* The REESSE1+ Public-key Cryptosystem

—— *A Multiproblem Public-key Cryptosystem*

Shenghui Su [1, 2], and Shuwang Lü [3]

[1] College of Computer Science, Beijing University of Technology, Beijing 100022, P.R.China
[2] School of Info Engi, University of Science & Technology Beijing, Beijing 100083, P.R China
sheenway@126.com
[3] School of Graduate, Chinese Academy of Sciences, Beijing 100039, P.R.China
swlu@ustc.edu.cn

A Version before March 15 2007

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

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

1 Introduction

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

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

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

Obviously, REESSE1+ is a type of multiproblem cryptosystem.

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

* The newest version of this paper can be accessed at http://www.arxiv.org/pdf/cs/0702046.

The security of a multiproblem cryptosystem is equivalent to that hardness whose time complexity is smallest in all the involved hardnesses.

A multiproblem cryptosystem must be a multivariate one or multiparameter one because only multiple variables may bring in multiple hardnesses.

In this paper, unless otherwise specified, sign '%' means 'modulo', and $\langle a, b\rangle$ represents the greatest common divisor of two integers. Let $|x|$ denote the order of x % M, and '% –1' denote '% $(M-1)$'.

2 A Coprime Sequence and the Lever Function

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

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

Proof: By reduction to absurdity.

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

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

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

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

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

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

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

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

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

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

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

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

calculation on $\ell(.)$ at 'a private terminal'.

3 Design of the REESSE1+ Public Key Cryptosystem

3.1 The Key Generation Algorithm

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

Assume that S, T, $Đ$, $đ$ are pairwise coprime integers, where the binary form of S only contains two '1' bits respectively at the beginning and the end, $T \geq 2^n$, $Đ \geq 2^n$, and $đ$ is a non-large integer.

(1) Randomly generate a coprime sequence $\{A_1, \ldots, A_n\}$, and compute $G = \prod_{i=1}^{n} A_i$.

(2) Find a prime $M > G$ making $\langle S, M-1\rangle = 1$, $đĐT \mid (M-1)$,
and $q \mid (M-1)$ for any prime $q \in [1, n+4]$.

(3) Pick δ making $\langle \delta, M-1\rangle = 1$, and $|\delta| = đĐT$.

(4) Compute $\alpha \leftarrow \delta^{\delta^n}$, $W \leftarrow G^{-1}(\alpha\delta^{-1})^{1/S}$, $\beta \leftarrow \delta^{(\delta+1)WS}$, and $\gamma \leftarrow \delta^{W^n} \% M$.

(5) Produce pairwise distinct $\ell(1), \ldots, \ell(n) \in \Omega = \{i\,\delta(\%-1) \mid i = 5, \ldots, n+4\}$.

(6) Compute $\{C_1, \ldots, C_n \mid C_i \leftarrow A_i W^{\ell(i)} \% M$, for $i = 1, \ldots, n\}$.

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

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

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

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

3.2 The Encryption Algorithm

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

(1) Set $\hat{\boldsymbol{G}} \leftarrow 1$, $i \leftarrow 1$.

(2) If $b_i = 1$, $\hat{\boldsymbol{G}} \leftarrow \hat{\boldsymbol{G}} C_i \% M$.

(3) Let $i \leftarrow i + 1$. If $i \leq n$, go to (2), or else end.

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

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

3.3 The Decryption Algorithm

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

(1) Compute $\hat{\boldsymbol{G}} \leftarrow \hat{\boldsymbol{G}}(W^{-1})^{\delta} \% M$.

(2) Set $b_1 \ldots b_n \leftarrow 0$, $G \leftarrow \hat{\boldsymbol{G}}$, $i \leftarrow 1$.

(3) If $A_i \mid G$, set $b_i \leftarrow 1$ and $G \leftarrow G / A_i$.

(4) Let $i \leftarrow i + 1$. If $i \leq n$ and $G \neq 1$, go to (3).

(5) If $G \neq 1$, go to (1), or else end.

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

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

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

3.4 The Digital Signature Algorithm

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

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

(2) Set $k_1 \leftarrow \sum_{i=1}^{n} b_i \ell(i)$ % –1, $G_0 \leftarrow \prod_{i=1}^{n} A_i^{\neg b_i}$.

(3) Pick Q making $Đ \mid (\delta Q - WH)$, $đ \nmid ((\delta Q)^n - (WH)^n)$ (% –1).

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

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

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

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

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

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

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

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

Further, we have theorem 2.

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

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

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

3.5 The Identity Verification Algorithm

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

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

(2) Compute $\hat{\boldsymbol{G}} \leftarrow \prod_{i=1}^{n} C_i^{b_i}$ % M.

(3) Compute $X \leftarrow (\alpha Q^{-1})^{QUT} \alpha^{Q^n T}$ % M, $Y \leftarrow (\hat{\boldsymbol{G}}^{QT} U^{-1})^{US} \beta^{UHT} \gamma^{H^n T}$ % M.

(4) If $X \equiv Y$, the identity is valid and F intact,

otherwise the identity is invalid or F already modified.

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

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

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

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

$$\begin{aligned} Q^{QU} V^S &\equiv (R G_0)^{SQU} \delta^{QU} (R^{-1} W G_1)^{QUS} \delta^{\lambda S} \\ &\equiv (W G_0 G_1)^{QUS} \delta^{QU} \delta^{\lambda S} \\ &\equiv \alpha^{QU} \delta^{((\delta+1)SU + \Sigma(\delta Q)^{n-1-i}(WH)^i)(\delta Q - WH)} \\ &\equiv \alpha^{QU} \delta^{-(\delta+1)WHSU} \delta^{\delta(\delta+1)QUS} \delta^{(\delta Q)^n - (WH)^n} \\ &\equiv \alpha^{QU} \beta^{-UH} \delta^{\delta(\delta+1)QUS} \alpha^{Q^n} \gamma^{-H^n} \text{ (\% } M\text{)}. \end{aligned}$$

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

$$V^{ST} \equiv (\alpha Q^{-1})^{QUT} \alpha^{Q^n T} \beta^{-UHT} \gamma^{-H^n T} \delta^{\delta(\delta+1)QUST}$$

$$\equiv X\beta^{-UHT}\gamma^{-H^nT}\delta^{\delta(\delta+1)QUST}\ (\%\ M).$$

In addition,

$$\begin{aligned} U^U V^T &\equiv (R\,W^{k_1-1}\,\delta^{\delta(\delta+1)})^{QUT}(R^{-1}WG_1)^{QUT}\delta^{\lambda T} \\ &\equiv (W^{k_1}G_1)^{QUT}\delta^{\delta(\delta+1)QUT}\delta^{\lambda T} \\ &\equiv \hat{\boldsymbol{G}}^{QUT}\delta^{\delta(\delta+1)QUT}\delta^{k\,\text{đ}\text{Đ}T} \\ &\equiv \hat{\boldsymbol{G}}^{QUT}\delta^{\delta(\delta+1)QUT}\ (\%\ M). \end{aligned}$$

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

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

By the double congruence theorem, there is

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

Making transposition and counteraction gives $X \equiv (\hat{\boldsymbol{G}}^{QT}U^{-1})^{US}\beta^{UHT}\gamma^{H^nT} \equiv Y\ (\%\ M)$.

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

3.6 The Double Congruence Theorem

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

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

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

What follows is the proof of theorem 3.

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

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

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

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

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

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

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

The above two equations multiplying yields

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

Furthermore, we have

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

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

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

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

Then, it holds that

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

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

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

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

If neither s nor t is relatively prime to $p - 1$, let $k = \langle s, p - 1\rangle$, $l = \langle t, p - 1\rangle$. Then we see $\langle s / k, p - 1\rangle = 1$

and $\langle t / l, p-1\rangle = 1$. Thus, there are $(x_0 x_1^{-1})^k \equiv 1$ and $(x_0 x_1^{-1})^l \equiv 1$. It is known from $\langle s, t\rangle = 1$ that $\langle k, l\rangle = 1$. In terms of the group theory [10], when $\langle k, l\rangle = 1$, only the element '1' belongs to two different sub-group at same time. Therefore, $x_0 x_1^{-1} \equiv 1$, namely $x_1 = x_0$, and x_0 bears uniqueness.

To sum up, we prove theorem 3.

3.7 Characteristics of REESSE1+

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

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

3.8 Correctness of the Decryption Algorithm

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

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

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

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

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

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

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

The above process gives out a method for seeking G.

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

3.9 Uniqueness of a Plaintext Solution to a Ciphertext

Because $\{C_i\}$ is one non-coprime sequence, the mapping from the subsequence $\{C_{x_1}, \ldots, C_{x_m}\}$ to the

product $\hat{\boldsymbol{G}}$ is theoretically many-to-one. It might possibly result in the nonuniqueness of the plaintext solution $b_1 \ldots b_n$ when $\hat{\boldsymbol{G}}$ is being unveiled.

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

$$\hat{\boldsymbol{G}} \equiv C_{x_1} \ldots C_{x_m} \equiv C_{y_1} \ldots C_{y_h} \ (\% M).$$

Then,

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

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

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

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

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

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

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

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

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

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

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

4 Securities of Encryption and Decryption

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Further, we can argue that the time complexity of the continued fraction attack is $O(n!) > O(2^n)$, for which we have a curt explanation as follows:

Theorem 4 If α is an irrational number and if r / s is a rational number in lowest terms, where r and s are integers with $s > 0$ such that $\mathrm{abs}(\alpha - r / s) < 1 / (2s^2)$, then r / s is a convergent of the simple continued fraction expansion of α [11].

Reference [11] does not offer a similar theorem with α being a rational number. Therefore, when α is a rational number, theorem 4 does not necessarily hold.

Additionally, due to the indeterminacy of $\ell(i)$, which indicates there exists $W^{\ell(k)} \equiv W^{\ell(k)+\tau}$ (% M) or $W^{\ell(k)} \equiv A_m W^{\ell(h)}$ (% M), $\ell(i) + \ell(j) = \ell(k)$ can not be determined in polynomial time, that is to say, determining $\ell(i) + \ell(j) = \ell(k)$ is unmeaning.

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

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

In terms of sect. 3.2, the ciphertext is a modular subset product, namely $\hat{\boldsymbol{G}} \equiv \prod_{i=1}^n C_i^{b_i}$ (% M), where $b_1 \ldots b_n$ is a plaintext block or a symmetric key, and $\{C_1, \ldots, C_n\}$ is a public key.

Obviously, $\prod_{i=1}^n C_i^{b_i} = LM + \hat{\boldsymbol{G}}$. Due to $L \in [1, M - 1]$, deriving $\prod_{i=1}^n C_i^{b_i}$ from $\hat{\boldsymbol{G}}$ is infeasible, which indicates inferring $b_1 \ldots b_n$ from $\hat{\boldsymbol{G}}$ is not a factorization problem.

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

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

where i is from 1 to n.

Because we need not only to figure out the value of $\sum_{i=1}^n b_i$ but also to find out the position of every $b_i =$

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

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

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

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

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

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

4.3 Avoiding the Adaptive-chosen-ciphertext Attack

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

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

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

5 Securities of Digital Signature and Identity Verification

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

Assume that p is a prime, and $k \mid (p-1)$. In terms of the probabilistic algorithm in section 1.6 of reference [12], the time complexity of finding out a random solution to $x^k \equiv c \ (\% p)$ is at least $\max(O(2^{k-1}), O(p/k))$. Thus, when $k > 80$ or $p/k > 2^{80}$, this algorithm is ineffectual currently.

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

It is known from the digital signature algorithm that

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

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

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

For the first equation, because δ is unknown, and the right of the equation is not a constant, it is impossible to solve the equation for $R G_0$. If δ is guessed, the probability of hitting δ is $1/|\delta| < 1/2^n$.

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

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

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

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

$$(\alpha Q^{-1})^{QUT}\alpha^{Q^nT} \equiv (\hat{G}^{QT}U^{-1})^{US}\beta^{UHT}\gamma^{H^nT} (\% M)$$

holds.

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

5.2.1 The Super Logarithm Problem

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

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

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

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

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

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

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

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

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

We consider two functions over the real set $\mathbb{R}$: $y = f(x) = g^x$, and $y = \varphi(x) = x^x$, where $x > 0$.

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

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

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

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

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

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

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

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

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

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

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

At present there is no better method for seeking a super logarithm than the exhaustive search, and thus

the time complexity of the solution to $x^x \equiv c\ (\%\ p)$ may be expected to be $O(p) > O(2^n)$, where n is the length of a message digest.

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

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

$$(\alpha Q^{-1})^{QUT}\alpha^{Q^nT} \equiv (\hat{\boldsymbol{G}}^{QT}U^{-1})^{US}\beta^{UHT}\gamma^{H^nT}\ (\%\ M),$$

an attacker may suppose the value of a variable.

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

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

If the attacker hits the small $đ$, obtains $Đ$ by factorizing $M-1$, and raises either side of the discriminant to the power of $đ$, then when $Đ \mid (\delta Q - WH)$, there is

$$(\alpha Q^{-1})^{đQUT} \equiv (\hat{\boldsymbol{G}}^{QT}U^{-1})^{đUS}\beta^{đUHT}\ (\%\ M).$$

Further,

$$(\alpha Q^{-1})^{đQT} \equiv (\hat{\boldsymbol{G}}^{QT}U^{-1})^{đS}\beta^{đHT}\ (\%\ M).$$

Hereat, if suppose the value of Q which is unknown, U may possibly be worked out in polynomial time. However, Q and U must satisfy both the discriminant and the condition $Đ \mid (\delta Q - WH)$, which is impossible since both δ and W are unknown.

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

Due to $Q \equiv (RG_0)^S\delta$, $U \equiv (RW^{k_1-1}\delta^{\delta(\delta+1)})^{QT}$ and $V \equiv (R^{-1}WG_1)^{QU}\delta^{\lambda}\ (\%\ M)$, an attacker may attempt the following attack method.

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

Let $Q \equiv a^S\delta$, $U \equiv b^{QT}\delta^{\delta(\delta+1)QT}$, and $V \equiv c^{QU}\delta^{\lambda}\ (\%\ M)$, where λ meets

$$\lambda S \equiv ((\delta+1)SU + \textstyle\sum_{i=0}^{n-1}(\delta Q)^{n-1-i}(HW)^i)(\delta Q - WH)\ (\%\ -1),$$

and Q, U make $Đ \mid (\delta Q - WH)$ as well as $đ \mid ((\delta+1)SU + \sum_{i=0}^{n-1}(\delta Q)^{n-1-i}(HW)^i)\ (\%\ -1)$. If the attacker seeks δ and W such taht

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

then the attack is likely successful.

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

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

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

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

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

Accordingly,

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

If $\langle q, M-1\rangle \mid v$, and the trivial solution to $x^n \equiv (q^{-1}v)\delta^n\ (\%\ -1)$ exists, then $W \equiv (q^{-1}v)^{n^{-1}}\delta\ (\%\ -1)$, where computing n^{-1} is referred to theorem 1, theorem 2 and reference [8]. Further, due to $W = 0$ is not a solution to the equations, there is

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

Transparently, the above equation is a true polynomial of degree $n-1$, which can not be solved via discrete logarithms. If Euler phi function $\varphi(M-1)$ is worked out, the attacker may seem to resort to the probabilistic algorithm in section 1.6 of reference [12]. However, the running time of this algorithm is at least $\max(O(2^{n-2}), O(\varphi(M-1)/(n-1)))$. When $n > 80$ or $\varphi(M-1)/(n-1) > 2^{80}$, seeking δ is infeasible at present. Furthermore, when $\langle u(q^{-1}v)^{(n-1)n^{-1}}, M-1\rangle \nmid v$, doing division of polynomials is infeasible.

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

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

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

it is a signature fraud on F'.

The values of Q and U are allowed to be utilized separately.

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

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

If the two signatures (Q_1, U_1) and (Q_2, U_2) on the files F_1 and F_2 are obtained, due to $Đ \mid (\delta Q_1 - WH_1)$ and $Đ \mid (\delta Q_2 - WH_2)$, we see that $Đ \mid (\delta(Q_1 + Q_2) - W(H_1 + H_2))$. Let $Q' = Q_1 + Q_2$, $H' = H_1 + H_2$, then $Đ \mid (\delta Q' - WH')$. However, inferring F' from H' is intractable in terms of the properties of hash functions.

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

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

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

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

It is understood from the discriminant $(\alpha Q^{-1})^{QUT}\alpha^{Q^nT} \equiv (\hat{\boldsymbol{G}}^{QT}U^{-1})^{US}\beta^{UHT}\gamma^{H^nT} (\% M)$ that

$$\hat{\boldsymbol{G}}^{QUST}\beta^{UHT}\gamma^{H^nT} \equiv (\alpha Q^{-1})^{QUT}\alpha^{Q^nT}U^{US} (\% M).$$

Assume that Q and U are known and $\hat{\boldsymbol{G}} = f(H)$. Clearly, it is harder than the DLP to seek H according to the above congruence. Also it is harder than the DLP to infer H from $\hat{\boldsymbol{G}}$ (see sect. 4.2).

6 Conclusions

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

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

The basic idea of optimization is to make use of binary compact sequences. That is, the mapping is not regarded as one to one but as three items to two bits between a non-coprime sequence and a symmetric key. (or a plaintext block). The n-bit symmetric key is partitioned evenly into $n/2$ units in order, and every unit has 2 bits and 4 combinations, namely 00, 01, 10, and 11. Let 00 map to 1, and the others do respectively to the three consecutive items of $\{C_i\}$. After optimized, the bit-length of M will decrease to 384 when $n = 80$, and 544 when $n = 112$.

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

REESSE1+ cryptosystem.

Acknowledgment

The authors would like to thank the Academicians Jiren Cai, Changxiang Shen, and Zhongyi Zhou for their important guidance, suggestions, and helps.

The authors would like to thank Mulan Liu, Huanguo Zhang, Dingyi Pei, Dengguo Feng, Xuejia Lai, Lequan Min, Zhiying Wang, and Maozhi Xu for their important advice, suggestions, and corrections.

References

[1] R. L. Rivest, A. Shamir and L. M. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," *Communications of the ACM*, 21(2), 1978, pp.120-126.

[2] T. ElGamal, "A Public-key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, v. 31, n. 4, 1985, pp. 469-472.

[3] A. J. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, London: CRC Press, 1997, ch. 2, 8.

[4] Oded Goldreich, *Foundations of Cryptography: Basic Tools*, Cambridge, UK: Cambridge University Press, 2001, pp. 31–35.

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

[6] R. C. Merkle and M. E. Hellman, "Hiding information and Signatures in Trapdoor Knapsacks," *IEEE Transactions on Information Theory*, v. 24, n. 5, 1978, pp. 525-530.

[7] A. Shamir, "A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem," *Proceedings of the 23th IEEE Symposium on the Foundations of Computer Science*, 1982, pp. 145-152.

[8] Song Y. Yan, *Number Theory for Computing*, 2nd ed., Berlin: Springer-Verlag, 2002, ch. 1.

[9] Paul Garrett, *Making, Breaking Codes: An Introduction to Cryptology*, New Jersey: Prentice-Hall, 2001, ch. 12.

[10] Thomas W. Hungerford, *Algebra*, New York: Springer-Verlag, 1998, ch. 1–3.

[11] Kenneth H. Rosen, *Elementary Number Theory and Its Applications* (5th ed.), Boston: Addison-Wesley, 2005, ch. 12.

[12] Henri Cohen, *A Course in Computational Algebraic Number Theory*, Berlin: Springer-Verlag, 2000, ch. 1, 3.

* Refuting the Pseudo Attack on the REESSE1+ Cryptosystem

Shenghui Su [1, 2], and Shuwang Lü [3]

[1] College of Computer Science, Beijing University of Technology, Beijing 100022, P.R.China
[2] School of Info Engi, University of Science & Technology Beijing, Beijing 100083, P.R China
sheenway@126.com
[3] School of Graduate, Chinese Academy of Sciences, Beijing 100039, P.R.China
swlu@ustc.edu.cn

Abstract: We illustrate through example 1 and 2 that the condition at theorem 1 in [8] dissatisfies necessity, and the converse proposition of *fact* 1.1 in [8] does not hold, namely the condition $Z/M - L/A_k < 1/(2A_k^2)$ is not sufficient for $f(i)+f(j)=f(k)$. Illuminate through an analysis and ex.3 that there is a logic error during deduction of *fact* 1.2, which causes each of *fact* 1.2, 1.3, 4 to be invalid. Demonstrate through ex.4 and 5 that each or the combination of $q_{u+1} > q_u\Delta$ at *fact* 4 and *table* 1 at *fact* 2.2 is not sufficient for $f(i)+f(j)=f(k)$, *property* 1, 2, 3, 4, 5 each are invalid, and *alg*.1 based on *fact* 4 and *alg*.2 based on *table* 1 are disordered and wrong logically. Further, manifest through a repeated experiment and ex.5 that the data at *table* 2 is falsified, and the example in [8] is woven elaborately. We explain why $C_x \equiv A_x W^{f(x)} \ (\% M)$ is changed to $C_x \equiv (A_x W^{f(x)})^{\delta} \ (\% M)$ in REESSE1+ v2.1. To the signature fraud, we point out that [8] misunderstands the existence of T^{-1} and $Q^{-1} \% (M-1)$, and forging of Q can be easily avoided through moving H. Therefore, the conclusion of [8] that REESSE1+ is not secure at all (which connotes that [8] can extract a related private key from any public key in REESSE1+) is fully incorrect, and as long as the parameter Ω is fitly selected, REESSE1+ with $C_x \equiv A_x W^{f(x)} \ (\% M)$ is secure.

Keywords: Public key cryptosystem; Security; Lever function; Continued fraction; Sufficient condition

1 Introduction

In April 2001, we put forward the REESSE1 public-key encryption scheme [1]. In September 2003, we proposed the REESSE1 public-key cryptosystem which is an extension of the first version, and includes both encryption and signature [2]. In May 2005, it was argued that the lever function $\ell(.)$ is necessary and sufficient for the security of the REESSE1 encryption [3]. In [3], the continued fraction method of analyzing the key transforms $C_x \equiv A_x W$ and $C_x \equiv A_x W^{\ell(x)} \ (\% M)$ with $x \in [1, n]$ and $\ell(x) \in \Omega$ was mentioned earlier than in any other publications. In November 2006, an abbreviation of the REESSE1+ cryptosystem was submitted to eprint.iacr.org [4].

As is pointed out in [4], the set $\Omega = \{5\delta, \ldots, (n+4)\delta \mid \delta \geq 1\}$ is not unique, and other Ω may be selected — $\Omega = \{n+1, \ldots, n+n\}$ with $\ell(i) + \ell(j) \neq \ell(k) \ \forall \ i, j, k \in [1, n]$ for example. Clearly, Ω is a security dominant parameter, and just like p and q in the RSA cryptosystem.

In May 2005, [5] pointed out that the REESSE1 signature scheme was insecure, which is right.

In July 2005, [6] thought unreasoningly that the REESSE1 encryption scheme was insecure, which is wrong, and rebutted thoroughly by us in [7]. Moreover, [7] illuminated definitely that the idea of the continued fraction analysis of REESSE1 did not originate from [6] (naturally also not from [8]), ant the idea firstly formally appeared in our 2004 application for a national fund project [7]. What needs to be pointed out further is that the authors of [8] are the related reviewers of our 2004 application.

In December 2006, [8] thought unreasoningly again that the REESSE1+ public-key cryptosystem is not secure at all, which connotes any private key in REESSE1+ can be extracted by [8]. It is of flubdub and gulf.

The ancients said 'stop an advancing army with troops, and stop onrushing water with earth'.

In what follows, the function f in [8] is namely the function ℓ in [4], namely $f(i)$, $f(j)$, $f(k)$ in [8] are equivalent to $\ell(i)$, $\ell(j)$, $\ell(k)$, unless otherwise specified, the sign $\overline{M}$ represents '$M-1$', the sign % does 'modulo', and unattached (x) does x-th expression.

---

* This appendix can be accessed separately at http://arxiv.org/abs/0704.0492.
Received 05 April 2007, and revised 19 December 2009.

In short, there exist 6 grave faults in [8]:

① The converse proposition of *fact* 1.1 does not hold.

Clearly, *fact* 1.1 implies that if $f(i) + f(j) = f(k)$, then $Z / M - p_u / q_u < 1 / (2 q_u^2)$ with $L / A_k = p_u / q_u$.

We will prove by a counterexample that the former is only sufficient, but not necessary, namely if $Z / M - p_u / q_u < 1 / (2 q_u^2)$, then $f(i) + f(j) = f(k)$ do not necessarily hold, and also namely $Z / M - p_u / q_u < 1 / (2 q_u^2)$ for $f(i) + f(j) = f(k)$ is only necessary, but not sufficient.

② *Fact* 1.2, 1.3 and 4 do not always hold.

Even if they hold, *fact* 1.2, 1.3 and 4 each are insufficient for $f(i) + f(j) = f(k)$, and further, *property* 1, 2, 3, 4 and 5 are invalid. Note *fact* 4 is essentially equivalent to each of *fact* 1.2 and 1.3.

③ The converse proposition of *fact* 2.2 does not hold, namely *table* 1 is insufficient for $f(i) + f(j) = f(k)$.

④ Both *algorithm* 1 based on *fact* 4 and *algorithm* 2 based on *table* 1 are disordered & wrong logically.

⑤ To achieve so-called "breaking", the *example* in [8] was woven elaborately, and *table* 2 was falsified, namely its authors intendedly mutilated the two tuple data to cause indeterminacy.

⑥ The inverse $T^{-1} \% \bar{M}$ does not exist, and $Q^{-1} \% \bar{M}$ not necessarily exist.

Additionally, the case of $\Omega = \{5 + \delta, \ldots, (n + 4) + \delta \mid \delta \geq n - 4\}$ with $f(i) + f(j) \neq f(k)\ \forall\ i, j, k \in [1, n]$ is not analyzed at all.

Therefore, the cryptanalysis of the REESSE1+ cryptosystem by [8] is a type of pseudo-attack and balderdash leading to which the most radical reason is that the authors of [8] are not aware of the indeterminacy of the lever function $\ell(.)$ namely $f(.)$, as is mentioned in [4]:

① if the order of W is $d < \bar{M}$, then there is $W^{f(x)} \equiv W^{f(x)+d} \ (\% M)$, and when $f(i) + f(j) = f(k)$, we see that $f(i) + d + f(j) + d \neq f(k) + d$;

② when $f(i) + f(j) \neq f(k)$, there exist $C_i \equiv A_i' W'^{f'(i)}$, $C_j \equiv A_j' W'^{f'(j)}$, and $C_k \equiv A_k' W'^{f'(k)} \ (\% M)$ such that $f'(i) + f'(j) \equiv f'(k) \ (\% \bar{M})$ with $A_k' \leq \acute{p}$, where $\acute{p}$ is the maximal prime allowed.

Another vital reason is that [8] always regarded necessary conditions for $f(i) + f(j) = f(k)$ as sufficient and necessary conditions, and [8] did not consider the whole space of private keys or public keys.

2 Theorem 1 vs the REESSE1+ Cryptosystem

2.1 Condition at Theorem 1 in [8] Dissatisfies Necessity

Theorem 1 in [8] is retailed as follows:

Theorem 1: Let α be a real number, and let r / s be a rational with $\gcd(r, s) = 1$ and $|\alpha - r / s| < 1 / (2s^2)$. Then r / s is a convergent of the continued fraction expansion of α. ※

Here, $|\alpha - r / s|$ represents the absolute value of $(\alpha - r / s)$.

The proof of theorem 1 is referred to [9].

The condition $|\alpha - r / s| < 1 / (2s^2)$ is only sufficient for r / s to be a convergent of the continued fraction of α, but not necessary. Namely if r / s is a convergent of the continued fraction of α, $|\alpha - r / s| < 1 / (2s^2)$ does not necessarily hold.

An example is taken.

Example 1.

Let $r / s = 2 / 13$, and then $1 / 2s^2 = 1 / (2 \times 13^2) = 0.002958579882$.

Let $\alpha = 2039 / 13001$, and then

$$2039 / 13001 - 2 / 13 = 0.002987935839$$
$$> 0.002958579882 = 1 / (2 \times 13^2).$$

On the other hand, the continued fraction of $2039 / 13001$ is $1 / (6 + (1 / (2 + 1 / (1 + \ldots\ 1 / 3))))$.

Thus, $2 / 13$ is a convergent of the continued fraction of $2039 / 13001$, which illustrates $|\alpha - r / s| < 1 / (2s^2)$ is not necessary for r / s to be a convergent of the continued fraction of α.

2.2 A_k Will Emerge But Is Undecidable If $f(i) + f(j) = f(k)$

Assume that $\acute{p}$ is the maximum prime in the cryptosystem, $\{A_1, \ldots, A_n\}$ is a coprime sequence with $0 < \forall A_x \le \acute{p}$, $M > \prod_{x=1}^{n} A_x$ is a prime, and $C_x \equiv A_x W^{f(x)} (\% M)$ for $x = 1, \ldots, n$ is a public key [4], where $n \ge 6$, and $f(x) \in \Omega = \{5\delta, \ldots, (n+4)\delta \mid \delta = 1\} = \{5, \ldots, n+4\}$.

Assume $f(k) = f(i) + f(j)$ with $i \ne k, j \ne k$, and $i, j, k \in [1, \ldots, n]$. Let

$$Z \equiv C_i C_j C_k^{-1} (\% M).$$

Then

$$Z \equiv A_i A_j (A_k)^{-1} (\% M)$$

$$Z (A_k) \equiv A_i A_j (\% M)$$

$$Z (A_k) - L M = A_i A_j,$$

where L is a positive integer.

Dividing the either side of the above equation by $(M A_k)$ yields

$$Z / M - L / A_k = A_i A_j / (M A_k). \tag{1}$$

Due to $M > \prod_{x=1}^{n} A_x$ and every $A_x \ge 2$, we have

$$Z / M - L / A_k < 1 / (2^{n-2-1} A_k^2). \tag{1'}$$

Obviously, when $n > 2 + 1$, (1′) may have a variant, namely

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

In terms of theorem 1, L / A_k is a convergent of the continued fraction of Z / M.

Let $p_0 / q_0, p_1 / q_1, \ldots, p_t / q_t$ be the convergent sequence of continued fraction of Z / M, and

$$L / A_k = p_u / q_u.$$

Note that if p_u / q_u satisfies (1″), then $p_{u+1} / q_{u+1}, p_{u+2} / q_{u+2}, \ldots, p_t / q_t$ also likely satisfies (1″). Therefore, there likely exist multiple values of L / A_k by (1″), and A_k is undetermined.

However, if we do not know in advance whether $f(i) + f(j) = f(k)$, then even if $Z / M - p_u / q_u < 1 / (2 q_u^2)$, we can not decide $f(i) + f(j) = f(k)$. Namely $Z / M - p_u / q_u < 1 / (2q_u^2)$ is only necessary for $f(i) + f(j) = f(k)$, but not sufficient, which will be discussed further in what follows.

3 Conditions at *Fact* 1.1 and 4 Each Are Insufficient for $f(i) + f(j) = f(k)$

Because *fact* 4 is essentially equivalent to each of *fact* 1.2 and 1.3, if the condition at *fact* 4 is insufficient for $f(i) + f(j) = f(k)$, the conditions at *fact* 1.2 and 1.3 each are also insufficient.

The condition $q_{u+1} > q_u \Delta = q_u (M / (2\prod_{x=n-2}^{m} prime\langle x \rangle))^{1/2}$ at *fact* 4 connotes (1″) at *fact* 1.1 because (1″) is the precondition of $q_{u+1} > q_u \Delta$ which is the dominant basis of *alg*.1 [8].

3.1 Converse Proposition of *Fact* 1.1 does not Hold and (1″) Is Insufficient for $f(i) + f(j) = f(k)$

Fact 1.1 in [8] is retailed as follows:

***Fact* 1.1** [8]: If $f(i) + f(j) = f(k)$, there exists a q_u such that $q_u = A_k$ in $\{p_0 / q_0, p_1 / q_1, \ldots, p_t / q_t\}$, the convergent sequence of continued fraction expansion of Z / M with $Z \equiv C_i C_j C_k^{-1} \% M$. ※

Due to $f(i) + f(j) = f(k)$, $Z / M = L / A_k + A_i A_j / (M A_k)$, $M > \prod_{x=1}^{n} A_x$ and $A_x \ge 2$, we have

$$Z / M - L / A_k = A_i A_j / (M A_k) < A_i A_j / (A_k \prod_{x=1}^{n} A_x).$$

Further,

$$Z / M - L / A_k < 1 / (2A_k^2). \tag{1″}$$

Let $Z / M = [0; a_1, a_2, \ldots, a_t]$ is the continued fraction expansion of Z / M.

By theorem 1, $\exists\ u \in [1, t]$ makes $Z / M - p_u / q_u < 1 / (2\ q_u^2)$.

Let $L / A_k = p_u / q_u$, where

$$p_u / q_u = a_0 + 1 / (a_1 + 1 / (a_2 + \ldots + 1 / (a_{u-1} + 1 / a_u))). \tag{2}$$

Notice that it is possible that $\exists\ h > 0$ makes $Z / M - p_{u+h} / q_{u+h} < 1 / (2\ q_{u+h}^2)$, and moreover not *fact* 1.1 but its converse is the inner logical base of *alg*.1 in [8].

Through a counterexample, we will prove that the converse proposition of *fact* 1.1 does not hold, that is, the condition $Z / M - p_u / q_u < 1 / (2\ q_u^2)$ is insufficient for $f(i) + f(j) = f(k)$.

Example 2.

For convenience in computing, let $n = 6$, $\{A_x\} = \{11, 10, 3, 7, 17, 13\}$, $\delta = 1$, and $M = 510931$.

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

From $C_x \equiv A_x W^{f(x)}$ (% M), we obtain

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

and its inverse sequence

$\{C_x^{-1}\} = \{266775, 236469, 435654, 149312, 434038, 425203\}$.

Randomly select $i = 1$, $j = 3$, and $k = 5$. In this case, $f(5) = 7 \neq f(1) + f(3) = 9 + 10$. Compute

$$\begin{aligned} Z &\equiv C_1\ C_3\ C_5^{-1} \\ &\equiv 113101 \times 175066 \times 434038 \\ &\equiv 186640\ (\%\ 510931). \end{aligned}$$

Presume that W in $C_1\ C_3$ is just neutralized by W^{-1} in C_5^{-1}, then

$$186640 \equiv A_1\ A_3\ A_5^{-1}\ (\%\ 510931).$$

According to (1),

$$186640 / 510931 - L / A_5 = A_1\ A_3 / (510931\ A_5).$$

By the Euclidean algorithm, $a_1, a_2, a_3, \ldots$ are found out, and thus the continued fraction of

$$186640 / 510931 = 1 / (2 + 1 / (1 + 1 / (2 + 1 / (1 + 1 / (4 + \ldots + 1 / 3))))).$$

Heuristically let

$$p_4 / q_4 = L / A_5 = 1 / (2 + 1 / (1 + 1 / (2 + 1 / 1))) = 4 / 11,$$

which indicates that probably $A_5 = 11$. On this occasion, there is

$$\begin{aligned} 186640 / 510931 - 4 / 11 &= 0.0016575801 \\ &< 1 / (2\ A_5^2) = 1 / (2 \times 11^2) = 0.0041322314. \end{aligned}$$

The above expression satisfies (1″), namely the condition at theorem 1, and thereby $A_5 = 11$ less than the maximum in $\{A_x\}$ is deduced, which is in direct contradiction to factual $A_5 = 17$.

So the condition $Z / M - p_u / q_u < 1 / (2\ q_u^2)$ is not sufficient for $f(i) + f(j) = f(k)$, namely the converse proposition of *fact* 1.1 does not hold.

3.2 Each of *Fact* 1.2, 1.3 and 4 does not Hold

Fact 1.2 in [8] is retailed as follows:

***Fact* 1.2** [8]: There is sharp increase from q_u to q_{u+1} since $q_{u+1} \geq (A_k\ M / (2A_i\ A_j))^{1/2}$.

The derivation of *fact* 1.2 in [8] is retailed as follows:

Let L / A_k be the u-th convergent, i.e., $q_u = A_k$ and $p_u = L$, i.e., $p_u / q_u = L / A_k$. Then we know that

$$|Z / M - p_{u+1} / q_{u+1}| < A_i\ A_j / (A_k\ M) = 1 / (2\ ((A_k\ M / (2A_i\ A_j))^{1/2})^2). \tag{2′}$$

According to theorem 1 and convergence of sequence $\{p_0 / q_0, p_1 / q_1, \ldots, p_t / q_t\}$, we obtain that

$$q_{u+1} \geq (A_k\ M / (2A_i\ A_j))^{1/2} = A_k\ (M / (2A_i\ A_j\ A_k))^{1/2}. \tag{3}$$

※

Is the above derivation right? See the following analysis.

Clearly, by the definition of a finite continued fraction, (2′) holds. In addition, in terms of [9], p_{u+1} are q_{u+1} are coprime, and there is $q_{u+1} \geq A_k = q_u$, which is a judgment foundation.

If $f(i) + f(j) = f(k)$, then there is $|Z / M - p_u / q_u| < 1 / (2\ q_u^2)$ with $L / A_k = p_u / q_u$. Furthermore, through practical observations, in most cases, there is also

$$|Z / M - p_{u+1} / q_{u+1}| < 1 / (2\ q_{u+1}^2). \tag{3′}$$

According to (2′) and (3′), we have either

$$|Z / M - p_{u+1} / q_{u+1}| < 1 / (2\ q_{u+1}^2) < 1 / (2\ ((A_k\ M / (2A_i\ A_j))^{1/2})^2), \tag{3″}$$

or

$$|Z / M - p_{u+1} / q_{u+1}| < 1 / (2\ ((A_k\ M / (2A_i\ A_j))^{1/2})^2) < 1 / (2\ q_{u+1}^2). \tag{3‴}$$

If (3″) holds, there exists $q_{u+1} \geq A_k\ (M / (2A_i\ A_j\ A_k))^{1/2}$, which also indicates $q_{u+1} \geq A_k = q_u$.

If (3‴) holds, there exists $A_k\ (M / (2A_i\ A_j\ A_k))^{1/2} \geq q_{u+1}$. Notice that in this case, $q_{u+1} \geq A_k = q_u$ is still possible.

Therefore, $q_{u+1} \geq A_k (M / (2A_i\ A_j\ A_k))^{1/2}$, namely *fact* 1.2 does not necessarily hold, which indicates that there is a logic error during the derivation of (3) in [8].

Moreover, from (2′) and (3′) we can judge that when n is large enough — 80 for example, the probability that (3‴) holds is greater than one that (3″) holds.

Now, we review *fact* 1.3 in [8]. It is retailed as follows:

***Fact* 1.3** [8]: Due to *fact* 1.2, there is also a sharp increase from a_u to a_{u+1}, since $q_{v+1} = a_{v+1}\ q_v + q_{v-1}$ for v = 1, 3, …, t. Here a_v s are items of Z / M determined by (2). ※

Obviously, because *fact* 1.2 does not hold, *fact* 1.3 does not also hold.

Further, because *fact* 1.2, namely (3) does not hold, naturally, *fact* 4 in [8] does not also hold, that is, $q_{u+1} > q_u\ (M / (2\prod_{x=n-2}^{m}\ prime\langle x\rangle))^{1/2}$ is not always valid.

Observe an example once more.

In example 3, suppose that the bit-length of a plaintext block is 8, and two bits of a block correspond to three items of a coprime sequence $\{A_x\}$, which means that the encryption algorithm is optimized through a compact binary sequence. In practice, we do just so.

Apparently, the length of $\{A_x\}$ is $3 \times (8 / 2) = 12$.

Example 3.

Let $\{A_x\}$ = {{23, 11, 17}, {41, 29, 26}, {15, 19, 37}, {31, 7, 43}}, and

M = 2022169 > 31 × 37 × 41 × 43 = 2022161.

Randomly select W = 1507351, $f(1) = 6, f(2) = 14, f(3) = 9, f(4) = 11, f(5) = 12, f(6) = 10, f(7) = 8, f(8) = 16, f(9) = 5, f(10) = 13, f(11) = 15$, and $f(12) = 7$.

From $C_x \equiv A_x\ W^{f(x)}$ (% M), we obtain $\{C_x\}$ =

{{572402, 1930240, 374715}, {25128, 265158, 350520},
{1674837, 1231458, 1448214}, {110225, 1198155, 757620}},

and $\{C_6^{-1}, C_7^{-1}\}$ = {93176, 1591882}. Let

$$\begin{aligned} Z &\equiv (C_4\ C_{12})\ (C_6^{-1}\ C_7^{-1}) \\ &\equiv (25128 \times 757620)\ (93176 \times 1591882) \\ &\equiv 776394 \times 1123251 \\ &\equiv 689616\ (\%\ 2022169). \end{aligned}$$

Then, $689616 / 2022169 - L / (A_6\ A_7) = (A_4\ A_{12}) / (2022169\ A_6\ A_7)$.

Further, the continued fraction of 689616 / 2022169 is

$$1 / (2 + 1 / (1 + 1 / (13 + 1 / (1 + (1 / (3 + 1 / (2 + 1 / (2 + 1 / (2 + 1 / (97 + 4 / 9)))))))))).$$

Heuristically let

$$L / (A_6 A_7) = 1 / (2 + 1 / (1 + 1/ (13 + 1/ (1 + (1/ (3 + 1 / 2)))))$$
$$= 133 / 390,$$

which indicates that probably $A_6 A_7 = 390$. Because the discriminant

$$689616 / 2022169 - 133 / 390 = 2.235477262\text{e-}6$$
$$< 1 / (2 \times 390^2) = 3.287310979\text{e-}6$$

satisfies the condition at theorem 1 in [8], $A_6 A_7 = 390$ is deduced out.

The integer 390 may be factorized into the pairs (2, 195), (3, 130), (5, 78), (6, 65), (10, 39), (13, 30), or (15, 26), where the elements of (10, 39), (13, 30), and (15, 26) are less than maximal number in $\{A_x\}$. Thus, true $(A_6, A_7) = (26, 15)$ is included in 6 potential cases. Here, $a_u = 2$ and also $a_{u+1} = 2$, and there is no sharp increase from a_u to a_{u+1}.

Additionally, this example also illustrates that when one attempts to infer the suitable factors of the product $A_{k_1} A_{k_2}$ by $f(i) + f(j) = f(k_1) + f(k_2)$ with every $f(x) \in \Omega = \{n + 1, \ldots, 2n\}$, indeterminacy is increased remarkably.

3.3 Condition at *Fact* 4 Is Insufficient for $f(i) + f(j) = f(k)$

In [6], the attackers attempted to seek A_k dominantly by the converse proposition of *fact* 1.1, and however, disturbing values of A_k are too many to determine the original value of A_k. Therefore, in [8], the attackers attempted to diminish indeterminacy of A_k through *fact* 4 which connotes *fact* 1.1, and is equivalent to each of *fact* 1.2 and 1.3.

To say the least, even if *fact* 4 is valid sometimes, we can prove by a counterexample that the condition at *fact* 4 is insufficient for $f(i) + f(j) = f(k)$.

Example 4.

Still let $n = 6$, $\{A_x\} = \{11, 10, 3, 7, 17, 13\}$, and $M = 510931 > 11 \times 10 \times 3 \times 7 \times 17 \times 13 = 510510$.

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

From $C_x \equiv A_x W^{f(x)}$ (% M), we obtain $\{C_x\} = \{113101, 79182, 175066, 433093, 501150, 389033\}$, and its inverse sequence $\{C_x^{-1}\} = \{266775, 236469, 435654, 149312, 434038, 425203\}$.

Randomly select $i = 1$, $j = 3$, and $k = 6$. In this case, $f(6) = 8 \neq f(1) + f(3) = 9 + 10$. Compute

$$Z \equiv C_1 C_3 C_6^{-1}$$
$$\equiv 113101 \times 175066 \times 425203$$
$$\equiv 425865 \text{ (\% } 510931).$$

Presume that W in $C_1 C_3$ is just neutralized by W^{-1} in C_6^{-1}, then

$$425865 \equiv A_1 A_3 A_6^{-1} \text{ (\% } 510931).$$

According to *alg*.1 in [8],

$$425865 / 510931 - L / A_6 = A_1 A_3 / (510931 A_6).$$

Compute the continued fraction of186640 / 510931 being

$$1 / (1 + 1 / (5 + 1 / (159 + 1 / 535))).$$

Heuristically let

$$L / A_6 = 1 / (1 + 1 / 5) = 5 / 6,$$

which indicates that probably $A_6 = 6$. Further, can verify that

$$425865 / 510931 - 5 / 6 = 0.000174518$$
$$< 1 / (2 \times 6^2) = 0.0138889$$

satisfies the condition at theorem 1 in [8].

Let $u = 2$, and $q_u = A_k = A_6 = 6$.

Then $p_{u+1} / q_{u+1} = p_3 / q_3 = 1 / (1 + 1 / (5 + 1 / 159)) = 796 / 955$, and

$$A_k (M / (2A_i A_j A_k))^{1/2} = 6 (510931 / (2 \times 11 \times 3 \times 6))^{1/2}$$
$$= 6 \times 35.9197 = 215.5186.$$

In addition, evidently

$prime\langle 1\rangle = 2$, $prime\langle 2\rangle = 3$, $prime\langle 3\rangle = 5$, $prime\langle 4\rangle = 7$, $prime\langle 5\rangle = 11$, $prime\langle 6\rangle = 13$, $prime\langle 7\rangle = 17$, and $prime\langle 8\rangle = 19$ which are according to [8].

Then, by *fact* 4 in [8], $m = 7$, and

$$\Delta = (M / (2\prod_{x=n-2}^{m} prime\langle x\rangle))^{1/2} = (15)^{1/2} = 3.8729.$$

Thus, $q_{u+1} = 955 > A_k (M / (2A_i A_j A_k))^{1/2} = 216$ satisfies *fact* 1.2 namely (3), $a_{u+1} = 159 > a_u = 5$ satisfies *fact* 1.3, and $q_{u+1} = 955 > q_u \Delta \approx 24$ satisfies *fact* 4 and *alg*.1.

By the condition at *fact* 4, $A_6 = 6 < \max A = 221$ is deduced, namely *alg*.1 will output {1, 3, 6, 6}. However, it is in direct contradiction to true $A_6 = 13$, which show the condition at *fact* 4 is not sufficient for $f(i) + f(j) = f(k)$, and every A_x will likely be evaluated to at least two eligible values (see example 5).

Because the condition at *fact* 4 is insufficient for $f(i) + f(j) = f(k)$, *property* 1, 2, 3, 4 and 5 are invalid. Further, the run result of *alg*.1 regarding an arbitrary public key $\{C_1, \ldots, C_n\}$ as an input will contain enormous disturbing data as $n \geq 80$, and it is infeasible that *alg*.2 find out the original coprime sequence $\{A_x\}$ in polynomial time (see example 5), which manifests that *alg*.1 and 2 are invalid.

4 Example in [8] Is Woven Elaborately and Data at *Table* 2 Is Falsified

4.1 Example in [8] Illustrates Nothing about Breaking

It is easily understood that according to *fact* 1.1 and 4, the authors of [8] can weave an example consistent with *alg*.1 and 2 since

① the lever function value $\{f(1), \ldots, f(n)\}$ may be known in advance;

② the coprime sequence $\{A_1, \ldots, A_n\}$ may be selected elaborately in advance;

③ the condition $Z / M - L / A_k < 1 / (2 A_k^2)$ at *fact* 1.1 is necessary for $f(i) + f(j) = f(k)$;

④ the condition $q_{u+1} > q_u (M / (2\prod_{x=n-2}^{m} prime\langle x\rangle))^{1/2}$ at *fact* 4 is necessary for $f(i) + f(j) = f(k)$ sometime.

However, as is indicated in the above rebutment, a consistent example does not illustrates that a related $\{A_x\}$ can be extracted accurately from an arbitrary public key $\{C_x\}$ when $\{f(x)\}$ and $\{A_x\}$ are unknown in advance. The authors of [8] at most broke "their own REESSE1+", which diverted themselves, but not our REESSE1+ with choice parameters. It is well understood that even though a cryptosystem is RSA or ECC, its parameter is must also selected; otherwise the cryptosystem is insecure.

The example in [8] is neither readable nor verifiable in short time, and the proportion of n to $\log_2 M$ is not also proper, which contravenes the optimization principle for the modulus M in the REESSE1+ cryptosystem. An obvious truth is that if M is too large, the length of a public key will increase rapidly. Therefore, M should be as small as possible while at least meets $M > \prod_{x=1}^{n} A_x$ meantime. Selection of the sequence $\{A_x\}$ in [8] also contravenes the optimization principle.

The intent for [8] to select such a large M that n is out of proportion to $\log_2 M$ seems to want to increase the necessity of the conditions at *fact* 1.1 and 4 for $f(i) + f(j) = f(k)$. However, it can not increase the sufficiency of the conditions.

4.2 Data at *Table* 2 Is Falsified for a Compatible Effect

In above paragraphs, we illustrate that the condition $Z / M - L / A_k < 1 / (2 A_k^2)$ at *fact* 1.1, namely (1″) is insufficient for $f(i) + f(j) = f(k)$. Property I will make us better understand it.

Property I: Let $C_x \equiv A_x W^{f(x)} (\% M)$, where every $x \in [1, n]$, $A_x \leq \rho$, $f(x) \in \{5, \ldots, n + 4\}$, $M > \prod_{x=1}^{n} A_x$ is a prime. Then, $\forall i, j, k \in [1, n]$, even if $f(i) + f(j) \neq f(k)$,

1) there always exist

$$C_i \equiv A'_i W'^{f'(i)}, C_j \equiv A'_j W'^{f'(j)}, \text{ and } C_k \equiv A'_k W'^{f'(k)} (\% M)$$

such that $f'(i) + f'(j) \equiv f'(k) (\% \overline{M})$ with $A'_k \leq \rho$.

2) C_i, C_j, C_k make (1″) hold with $A'_k \leq \rho$ in all probability.

Proof:

1)

Let O_d be an oracle for a discrete logarithm.

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

In terms of group theories, $\forall A'_k \in \{2, \ldots, \acute{\rho}\}$, the equation

$$C_k \equiv A'_k W'^{f'(k)} \ (\% M)$$

has a solution. $f'(k)$ may be taken through O_d.

$\forall f'(i) \in [1, \overline{M}]$, and let

$$f'(j) \equiv f'(k) - f'(i) \ (\% \overline{M}).$$

Then, from $C_i \equiv A'_i W'^{f'(i)}$ and $C_j \equiv A'_j W'^{f'(j)} \ (\% M)$, we can obtain many distinct pairs (A'_i, A'_j), where $A'_i, A'_j \in (1, M)$, and $f'(i) + f'(j) \equiv f'(k) \ (\% \overline{M})$.

2)

Let

$$\begin{aligned} Z &\equiv C_i C_j C_k^{-1} \\ &\equiv A'_i A'_j W'^{f'(i)+f'(j)} (A'_k W'^{f'(k)})^{-1} \ (\% M) \end{aligned}$$

with $f'(i) + f'(j) \equiv f'(k) \ (\% \overline{M})$ but $f(i) + f(j) \neq f(k)$.

Further, there is $A'_i A'_j \equiv C_i C_j C_k^{-1} A'_k \ (\% M)$.

It is easily seen from the above equations the values of W' and $f'(k)$ do not influence the value of $A'_i A'_j$.

If $A'_k \in [2, \acute{\rho}]$ changes, $A'_i A'_j$ also changes. Thus, $\forall i, j, k \in [1, n]$, the number of value of $A'_i A'_j$ is $\acute{\rho} - 1$.

Let $M = 2q\acute{\rho}^2 A'_k$, where q is a rational number.

According to (1),

$$\begin{aligned} Z / M - L / A'_k &= A'_i A'_j / (M A'_k) \\ &= A'_i A'_j / (2q \acute{\rho}^2 A'^2_k). \end{aligned}$$

When $A'_i A'_j \leq q \acute{\rho}^2$, there is

$$\begin{aligned} Z / M - L / A'_k &\leq q \acute{\rho}^2 / (2q\acute{\rho}^2 A'^2_k) \\ &= 1 / (2A'^2_k) \end{aligned}$$

which satisfies (1″).

Assume that the value of $A'_i A'_j$ distributes uniformly on $(1, M)$. Then, the probability that $A'_i A'_j$ makes (1″) hold is

$$\begin{aligned} P_{\forall i, \ j, \ k \in [1,n]} &= (q \acute{\rho}^2 / (2 q \acute{\rho}^2)) (1 / 2 + \ldots + 1 / \acute{\rho})) \\ &\geq (1 / 2)(2 (\acute{\rho} - 1) / (\acute{\rho} + 2)) \\ &= 1 - 3 / (\acute{\rho} + 2). \end{aligned}$$

It is seen that the probability is very large. □

According to property I.2, for a certain $C_k \in \{C_1, \ldots, C_n\}$ and $\forall C_i, C_j \in \{C_1, \ldots, C_n\}$, A_k will have roughly n^2 values by (1″) namely the condition at *fact* 1.1, including the repeated, and considering the symmetry, almost every value has at least one counterpart.

Of course, if the condition at *fact* 4, namely $q_{u+1} > q_u \Delta$ which connotes (1″) is used as a constraint, the number of values of $A_k = q_u$ will decrease. Example 4 already shows that even though $f(i) + f(j) \neq f(k)$, an eligible A_k can still be found.

Notice that when i, j, k all fix on, it is fully possible that L / A_k has multiple satisfactory values, which implies multiple convergents of the continued fraction of Z / M likely meet (1″) and even $q_{u+1} > q_u \Delta$.

To clarify the matter thoroughly, we program by *alg*.1 in MS Visual C++, make an executable file, repeat the experiment regarding the public key at the example in [8] as input, and obtain the following output which is classified the same as in [8]:

A_k	Tuples (i, j, k)
$A_1 = 9$	(9, 9, 1)
$A_2 = 253$	(7, 5, 2), (9, 6, 2), (5, 7, 2), (6, 9, 2)
$A_3 = 16127$	(10, 7, 3), (7, 10, 3)
$A_4 = 3$	(8, 3, 4), (3, 8, 4)
$A_4 = 205$	(9, 3, 4), (6, 5, 4), (5, 6, 4), (7, 7, 4), (3, 9, 4)
$A_4 = 152391460756$	(8, 7, 4), (7, 8, 4)
$A_6 = 53022327$	(4, 3, 6), (3, 4, 6)
$A_6 = 318461273008612$	(4, 3, 6), (3, 4, 6)
$A_6 = 4471789987666990$	(5, 3, 6), (3, 5, 6)
$A_6 = 1572955621791218$	(5, 5, 6)
$A_8 = 2809$	(5, 5, 8), (9, 7, 8), (7, 9, 8)
$A_{10} = 49$	(9, 5, 10), (5, 9, 10)
$A_{10} = 1894$	(9, 6, 10), (6, 9, 10)
$A_{10} = 6957$	(9, 7, 10), (7, 9, 10)

Table I: Output of the program by *alg*. 1 given the public key at the example in [8]

Obviously, *table* 2 in [8] misrepresented $A_3 = 16127$ as $A_4 = 16127$, and $A_6 = 53022327$ as $A_{10} = 53022327$. What gets worse is that *table* 2 mutilated the two tuple data (4, 3, 6, 318461273008612) and (3, 4, 6, 318461273008612), which is a type of data falsification. These two tuple data illustrate that for fixed i, j, k, the L / A_k may have several satisfactory values, namely the several convergents of the continued fraction of Z / M meet *fact* 4 meantime, which reflects the insufficiency of the condition $q_{u+1} > q_u\Delta$ further, increases the indeterminacy of A_k greatly, and weakens the reliability of *alg*.1 in [8] greatly.

4.3 Example in [8] Is Woven Elaborately and Alg.2 in [8] Is Invalid

In the above, it is mentioned that at most the authors of [8] broke "their own REESSE1+", because the example in [8] is woven elaborately, and the parameters $\{A_x\}$ and $\{f(x)\}$ are selected deliberately.

If we use another set of parameters for producing a public key as the input of the program by *alg*.1, the output result will contains so many disturbing data that the original sequence $\{A_1, \ldots, A_n\}$ can not be distinguished in polynomial time.

Example 5.

Let $n = 10$, $\{A_x\} = \{437, 221, 77, 43, 37, 29, 41, 31, 15, 2\}$, and

$M = 13082761331670077 > \prod_{x=1}^{n} A_x = 13082761331670030$.

Arbitrarily select $W = 944516391$, $f(1) = 11$, $f(2) = 14$, $f(3) = 13$, $f(4) = 8$, $f(5) = 10$, $f(6) = 5$, $f(7) = 9$, $f(8) = 7$, $f(9) = 12$, $f(10) = 6$.

According to $C_x \equiv A_x W^{f(x)} \ (\% M)$, we obtain

$\{C_x\}$ = {3534250731208421, 12235924019299910, 8726060645493642, 10110020851673707, 2328792308267710, 8425476748983036, 6187583147203887, 10200412235916586, 9359330740489342, 5977236088006743}.

Input the public key $\{C_x\}$ into the program by *alg*.1, and obtain $\Delta = 506$, $\max A = 58642670$, and the following tuples greater than 100:

A_k	Tuples (i, j, k)
$A_1 = 187125$	(1, 1, 1)
$A_1 = 121089$	(2, 1, 1), (1, 2, 1)
$A_1 = 77$	(5, 3, 1), (3, 5, 1)
$A_1 = 23$	(8, 6, 1), (6, 8, 1), (10, 10, 1)
$A_1 = 437$	(10, 6, 1), (6, 10, 1)
$A_2 = 1251$	(1, 1, 2)
$A_2 = 187125$	(2, 1, 2), (1, 2, 2)
$A_2 = 121089$	(2, 2, 2)
$A_2 = 17$	(8, 4, 2), (6, 5, 2), (5, 6, 2), (10, 7, 2), (4, 8, 2), (7, 10, 2)
$A_2 = 221$	(10, 4, 2), (7, 6, 2), (6, 7, 2), (8, 8, 2), (4, 10, 2)
$A_2 = 77$	(9, 8, 2), (8, 9, 2)
$A_2 = 4204$	(10, 10, 2)
$A_3 = 187125$	(3, 1, 3), (1, 3, 3)
$A_3 = 12$	(7, 1, 3), (1, 7, 3)
$A_3 = 121089$	(3, 2, 3), (2, 3, 3)
$A_3 = 77$	(6, 4, 3), (4, 6, 3), (10, 8, 3), (8, 10, 3)
$A_3 = 11$	(10, 4, 3), (7, 6, 3), (6, 7, 3), (8, 8, 3), (4, 10, 3)
$A_3 = 2113$	(8, 7, 3), (7, 8, 3)
$A_3 = 769$	(9, 8, 3), (8, 9, 3)
$A_4 = 187125$	(4, 1, 4), (1, 4, 4)
$A_4 = 121089$	(4, 2, 4), (2, 4, 4)
$A_4 = 76$	(10, 6, 4), (6, 10, 4)
$A_4 = 56$	(10, 9, 4), (9, 10, 4)
$A_5 = 187125$	(5, 1, 5), (1, 5, 5)
$A_5 = 630269$	(6, 1, 5), (1, 6, 5)
$A_5 = 121089$	(5, 2, 5), (2, 5, 5)
$A_5 = 41$	(8, 2, 5), (2, 8, 5)
$A_5 = 97$	(4, 3, 5), (3, 4, 5)
$A_5 = 37$	(6, 6, 5), (10, 6, 5), (6, 10, 5)
$A_6 = 187125$	(6, 1, 6), (1, 6, 6)
$A_6 = 121089$	(6, 2, 6), (2, 6, 6)
$A_7 = 187125$	(7, 1, 7), (1, 7, 7)
$A_7 = 121089$	(7, 2, 7), (2, 7, 7)

$A_7 = 3$	(9, 3, 7), (3, 9, 7)
$A_8 = 187125$	(8, 1, 8), (1, 8, 8)
$A_8 = 34945619$	(6, 2, 8), (2, 6, 8)
$A_8 = 121089$	(8, 2, 8), (2, 8, 8)
$A_9 = 187125$	(9, 1, 9), (1, 9, 9)
$A_9 = 121089$	(9, 2, 9), (2, 9, 9)
$A_9 = 5$	(6, 4, 9), (4, 6, 9), (10, 8, 9), (8, 10, 9)
$A_9 = 15$	(8, 6, 9), (6, 8, 9), (10, 10, 9)
$A_{10} = 259970$	(4, 1, 10), (1, 4, 10)
$A_{10} = 187125$	(10, 1, 10), (1, 10, 10)
$A_{10} = 121089$	(10, 2, 10), (2, 10, 10)
$A_{10} = 7629$	(8, 3, 10), (3, 8, 10)

Table II: Output of the program by *alg*. 1 given the public key at example 5

From table II, we observe that

A_k from 5 tuples is $A_2 = 221$ or $A_3 = 11$,

A_k from 4 tuples is $A_3 = 77$ or $A_9 = 5$,

A_k from 3 tuples is $A_1 = 23$, $A_5 = 37$, or $A_9 = 15$,

A_k from 2 tuples is $A_1 = 77$, $A_2 = 77$, $A_3 = 12$, $A_4 = 56$, $A_5 = 41$, or $A_7 = 3$ etc, and

A_k from 1 tuples is $A_1 = 187125$, $A_2 = 1251$, $A_2 = 121089$, or $A_2 = 4204$.

Among these A_k's, there exist at least 2^{n-5} compatible combinations.

For instance, arbitrarily select compatible $A_3 = 11$, $A_9 = 5$, $A_1 = 23$, $A_5 = 41$, and $A_2 = 1251$, and find out $f(3) = 14$, $f(9) = 13$, $f(1) = 12$, $f(5) = 11$, and $f(2) = 10$ by Table 1 in [8].

Again for instance, arbitrarily select compatible $A_3 = 11$, $A_9 = 5$, $A_5 = 37$, $A_7 = 3$, and $A_1 = 187125$, and find out $f(3) = 14$, $f(9) = 13$, $f(5) = 12$, $f(7) = 11$, and $f(1) = 10$ by Table 1 in [8].

Therefore, if keep $\Omega = \{5, .., n + 4\}$ unvaried, we may select fit $\{A_x\}$ and W so as to make the time complexity of the continued fraction attack by $q_{u+1} > q_u\Delta$ and *table* 1 get to at least $O(2^n)$, which elucidates that the example woven elaborately in [8] has no practical meaning, and *alg*.2 in [8] is invalid.

However, we had best select fit Ω while let $\{A_x\}$ and W random so as to avoid attack by (1′) (see sect.5.1).

4.4 Distribution of Tuples Relating A_k does not Follow *Table* 1 in [8]

In addition, from table II we also observe that $A_2 = 17$ involves 6 tuples, and $A_5 = 37$ involves 3 tuples (but in fact, 6 tuples is impossible, and $f(5) = 10$), which indicates that the distribution of tuples relating A_k does not follow *table* 1 in [8]. Besides, considering $A_3 = 11$ from 5 tuples, $A_9 = 5$ from 4 tuples etc, we see that *table* 1 is insufficient for $f(i) + f(j) = f(k)$, that is, the converse proposition of fact 2.2 does not hold.

5 Why Is $C_x \equiv A_x W^{f(x)} \ (\% \ M)$ Changed to $C_x \equiv (A_x W^{f(x)})^{\delta} \ (\% \ M)$ in REESSE1+ v2.1

5.1 Lever Set Ω Needs to Be Complicated When $C_x \equiv A_x W^{f(x)} \ (\% \ M)$

In REESSE1, $C_x \equiv A_x W^{f(x)} \ (\% \ M)$ with $f(x) \in \Omega = \{5, \ldots, n + 4\}$.

In REESSE1+, $C_x \equiv A_x W^{f(x)} \ (\% \ M)$ with $f(x) \in \Omega = \{5\delta, \ldots, (n+4)\delta \mid \delta \geq 1\}$, $\{5 + \delta, \ldots, (n+4) + \delta \mid \delta \geq$

$n-4\}$, {5, 7, …, 19, 53, 55, …} etc.

If let $W' = W^{\delta} (\% M)$, we see that $\{5\delta, \ldots, (n+4)\delta \mid \delta \geq 1\}$ is substantially the same as $\{5, \ldots, n+4\}$.

Although [8] by $Z / M - L / A_k < 1 / (2A_k^2)$ and $q_{u+1} > q_u \Delta$ can not break REESSE1+ with $C_x \equiv A_x W^{f(x)} (\% M)$ and $\Omega = \{5\delta, \ldots, (n+4)\delta \mid \delta \geq 1\}$, attack by $Z / M - L / A_k < 1 / (2^{n-2-1} A_k^2)$, namely (1′) will filter out the most of disturbing data as n is large, which makes REESSE1+ be faced with danger. Therefore, in REESSE1+ with $C_x \equiv A_x W^{f(x)} (\% M)$, Ω needs to be complicated, namely had best select $\Omega = \{5, 7, \ldots, 19, 53, 55, \ldots\}$ which is an odd set of $2n$ elements such that ① $\forall e_1, e_2 \in \Omega, e_1 \neq e_2$, ② $\forall e_1, e_2, e_3 \in \Omega, e_1 + e_2 \neq e_3$, ③ $\forall e_1, e_2, e_3, e_4 \in \Omega, e_1 + e_2 + e_3 \neq e_4$.

5.2 Key Transform $C_x \equiv A_x W^{f(x)} (\% M)$ Needs to Be Strengthened When Still $\Omega = \{5, \ldots, n+4\}$

In REESSE1+ with $C_x \equiv A_x W^{f(x)} (\% M)$ and $f(x) \in \Omega = \{5, 7, \ldots, 19, 53, 55, \ldots\}$, because the elements of Ω are relatively large, decryption speed will decrease greatly.

To keep $\Omega = \{5, \ldots, n+4\}$ unvaried, the key transform should be strengthened, so in REESSE1+ v2.1, we let $C_x \equiv (A_x W^{f(x)})^{\delta} (\% M)$. In this way, REESSE1+ v2.1 is not only secure but also swift.

6 Attack on the Signature Is an Eisegesis

6.1 $T^{-1} \% \bar{M}$ does not Exist and $Q^{-1} \% \bar{M}$ not Necessarily Exist

Section 4 of the original [8] deduces $U \equiv ((Q / H)^{1/S} \hat{\boldsymbol{G}} (GW)^{-1} \delta^{\delta(\delta+1)-1/S})^{QT} (\% M)$, which is right.

However, $(GW)^{-1} \delta^{\delta(\delta+1)-1/S} \equiv ((Q / H)^{-S-1} \hat{\boldsymbol{G}}^{-1}) U^{(QT)-1} (\% M)$ further given in [8] is wrong because $T^{-1} \% \bar{M}$ with $T \mid \bar{M}$ does not exist, and neither does $Q^{-1} \% \bar{M}$ exist when $\gcd(Q, \bar{M}) > 1$. In the signature algorithm, it is easy to let $\gcd(Q, \bar{M}) > 1$.

Denote $x = (GW)^{-1} \delta^{\delta(\delta+1)-1/S} (\% M)$.

Then, the trivial solution to $x^{QT} \equiv U ((Q / H)^{1/S} \hat{\boldsymbol{G}})^{-QT} (\% M)$ does not exist when $\gcd(T, \bar{M} / T) > 1$.

Due to stipulating $T \geq 2^n$ in the key generation algorithm, the time complexity of finding out a random solution to $x^{QT} \equiv U((Q / H)^{1/S} \hat{\boldsymbol{G}})^{-QT}$ is at least max $(O(2^{n-1}), O(M / (QT)))$ through the probabilistic algorithm [10].

If a solution to $x^{QT} \equiv U((Q / H)^{1/S} \hat{\boldsymbol{G}})^{-QT}$ is found through the discrete logarithm method, the probability that the solution is just equal to $(GW)^{-1} \delta^{\delta(\delta+1)-1/S} (\% M)$ is at most $1 / 2^n$.

If denote $x = ((GW)^{-1} \delta^{\delta(\delta+1)-1/S})^{T} (\% M)$, then $x^{Q} \equiv U ((Q / H)^{1/S} \hat{\boldsymbol{G}})^{-QT} (\% M)$.

When $\gcd(Q, \bar{M}) > 5$ and $M / Q > 2^n$, seeking a solution to $x^{Q} \equiv U ((Q / H)^{1/S} \hat{\boldsymbol{G}})^{-QT}$ is also at least the discrete logarithm problem.

6.2 Forging Attack in [8] May Be Easily Avoided through Turning $D \mid (\delta Q - W)$ to $D \mid (\delta Q - WH)$

In REESSE1+ [4], we definitely pointed out that $Q \neq Q_1$, where Q is produced currently, and Q_1 is any of signature foreparts produced ever before. Of course, $Q \neq Q_1$ implied that the linear combination of Q_1 with Q_2 should be excluded from signature foreparts. However, such exclusion is infeasible in polynomial time.

Therefore, in practical applications, it is suggested as a shortcut that users move the parameter H in $Q \equiv (RG_0)^{S} H\delta (\% M)$ into $D \mid (\delta Q - W)$, and make $D \mid (\delta Q - W)$ become $D \mid (\delta Q - WH)$. In this wise, the forging attack in [8] is easily avoided, namely Q' can not be forged out at least in polynomial time.

Notice that correspondingly, the λS in the signature algorithm and the discriminant in the verification algorithm should also be adjusted.

7 Conclusion

The above rebuttal shows that each or the combination of (1″), $q_{u+1} > q_u \Delta$, and *table* 1 is not sufficient for $f(i) + f(j) = f(k)$, there exist logic errors in the deduction of (3), and *alg*.1 based on *fact* 4 and *alg*.2 based on *table* 1 are not valid. Additional, the signature forgery attack in [8] is easily avoided. Hence, the conclusion

of [8] that REESSE1+ is not secure at all (which connotes that [8] can extract a related private key from any public key in REESSE1+) is completely incorrect, as long as Ω is fitly selected, REESSE1+ with $C_x \equiv A_x W^{f(x)} (\% M)$ is secure, and the private key attack in [8] like [6] is a pseudo attack.

The authors of [8] attempt to convince people or credulous one of their opinion through an example woven elaborately, and their purpose is to want to suffocate REESSE1+, suppress us, and promote themselves. Especially, [8] like [6] does not list the origin of idea of the continued fraction analysis of REESSE1, and falsifies the data at *table* 2, which violates scientific research ethics and honestness.

We welcome unmalicious, co-promotive, and normal academic criticism which is utterly necessary.

References

[1] Shenghui Su, *The REESSE1 Public-key Encryption Algorithms*, Int. C1: H04L 9/14, ZL01110163.6, Chinese Patent, Apr. 2001.

[2] Shenghui Su, *The REESSE1 Public-key Cryptosystem*, Computer Engineering & Science, Chinese, v25(5), 2003, pp.13-16.

[3] Shenghui Su, Yixian Yang and Bingru Yang, *The Necessity and Sufficiency Analysis of the Lever Function in the REESSE1 Encryption Scheme*, Acta Electronica Sinica, Chinese, v34(10), 2006, pp.1892-1895. (Received May 13, 2005)

[4] Shenghui Su and Shuwang Lü, *The REESSE1+ Public-key Cryptosystem*, http://eprint.iacr.org/2006/420.pdf.

[5] Shengli Liu, Fangguo Zhang and Kefei Chen, *Cryptanalysis of REESSE1 Digital Signature Algorithm*, CCICS 2005, Xi'an, China, May 2005.

[6] Shengli Liu, Fangguo Zhang and Kefei Chen, *Cryptanalysis of REESSE1 Public Encryption Cryptosystem*, Information Security, Chinese, n7, 2005, pp.121-124.

[7] Shenghui Su, *Refuting the Pseudo Attack on the REESSE1 Public-key Algorithms for Encryption*, Computer Engineering and Applications, Chinese, v42(20), 2006, pp.129-133.

[8] Shengli Liu and Fangguo Zhang, *Cryptanalysis of REESSE1+ Public Key Cryptosystem*, http://eprint.iacr.org/2006/480.pdf, Dec. 22, 2006.

[9] Kenneth H. Rosen, *Elementary Number Theory and Its Applications* (5th ed.), Boston: Addison-Wesley, 2005, ch. 12.

[10] Henri Cohen, *A Course in Computational Algebraic Number Theory*, Berlin: Springer-Verlag, 2000, ch. 1, 3.

Remark

The first version of this paper was sent to the authors of [8] via email on Mar. 6, 2007, and the draft of this revised version was sent to the authors of [8] via email between Oct. 23 and Nov. 12, 2009 repeatedly.

The authors of [8] revised section 5 of [8] on Mar. 12, 2007 after they read this paper and the eprint.iacr.org′s demand that [8] should be withdrawn or modified, but the modification avoided the heavy and chose the light.