEEE Transactions on Information Theory, vol. 31, no. 4, 1985, pp. 469-472.
[3] W. Diffie and M. E. Hellman, New Directions in Cryptography, IEEE Transactions on Information Theory, vol. 22, no. 6, 1976, pp. 644-654.
[4] O. Goldreich, Foundations of Cryptography: Basic Tools, Cambridge University Press, Cambridge, UK, 2001.
[5] A. C. Yao, Theory and Applications of Trapdoor Functions, Proc. of 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/~sail/papers/enyce.pdf, Jul. 2004.
[7] M. Bellare and P. Rogaway, The Exact Security of Digital Signatures — How to Sign with RSA and Rabin, Proc. of 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, 1978, pp. 525-530.
[9] A. Shamir, A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem, Proc. of the 23th IEEE Symposium on the Foundations of Computer Science, IEEE, 1982, pp. 145-152.
[10] E. F. Brickell, Solving Low Density Knapsacks, Proc. of Advance in Cryptology: CRYPTO ’93, Plenum Press, 1984, pp. 25-37.
[11] M. J. Coster, A. Joux, B. A. LaMacchia etc., Improved Low-Density Subset Sum Algorithms, Computational Complexity, vol. 2, issue 2, 1992, pp. 111-128.
[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, 1994, pp. 181-199.
[13] I. F. Blake, G. Seroussi, and N. P. Smart, Elliptic Curves in Cryptography, Cambridge University Press, Cambridge, UK, 1999.
[14] S. Y. Yan, Number Theory for Computing, Proc. of the 23th IEEE Symposium on the Foundations of Computer Science, IEEE, 1982, pp. 145-152.
[15] J. Hoffstein, J. Pipher, and J. H. Silverman, NTRU: A Ring-based Public Key Cryptosystem, Proc. of 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, 2003, pp. 667-685.
[17] J. M. Chen and B. Y. Yang, A More Secure and Efficacious TTS Signature Scheme, Proc. of 6th Int. Conference on Information Security & Cryptology, Springer-Verlag, 2003, pp. 320-338.
[18] S. Su, The REESSE1 Public-key Cryptosystem, Computer Engineering & Science (in Chinese), vol. 25, no. 5, 2003, pp. 13-16.
[19] S. Liu, F. Zhang, and K. Chen, Cryptanalysis of REESSE1 Digital Signature Algorithm, Proc. of CCICS 2005, Science Press of China, 2005, pp. 200-205.
[20] D. Naccache and J. Stern, A new public key cryptosystem, Proc. of Advances in Cryptology: EUROCRYPT '97, Springer-Verlag, 1997, pp. 27-36.
[21] P. W. Shor, Polynomial-time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, SIAM J. on Computing, vol. 26, issue 5, 1997, pp.1484-1509.
[22] T. W. Hungerford, Algebra, Springer-Verlag, New York, 1998.
[23] K. H. Rosen, Elementary Number Theory and Its Applications (5th ed.), Addison-Wesley, Boston, 2005, ch. 12.
[24] M.J. Wiener, Cryptanalysis of Short RSA Secret Exponents, IEEE Transactions on Information Theory, v36(3), 1990, pp. 553-558.
[25] A. J. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, London, UK, 1997.
[26] P. Garrett, Making, Breaking Codes: An Introduction to Cryptology, Prentice-Hall, New Jersey, 2001.
[27] S. Su and S. Lü, To Solve the High Degree Congruence $x^a \equiv n \equiv a \pmod{p}$ in GF(p), Proc. of Int. Conference on Computational Intelligence and Security, IEEE press, 2007, pp. 672-676.
[28] D. Z. Du and K. Ko, Theory of Computational Complexity, John Wiley & Sons, New York, 2000.
[29] D. R. Stinson, Cryptography: Theory and Practice (2nd ed.), CRC Press, London, UK, 2002.
[30] M. Davis, The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions, Dover Publications, Mineola, 2004.
[31] H. Ritter, Breaking Knapsack Cryptosystems by Max-Norm Enumeration, Proc. of 1st International Conference of the Theory and Application of Cryptology, CTU Publishing House, 1996, pp. 480-492.
[32] V. Niemi, A New Trapdoor in Knapsacks, Proc. of 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, 2006, pp. 1-25.
[34] P. Q. Nguên and D. Stehle, Floating-Point LLL Revisited, Proc. of 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, Proc. of 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, Proc. of Advance in Cryptology: Crypto '98, Springer-Verlag, 1998, pp. 13-25.
[37] V. Shoup, OAEP Reconsidered, Proc. of Advance in Cryptology: Crypto '01, Springer-Verlag, 2001, pp. 239-259.
[38] M. Bellare and P. Rogaway, Random Oracles are Practical: A Paradigm for Designing Efficient Protocols, Proc. of the 1st ACM Conference on Computer and Communications Security, ACM, 1993, pp. 62-73.
[39] R. Canetti, O. Goldreich, and S. Halevi, The Random Oracle Methodology Revisited, Proc. of the 30th Annual ACM Symposium on Theory of Computing, New York: ACM Press, 1998, pp. 209-218.
[40] D. Pointcheval and J. Stern, Security Proofs for Signature Schemes, Proc. of Advance in Cryptology: Eurocrypt ’96, LNCS 1070, Springer-Verlag, Berlin, 1996, pp. 387-398.
[41] M. Bellare and G. Neven, Multi-Signatures in the Plain Public-Key Model and a General Forking Lemma, Proc. of the 13th ACM Conference on Computer and Communications Security, Alexandria, Virginia, 2006, pp. 390-399.
[42] A. Young and M. Yung, Malicious Cryptography: Exposing Cryptovirology, Wiley press, 2004, pp. 344.
[43] C. Schnorr, Efficient Identification and Signature for Smart Card, Proc. of Advance in Cryptology: Crypto ’89, LNCS 435, Springer-Verlag, Berlin, 1989, pp. 239–251.
[44] H. Cohen, A Course in Computational Algebraic Number Theory, Springer-Verlag, New York, 2000.
[45] N. A. Moldovyan, Digital Signature Scheme Based on a New Hard Problem, Computer Science Journal of Moldova, vol. 16, no. 2, 2008, pp. 163-182.
[46] R. R. Stoll, Set Theory and Logic, Dover Publications, Mineola, 1979.
[47] S. Su, S. Lü, and X. Fan, Asymptotic Granularity Reduction and Its Application, Theoretical Computer Science, vol. 412, issue 39, Sep. 2011, pp. 5374-5386.
[48] D. Coppersmith, Small Solutions to Polynomial Equations and Low Exponent RSA Vulnerabilities, Journal of Cryptology, vol. 10, no. 4, 1997, pp. 223-260.

Authors
Shenchui 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 was long engaged in design of algorithms and development of software. Since 2000, he has been indulged in designing the REESSE / JUNA cryptosystems and digital signers. He is currently with college of computers, Beijing University of Technology. His research area covers computational complexity, cryptographic algorithms, and digital identities.

Shuwang Lü received a bachelor degree in electronics from University of Science and Technology China. He 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, 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.

> http://arxiv.org/pdf/cs/0702046 (revised Dec 2012) <
Appendix A — Indeterminate Encryption

Below, a small example is given.

This example is only used to explain how to encrypt a plaintext and decrypt a ciphertext through the algorithms in Section 5.3.1, and thus the quantities h, S, α, β and the some constraints are not considered.

Let n = 6 and P = 19.

1) Generation of a key pair

1. Select d = 21, D = 95, and T = 143.
2. Randomly generate a coprime sequence \( \{A_i\} = \{17, 10, 13, 9, 19, 7\} \).
3. Find the prime M = 174594421 such that \( M > 19^6 \) and \((dD T) \mid (M) = 174594420\).
4. Pick \( W = 155629 \) and \( δ = 3761 \) making \( \text{gcd}(δ, M) = 1 \) and \( \overline{M} = dDT \).
5. Randomly produce pairwise distinct \( \{l(i)\} = \{7, 15, 5, 11, 13, 9\} \).
6. Compute \( \{C_i\} = \{116331875, 87811986, 61498911, 6213388, 8089113, 9766243\} \) by \( C_i ≡ (A_i, W^{l(i)}) \% M \).

Regard \( \{C_i, M\} \) as a public key, and \( \{A_i, W, δ\} \) as a private key. Discard the quantities \( d, D, \{l(i)\} \) which cannot be divulged.

2) Encryption

Assume that \( \{C_i, M\} \) is a public key. Let \( b_1 \ldots b_n = 100110 \) be a plaintext.

1. Obtain \( b_1 \ldots b_n = 100310, k = 1 \).
2. Randomly generate \( s_1 \ldots s_n = 010001 \).
3. Modify \( s_i \) and \( b_i \) by \( s_i \leftarrow 1 \) and \( b_i \leftarrow b_i + k = 2 \).
4. Compute \( \mathcal{G} = (C_1^2 \times 0 \times 1 \times 1 \times 1 \times 1 \times 1) \% (C_2^3 \times 1 \times 2 \times 0 \times 1 \times 1 \times 1) \% (C_3^3 \times 0 \times 1 \times 3 \times 0 \times 1 \times 1) \% (C_4^2 \times 0 \times 4 \times 3 \times 1 \times 0 \times 1) \% (C_5^2 \times 1 \times 5 \times 2 \times 1 \times 1 \times 1) \% (C_6^2 \times 2 \times 0 \times 0 \times 1 \times 1 \times 1) \).

Due to \( \mathcal{G} = 116331875 \times 156599315 \times 124996494 = 75924783 \) (% M) by \( \mathcal{G} = \Pi \{C_i (s_i + (j - b_i - 1) \% 1)\}^3 \) (% M) with \( C_{n+1} = 1 \).

So, the ciphertext \( \mathcal{G} = 75924783 \) is obtained. Notice that because \( s_1 \ldots s_n \) is randomly generated, on inputting the identical plaintext 100110 many times, we will obtain many distinct ciphertexts.

3) Decryption

Assume that \( \{17, 10, 13, 9, 19, 7\}, 155629, 3761 \) is a related private key.

Let \( \mathcal{G} = 75924783 \) be a ciphertext.
1. Compute \( \mathcal{G} = \mathcal{G}^2 \% M = 75924783 \% 156599315 \% 124996494 = 75924783 \) (% M) by \( \mathcal{G} = \mathcal{G} W^2 \% M \).
2. Set \( b_1 \ldots b_n = 0 \ldots 0, \mathcal{G} = 0, j = 0, k = 0, G = \mathcal{G} = 144398410, i = 1 \).

Inner loop 1:
5. Due to \( (A_2 \mathcal{G} = 10^1) \| (G = 144398410), \mathcal{G} = 0 + 1 = 1, \) goto 5.
6. Due to \( (A_2 \mathcal{G} = 10^2) \| (G = 144398410), \) the next.
7. Due to \( \mathcal{G} = \mathcal{G} + 1 \neq 0, \)
compute \( G = 144398410 / A_2^1 = 14439841; \)
obowing to \( k = 1 > 0, \) let \( b_2 = 1; \)
obowing to \( k = 1 \neq 0, \) let \( i = 2 + 1 = 3; \)
obowing \( k + 1 \geq 2 > \mathcal{G} = 1, \) let \( i = 6 + 1 = 7; \)
set \( \mathcal{G} = 0, k = 0, \)
8. Due to \( i = 7 > 6, \) the next.
9. Due to \( G = 14439841 \neq 1, \) goto 5.

Outer loop 2:
6. Due to \( G = 14439841 \neq 1, \) goto 5.

Outer loop 3:
5. Compute \( G = 131367179 \times 154229249 = 61370000 \) (% M) by \( \mathcal{G} = \mathcal{G} W^2 \% M. \)
6. Set \( b_1 \ldots b_n = 0 \ldots 0, \mathcal{G} = 0, j = 0, k = 0, G = \mathcal{G} = 61370000, i = 1 \).

Inner loop 1:
5. Due to \( (A_2 \mathcal{G} = 17^1) \| (G = 6137000), \mathcal{G} = 0 + 1 = 1, \) goto 5.
6. Due to \( (A_2 \mathcal{G} = 17^2) \| (G = 6137000), \) the next.
7. Due to \( \mathcal{G} = \mathcal{G} + 1 \neq 0, \)
compute \( G = 6137000 / A_2^1 = 361000; \)
obowing to \( k = 0, \) and \( i + \mathcal{G} – 1 < 6, \) let \( b_2 + \mathcal{G} = b_1 = 1; \)
obowing to \( k = 0, \) let \( i = 1 + \mathcal{G} = 1 + 1 = 2; \)
obowing \( k + 1 < 1 = \mathcal{G} = \mathcal{G}, \) the next;
set \( \mathcal{G} = 0, k = 0, \)
8. Due to \( i = 2 \leq 6 \) and \( G 
eq 1, \) goto 5.

Inner loop 2:
6. Due to \( (A_2 \mathcal{G} = 10^1) \| (G = 361000), \mathcal{G} = 0 + 1 = 1, \) goto 5.
7. Due to \( (A_2 \mathcal{G} = 10^2) \| (G = 361000), \) the next.
8. Due to \( \mathcal{G} = \mathcal{G} + 3 \neq 0, \)
compute \( G = 361000 / A_2^3 = 361; \)
obowing to \( k = 0, \) and \( i + \mathcal{G} – 1 < 6, \) let \( b_3 + \mathcal{G} = b_1 = 1; \)
obowing to \( k = 0, \) let \( i = 2 + \mathcal{G} = 2 + 3 = 5; \)
obowing \( k + 1 = 1 < \mathcal{G} = 3; \) the next;
set \( \mathcal{G} = 0, k = 0, \)
9. Due to \( i = 5 \leq 6 \) and \( G 
eq 1, \) goto 5.

Inner loop 3:
6. Due to \( (A_5 \mathcal{G} = 19^1) \| (G = 361), \mathcal{G} = 0 + 1 = 1, \) goto 5.
7. Due to \( (A_5 \mathcal{G} = 19^2) \| (G = 361), \mathcal{G} = 1 + 1 = 2, \) goto 5.
8. Due to \( (A_5 \mathcal{G} = 19^3) \| (G = 361), \) the next.
9. Due to \( \mathcal{G} = \mathcal{G} + 2 \neq 0, \)
compute \( G = 361000 / A_2^2 = 1; \)
obowing to \( i + \mathcal{G} – 1 = 5 + 2 – 1 = 6, \) let \( b_5 = b_5 = 1; \)
obowing to \( k = 0, \) let \( i = 5 + 2 + 1 = 7; \)
obowing \( k + 1 = 1 < \mathcal{G} = 2; \) the next;
10. Due to \( i = 7 > 6 \) or \( G = 1, \) the next.
11. Due to \( G = 1, \) end.

Appendix B — Computation of Density of a Knapsack from ASPP

1) Wrong Computation of Density in Section 5.2

It is known from Section 3.2 that a ciphertext is an ASPP $\mathcal{G} = \prod_{i=1}^{n} C_i^n \bmod M$.

Let $C_1 = g^{u_1}, \ldots, C_n = g^{u_n}, \mathcal{G} = g^{x} \bmod M$, where $g$ is a generator of $\langle \mathbb{Z}_M^{\ast} \rangle$ randomly selected.

Then, seeking $b_1, \ldots, b_n$ from $\mathcal{G}$ is equivalent to solving the congruence

$$u_1 b_1 + \ldots + u_n b_n = v \bmod M,$$  

(1)

where $\{u_1, \ldots, u_n\}$ is called a compact sequence (knapsack), $v \leq M$.

Note that $\mathbb{P}$ we stipulate that $b_1 \ldots b_n \neq 0$ contains at most $n/2$- 0-bits; $\mathbb{Q}$ if $g$ is different, $\{u_1, \ldots, u_n\}$ will be different for the same $\{C_1, \ldots, C_n\}$, and thus $\{u_1, \ldots, u_n\}$ has randomness.

When (1) will be reduced through the LLL lattice basis reduction algorithm, it should be converted into a non-modular form:

$$b_1 u_1 + \ldots + b_n u_n = v + kM,$$  

(2)

where $k \in [0, n]$ is an integer. To seek the original solution to (2), $k$ must traverse from 0 to $n$.

Let $D$ be the density of the compact sequence $\{u_1, \ldots, u_n\}$. We easily see that in Section 5.2, the formula $D \approx n^2 / \lg M$ is wrong, which is first pointed out by Xiangdong Fei (an associate professor from Nanjing University of Technology).

2) Right Computation of Density

Considering the structure of a lattice basis from (1) and the bit-length of a bit shadow $b_i$, the right computation of density of an ASPP knapsack should be

$$D = \sum_{i=1}^{n} \frac{\lg (n/2+1)}{\lg M} = n \frac{\lg (n/2+1)}{\lg M}$$

with $b_i \in [0, n/2+1]$ or

$$D = \sum_{i=1}^{n} \frac{\lg n}{\lg M} = n \frac{\lg n}{\lg M}$$

with $b_i \in [0, n]$.

Assume that $b_i \in [0, n]$. Concretely speaking, for $n = 80$ with $\lg M = 696, D = 80 \times 1/696 \approx 0.8046 < 1$; for $n = 96$ with $\lg M = 864, D = 96 \times 1/864 \approx 0.7778 < 1$; for $n = 112$ with $\lg M = 1030, D = 112 \times 1/1030 = 0.7612 < 1$; for $n = 128$ with $\lg M = 1216, D = 128 \times 1/1216 = 0.8421 < 1$.

These densities mean that the original solution to (1) may possibly be found through LLL lattice basis reduction (not certainly, and even with very low probability) because $D < 1$ only assure that the shortest vector is unique, but it cannot assure that the vector of the original solution to an ASPP is just the shortest vector in the reduced basis.

3) Bit Shadows Enhance Resistance of a Low Density ASSP Knapsack to Attacks

The LLL algorithm is to reduce a lattice basis (1, 0, ..., 0, $\vec{N}u_1), (0, 1, ..., 0, \vec{N}u_2), ..., (0, 0, ..., 1, \vec{N}u_n), (1/2, 1/2, ..., 1/2, \vec{N}(v + kM))$, where $\vec{N} > 1/2(n)^{1/2}$. No matter whether $(v + kM)$ is a classical subset sum or an anomalous subset sum, and whether the density is less than 1 or greater than 1, the LLL algorithm runs by its inherent rules. Lastly, the $n+1$ vectors $(\vec{e}_1, \ldots, \vec{e}_n, \vec{e}_{n+1})$’s which occur in the reduced basis are the first $n+1$ approximately shortest vectors, including the shortest vector, of which quite some satisfy $\vec{e}_1 u_1 + \ldots + \vec{e}_n u_n = v + kM$ (omitting the term $\vec{e}_{n+1} = 0$). If $D < 1$, the shortest vector is unique; and if $(v + kM)$ is a classical subset sum, the shortest vector is just the original solution.

We know from the above discussion that it has two necessary conditions to solve a SSP or ASPP through LLL lattice basis reduction: $\mathbb{R}$ the vector of the original solution is the shortest; $\mathbb{G}$ the shortest vector in the lattice is unique. $D < 1$ assures that the shortest vector is unique; and a classical subset sum assures that the vector of the original solution is just the shortest vector.

Return to (1). Even though the density of an ASSP knapsack is less than 1, the original solution is not necessarily found since $b_i \in [0, n/2+1]$ is a bit shadow (indicates that the original solution does not necessarily occur in the reduced basis), and there likely exist many solutions in the lattice.

For example, let $n = 4$ (short but without loss of generality), $M = 263$, $\{u_1, \ldots, u_4\} = \{48, 71, 257, 4\}$, and $v = 261$ ($u$ and $v$ are obtained through the discrete logarithms of $C_i$ and $\mathcal{G}$). Here, the density of $\{u_1, \ldots, u_4\}$ is $D = n \frac{\lg (n/2+1)}{\lg M} = 4 \times 2/9 = 0.8889 < 1$.

Assume that a plaintext $b_1 \ldots b_4 = 1100$, and its related bit shadow string $b_1 \ldots b_4 = 1300$ (thus $1 \times 48 + 3 \times 71 = 261$).

However, according to LLL lattice basis reduction, sought solution will be 0011 (notice, not the original) because there is $1 \times 257 + 1 \times 4 = 261$, and $(0, 0, 1, 1) \{1/2 + 1/2\}$ is the shortest vector in the lattice while $(1, 3, 0, 0) \{1^2 + 3^2\}$ is not the shortest, and even it will not occur in the reduced basis consisting of 5 vectors.

4) Density of an Optimized ASSP Knapsack

The modulus of prototypical REESSE1+ is relatively large, so in practice, it needs to be optimized (optimized REESSE1+ is called JUNA).

Return to Section 6. When $n = 80, 96, 112, 128$, correspondingly there is $\lg M = 384, 464, 544, 640$. The density of an optimized ASSP knapsack is $D = (3n/2)\frac{\lg (n/4+1)}{\lg M}$ with $b_i \in [0, n/4+1]$ (on the assumption that $B_1 \ldots B_{n/2}$ contains at most $n/4$ 00-pairs).

Concretely speaking,

for $n = 80$ with $\lg M = 384, D = 120 \times 5/384 \approx 1.5625 > 1$; for $n = 96$ with $\lg M = 464, D = 144 \times 5/464 \approx 1.5517 > 1$; for $n = 112$ with $\lg M = 544, D = 168 \times 5/544 \approx 1.5441 > 1$; for $n = 128$ with $\lg M = 640, D = 192 \times 6/640 \approx 1.8000 > 1$.

Under the circumstances, owing to $D > 1$ (indicates there will exist many solutions to the ASSP, and even the shortest vector is also nonunique), it is impossible to find the original plaintext $b_1 \ldots b_6$ through LLL lattice basis reduction. Therefore, at present there exists no subexponential time solution to the ASSP used in the optimized encryption scheme.

It should be noted that even if $80 \leq n \leq 128$, the LLL
lattice basis reduction algorithm right cannot find the original solution to SSP when \( D > 1 \), which is proven by our experiments.

**Appendix C — Offering a Reward**

It may be regarded as a type of proof by experiment.

Here, \( n \geq 80 \) is the length of a binary string \( b_1...b_n \neq 0 \) of which the bit-pair string is \( B_1...B_n/2 \) containing at most \( n/4 \) 00-pairs, the sign \( \% \) denotes 'modulo', \( \mathcal{M} \) means '\( M-1 \)' with \( M \) prime, and \( \lg x \) means the logarithm of \( x \) to the base 2.

The analysis in the paper shows that any effectual attack on REESSE1+ will be reduced to the solution of four intractabilities: a multivariate permutation problem (MPP), an anomalous subset product problem (ASPP), a transcendental logarithm problem (TLP), and a polynomial root finding problem (PRFP) so far.

It is well known that it is infeasible in subexponential time to find a large root to the PRFP \( ax^n + bx^{n-1} + cx + d = 0 \) (\% \( M \)) with \( a \notin \{0, 1\} \), \( |b| + |c| \neq 0 \), \( d \neq 0 \), and \( n, M \) large enough.

Let \( n = 80, 96, 112, 128 \) with \( \lceil \lg M \rceil = 384, 464, 544, 640 \) for the optimized REESSE1+ encryption scheme or with \( \lceil \lg M \rceil = 80, 96, 112, 128 \) for the lightweight REESSE1+ signing scheme.

Assume that \( \{C_1, ..., C_{3n/2}\}, M \) is a public key, and \( \{A_1, ..., A_{3n/2}\}, \{l(1), ..., l(3n/2)\}, W, \delta, \bar{M} \) with \( W, \delta \in (1, \mathcal{M}) \), \( A_i \in \{2, 3, ..., 1201\} \), and \( l(i) \in \{+/-5, +/-7, ..., +/-2(3n/2) + 3\} \) is a private key, where the sign +/- means that the plus sign or minus sign – is selected, and unknown to the masses.

The authors promise solemnly that

1. anyone who can extract the original private key definitely from the MPP

\[ C_i = (A_i W^{l(i)})^\delta \quad (\% M) \] for \( i = 1, ..., 3n/2 \)

in DLP subexponential time will be awarded \$100000 when \( n = 80, 96, 112, 128 \) with \( \lceil \lg M \rceil = 384, 464, 544, 640 \), or \$100000 with \( \lceil \lg M \rceil = 80, 96, 112, 128 \);

2. anyone who can recover the original plaintext \( b_1...b_n \) definitely from the ASPP

\[ \mathcal{G} = \prod_{i=1}^n (C_{i(l(i)-1)+i})^\delta \quad (\% M) \] with \( C_0 = 1 \) and \( B_i \) a bit-pair shadow

in DLP subexponential time will be awarded \$100000 when \( n = 80, 96, 112, 128 \) with \( \lceil \lg M \rceil = 384, 464, 544, 640 \), or \$100000 with \( \lceil \lg M \rceil = 80, 96, 112, 128 \), where \( B_i = 0 \) if \( B_i = 00 = 1 + the number of successive 00-pairs before \( B_i \) if \( B_i \neq 00 \), or \( 1 + the number of successive 00-pairs before \( B_i \) + the number of successive 00-pairs after the rightmost non-00-pair if \( B_i \) is the leftmost non-00-pair as \( b_1...b_{12} = 010000110100 = B_1...B_6 = 01 00 00 11 01 00 \) with \( B_1...B_6 = 2 0 0 3 1 0 \).

3. anyone who can find the original large answer \( x \in (1, \mathcal{M}) \) definitely to the TLP

\[ y = (g x)^\delta \quad (\% M) \] with known \( g, y \in (1, \mathcal{M}) \) in DLP subexponential time will be awarded \$100000 when \( n = 80, 96, 112, 128 \) with \( \lceil \lg M \rceil = 384, 464, 544, 640 \), or \$10000 with \( \lceil \lg M \rceil = 80, 96, 112, 128 \).

Of course, any solution must be described with a formal process, and can be verified with our examples. The time of solving a problem should be relevant to arithmetic steps, but irrelevant to CPU speeds.

The DLP subexponential time means the running time of an algorithm for solving the DLP in the prime field \( \mathbb{Z}/(M) \) through Index-calculus method at present, namely \( L_M[1/3, 1.923] \).

Note that the TLP is written as \( y = (gx)^\delta \quad (\% M) \) instead of \( y = x^\delta \quad (\% M) \) due to the asymptotic property of \( M \), and in the paper, some pieces of evidence found incline people to believe that the subset product problem (SPP) \( \mathcal{G}_i = \prod_{i=1}^n C_i^{\delta} \) (\% \( M \)) is harder than the DLP asymptotically, but due to \( \lceil \lg M \rceil \leq 640 \) and the density of a related knapsack being low, SPP can almost be solved in DLP subexponential time.