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

Last Updated September 30, 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 and verification algorithms, and infers that the probability that a plaintext solution is not unique is nearly zeroth. The authors discuss the relation between the lever function and a random oracle, and analyze the security of REESSE1+ against recovering a plaintext from a ciphertext, extracting a private key from a public key or a signature, and faking a digital signature via a public key or via known signatures with a public key. On the basis of analysis, believe that the security of REESSE1+ is at least equal to the time complexity of $O(2^n)$ at present. At last, expound 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][5], which guarantees that, for a sufficiently large setting of a security parameter such as a modular length or a sequence length, "breaking" the cryptosystem will take much more effort than using the cryptosystem. This is referred to as asymptotic security in which the advantage of any computationally bounded adversary is negligible [6]. Asymptotic security is distinguished from concrete security or exact security which is practice-oriented and aims to give more precise estimates of the computational complexities of attack tasks [7].

Along with the elevation of computer speeds, the security parameter will become larger and larger. Therefore, to decrease the bit-length of a security parameter and 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 efficient and more one-way than ElGamal itself. By now, an algorithm for ECC discrete logarithms of complexity sub-exponential in the bit-length of a modulus has not been discovered yet [8]. 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 [9], the subset sum problem which serves as a trapdoor function can not preclude a private key from being inferred through the Shamir method [10]. It is should be noticed that when the knapsack density $D < 1$, the subset sum problem will degenerate to a polynomial time problem from a NPC problem thanks to the L^3 base-reduced algorithm [11][12] which is employed for finding the shortest vector or a approximately shortest vector of a lattice base [13].

Different from MH, RSA, and ElGamal, REESSE1+ brings 3 independent variables into the general key transform. Its security is not based on the classical or arisen hardnesses: the subset sum problem, the integer factorization problem, the discrete logarithm problem and the shortest vector problem, but on three new

computational 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.

It is not difficult to understand that if the security of a public-key cryptosystem is based on not less than three hardnesses each of which can not be solved in polynomial time, this cryptosystem may be called a ***multiproblem cryptosystem***. Clearly, The security of a multiproblem cryptosystem is equivalent to that hardness whose time complexity is smallest in all the involved hardnesses.

Accordingly, REESSE1+ is a type of multiproblem cryptosystem. A multiproblem cryptosystem must be a multivariate one or multiparameter one because only multiple variables can bring in multiple hardnesses. Correspondingly, REESSE1+ may be also regarded as a type of multivariate system in a broad sense.

The essence of the multivariate cryptosystem does not lie in what its key transform is a set of quadratic polynomials but what its general key transform contains at least three independent variables.

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 $A_1, A_2, \ldots,$ 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 [14], 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 $|Ł_n| \ge \mathrm{A}_n^n$.
- 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 $đ \geq 5$ is non-large.

We denote by H the sub-group of order $đ$ of $\mathbb{Z}_M^*$.

(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 $đĐT \mid (M-1)$, $q \mid (M-1)$ for any prime $q \in [1, n+4]$, and $1\hat{g}^1, \ldots, đ\hat{g}^đ \,\%\, đ$ pairwise distinct $\exists\, \hat{g} \in H$.

(3) Pick S making $\langle S, M-1\rangle = 1$, $S^{-1} \,\%\, -1$ small, and δ making $\langle \delta, M-1\rangle = 1, |\delta| = đĐT, đ \mid \delta + Đ \,\%\, -1$.

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

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

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

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

Remark: Seeking a S-th root of $x^S \equiv c \,(\%\, M)$ is referred to theorem 1 in section 3.4.

The set $\Omega = \{i\delta \,(\%\, -1) \mid i = 5, \ldots, n+4\}$ is not unique — $\Omega = \{i + \delta \,(\%\, -1) \mid i = 5, \ldots, n+4\}$ for example, and moreover the evaluations of i may not be successive. In practice, the principles for selecting Ω are that 1) $\ell(i) \geq 5$; 2) the elements of 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 [15].

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 $(\{A_i\}, \{\ell(i)\}, W, \delta, Đ, đ, \hat{g})$ 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 $i \le n$ and $G \ne 1$, go to (3).

(5) If $G \ne 1$, 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

It should be noted that in decryption, $\{\ell(i)\}$, $Đ$, $đ$ and $\hat{g}$ are unhelpful.

3.4 The Digital Signature Algorithm

Assume that ($\{A_i\}$, $\{\ell(i)\}$, W, δ, $Đ$, $đ$, $\hat{g}$) 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) Randomly pick κ with $đ \nmid \kappa$ making $\delta^{(\delta Q - WH)T} \equiv \hat{g} \,(\%\, M)$, $đ \nmid (\delta Q)^n - (WH)^n \,\%\, -1$, and $đ \nmid \bar{U} \,\%\, -1$, where Q meets $\kappa Đ \equiv \delta Q - WH \,(\%\, -1)$, R meets $Q \equiv (R G_0)^S \delta \,(\%\, M)$, and $\bar{U} \equiv (R W^{k_1 - 1} \delta^{\delta(\delta + Đ)})^{QT} \,(\%\, M)$.

(4) $\forall\, r \in [1, đ]$, compute $U \leftarrow \bar{U} \hat{g}^r \,\%\, M$. If $đ \nmid (\delta + Đ + r) US + \sum_{i=0}^{n-1} (\delta Q)^{n-1-i} (HW)^i \,\%\, -1$, go to (4).

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

$$\lambda S \equiv ((\delta + Đ + r) US + \sum_{i=0}^{n-1} (\delta Q)^{n-1-i} (HW)^i)(\delta Q - HW) \,(\%\, -1),$$

which indicates $đ Đ \mid \lambda$.

Due to $\bar{U} = (R W^{k_1 - 1} \delta^{\delta(\delta + Đ)})^{QT} \,\%\, M$ and $\hat{g} = \delta^{(\delta Q - WH)T} \,\%\, M$, we see

$$U \equiv \bar{U} \hat{g}^r \equiv (R W^{k_1 - 1})^{QT} \delta^{(\delta Q(\delta + Đ + r) - WHr)T} \,(\%\, M).$$

Let $e_1 = \hat{g}^{r_1} \,\%\, M$, $e_2 = \hat{g}^{r_2} \,\%\, M$, and $\Delta \equiv (\delta + Đ + r_1) S \bar{U} e_1 - (\delta + Đ + r_2) S \bar{U} e_2 \,(\%\, -1)(\%\, đ) \equiv ((e_1 - e_2)(\delta + Đ) + (r_1 e_1 - r_2 e_2)) S \bar{U} \,(\%\, -1)(\%\, đ)$.

Considering $đ \mid M - 1$ and $đ \mid \delta + Đ \,\%\, -1$, then, $\Delta \equiv (r_1 e_1 - r_2 e_2) S \bar{U} \,(\%\, đ)$.

In terms of the key generation algorithm, $(r_1 e_1 - r_2 e_2) \ne 0 \,(\%\, đ)$.

In addition, $\langle S, đ \rangle = 1$ and $đ \nmid \bar{U} \,\%\, -1$, and thus $\Delta \ne 0 \,(\%\, đ)$. Namely $(\delta + Đ + r) US \,(\%\, đ)$ for $r = 1, \ldots, đ - 1$ are pairwise distinct, which implies that the probability of finding out suitable U is $((M-1)/đ) / (M - 1) = 1/đ$. 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 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 $c^{\mu} \,\%\, p$ is one k-th root.

Further, we have theorem 2.

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

The proofs of theorem 1 and 2 are referred to bibliography [16].

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

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 section 3.1 that $\alpha \equiv \delta^{\delta^n} \equiv \delta(WG_0G_1)^S\ (\%\ M)$, $\beta \equiv \delta^{(\delta+Đ)WS}\ (\%\ M)$, and $\gamma \equiv \delta^{W^n}\ (\%\ M)$.

Let $V \equiv (R^{-1}WG_1)^{QU}\delta^{\lambda}\ (\%\ M)$. Because λ meets $\lambda S \equiv ((\delta+Đ+r)US+\sum_{i=0}^{n-1}(\delta Q)^{n-1-i}(HW)^i)(\delta Q - WH)\ (\%\ -1)$, may let $\lambda = k đ Đ$, where k is a certain integer, and then

$$\begin{aligned} Q^{QU}V^S &\equiv (RG_0)^{SQU}\delta^{QU}(R^{-1}WG_1)^{QUS}\delta^{\lambda S} \\ &\equiv (WG_0G_1)^{QUS}\delta^{QU}\delta^{\lambda S} \\ &\equiv \alpha^{QU}\delta^{((\delta+Đ+r)US+\sum_{i=0}^{n-1}(\delta Q)^{n-1-i}(WH)^i)(\delta Q-WH)} \\ &\equiv \alpha^{QU}\delta^{-(\delta+Đ)WHUS}\delta^{(\delta Q(\delta+Đ+r)-rWH)US}\delta^{(\delta Q)^n-(WH)^n} \\ &\equiv \alpha^{QU}\beta^{-UH}\delta^{(\delta Q(\delta+Đ+r)-rWH)US}\alpha^{Q^n}\gamma^{-H^n}\ (\%\ M). \end{aligned}$$

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

$$\begin{aligned} V^{ST} &\equiv (\alpha Q^{-1})^{QUT}\alpha^{Q^nT}\beta^{-UHT}\gamma^{-H^nT}\delta^{(\delta Q(\delta+Đ+r)-WHr)UST} \\ &\equiv X\beta^{-UHT}\gamma^{-H^nT}\delta^{(\delta Q(\delta+Đ+r)-WHr)UST}\ (\%\ M). \end{aligned}$$

In addition,

$$\begin{aligned} U^UV^T &\equiv (RW^{k_1-1})^{QUT}\delta^{(\delta Q(\delta+Đ+r)-WHr)UT}(R^{-1}WG_1)^{QUT}\delta^{\lambda T} \\ &\equiv (W^{k_1}G_1)^{QUT}\delta^{(\delta Q(\delta+Đ+r)-WHr)UT}\delta^{\lambda T} \\ &\equiv \hat{\boldsymbol{G}}^{QUT}\delta^{(\delta Q(\delta+Đ+r)-WHr)UT}\delta^{k đ ĐT} \\ &\equiv \hat{\boldsymbol{G}}^{QUT}\delta^{(\delta Q(\delta+Đ+r)-WHr)UT}\ (\%\ M). \end{aligned}$$

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

$$V^{ST} \equiv (\hat{\boldsymbol{G}}^{QT}U^{-1})^{US}\delta^{(\delta Q(\delta+Đ+r)-WHr)UST}\ (\%\ M).$$

By the double congruence theorem, there is

$$V^{ST} \equiv X\beta^{-UHT}\gamma^{-H^nT}\delta^{(\delta Q(\delta+Đ+r)-WHr)UST} \equiv (\hat{\boldsymbol{G}}^{QT}U^{-1})^{US}\delta^{(\delta Q(\delta+Đ+r)-WHr)UST}\ (\%\ M).$$

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 [14], there exists a pair of integers u and v making $us + vt = 1$. Thus,

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

The above two equations multiplying yields

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

Furthermore, we have

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

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 [17], 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.

- 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 $\hat{\boldsymbol{G}}(W^{-\delta})^k \equiv G\ (\%\ M)$.

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

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

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

An oracle is a mathematical abstraction, a theoretical black box, or a subroutine the running time of which may not be considered [15][18]. 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 [19].

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

In fact, it draws attention that certain artificial signature and encryption schemes are proven secure in the random oracle model, but are trivially insecure when any real function such as the hash function MD5 or SHA-1 is substituted for the random oracle [20]. Nevertheless, for any more natural protocol, a proof of security in the random oracle model gives very strong evidence that an attacker have to discover some unknown and undesirable property of the hash function used in the protocol.

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

It is known from section 3.7 that the randomness of $\ell(.)$ does exist. For example, with respect to $h = 1$, $m = 2$ and $\delta = 1$, $\forall\, x_1, x_2, y_1 \in [5, n + 4]$, when $\ell(x_1) + \ell(x_2) \neq \ell(y_1)$, there exist

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

such that $\ell'(x_1) + \ell'(x_2) = \ell'(y_1)$.

Suppose that O_d is an oracle for a discrete logarithm.

Suppose that R_ℓ is an oracle for the lever function $\{\ell(i)\}$.

The structure of R_ℓ is as follows:

Input: $\{C_1, \ldots, C_n\}$. Output: $\{\ell(1), \ldots, \ell(n)\}$.

(1) Randomly pick a coprime sequence $\{A_1, \ldots, A_n\}$ and W,
where every $A_i \le P_u$, and $W \in \mathbb{Z}_M^*$ is a generator.
(Letting W be a generator does not affect our discussion)

(2) Compute $\ell(i)$ in $C_i \equiv A_i W^{\ell(i)} \pmod M$ by calling O_d for $i = 1, \ldots, n$.

(3) If there are at least x_1, x_2, y_1 such that $\ell(x_1) + \ell(x_2) = \ell(y_1)$, return $\{\ell(1), \ldots, \ell(n)\}$; or else go to (1).

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

It should be noted that if the time complexity of O_d is not considered, the above algorithm can be executed in polynomial time.

The above algorithm illustrates that the output $\{\ell(i)\}$ depends not only on $\{C_i\}$ but also on random W and $\{A_i\}$. Namely for the same input $\{C_i\}$, the output $\{\ell(i)\}$ is nonunique or randomized. Therefore, R_ℓ is exactly a random oracle, and it is impossible that through R_ℓ the adversary obtains the specific $\{\ell(i)\}$ and other private part generated by the key algorithm.

Notice that in the above algorithm, for the adversary, may every $\ell(i)$ be outside of $[5, n + 4]$ and inconsecutive, only if there are at least x_1, x_2, y_1 making $\ell(x_1) + \ell(x_2) = \ell(y_1)$ hold. We should still remember

that in definition 2, the codomain of $\ell(.)$ is $[1, M-1]$, and the continuity of $\ell(.)$ is not be required.

It is easily understood that in classical cryptosystems the R_ℓ of a resemblant property is inexistent.

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 $\text{abs}(\alpha - r / s) < 1 / (2s^2)$, then r / s is a convergent of the simple continued fraction expansion of α [21].

Reference [21] does not offer a similar theorem with α being a rational number side by side. Therefore, when α is a rational number, theorem 4 should not hold.

The foregoing discussion expounds soundly once more why the indeterministic reasoning based on the continued fraction method is ineffectual.

4.2 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

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

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 $\tilde{N} > A_n^n = n!$.

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 section 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 $\alpha \equiv \delta^{\delta^n}$, $\beta \equiv \delta^{(\delta+D)WS}$ 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 section 5.2.3).

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

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

In terms of section 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 $\hat{\boldsymbol{G}} \equiv C^{\sum_{i=1}^{n} b_i}$ (% 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,

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

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 $\hat{\boldsymbol{G}}$ is converted into the coprime sequence product G, computing the plaintext $b_1 \ldots b_n$ is tractable. Namely there is a trapdoor for $\hat{\boldsymbol{G}}$. 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.

4.4 Avoiding the Adaptive-chosen-ciphertext Attack

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

During the late 1990s, Daniel Bleichenbacher demonstrated a practical adaptive-chosen-ciphertext attack on SSL servers using a form of RSA encryption [22]. Almost at the same time, The Cramer-Shoup asymmetric encryption algorithm was proposed [23]. It is the first efficient scheme proven to be secure against adaptive-chosen-ciphertext attack using standard cryptographic assumptions, which indicates that not all uses of cryptographic hash functions require random oracles — some require only the property of collision resistance, and an extension of the Elgamal algorithm which is extremely malleable.

It is lucky that REESSE1+ can avoid the adaptive-chosen- ciphertext attack. In 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. Hence, 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 adaptive-chosen-ciphertext attack is to append a stochastic fixed-length binary sequence to the terminal of every plaintext bock when it is encrypted. For example, a concrete implementation is referred to the OAEP+ scheme [24].

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 [25], 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

$$Q \equiv (RG_0)^S \delta \ (\% \ M), \text{ and } U \equiv (RW^{k_1-1}\delta^{\delta(\delta+D)})^{QT}\delta^{(\delta Q-WH)Tr} \ (\% \ M).$$

When an attacker wants to seek RG_0, or $RW^{k_1-1}\delta^{\delta(\delta+D)}$, it is equivalent to solving the two congruences

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

For the first congruence, because δ is unknown, and the right of the equation is not a constant, it is impossible to solve the equation for RG_0. If δ is guessed, the probability of hitting δ is $1 / |\delta| < 1 / 2^n$.

For the second congruence, there is $|\delta^{-(\delta Q-WH)Tr}| = đ$ with $đ \geq 5$ being small. Assume $đ$ is guessed, and a solution to $x^{đ} \equiv 1 \ (\% \ M)$ without the trivial one can be found in sub-exponential time. Then $\delta^{-(\delta Q-WH)Tr}$ may be possibly hit. Notice that thanks to usually $M / đ > 2^{80}$, the probabilistic polynomial algorithm in [28] for a high degree congruence is ineffectual.

On the basis of $\delta^{-(\delta Q-WH)Tr}$ known, if $\langle Q, M-1\rangle = 1$ and $\langle T, (M-1)/T\rangle = 1$ with $T \mid (M-1)$, there exists the trivial solution to the second congruence. However, the probability that it is just the specific $RW^{k_1-1}\delta^{\delta(\delta+D)}$ is equal to or less than $1/T \leq 1/2^n$.

Additionally, substituting $G_0^{-1}(Q\delta^{-1})^{S^{-1}} \% M$ deriving from $Q \equiv (RG_0)^S \delta \ (\% \ M)$ for R in the U expression gives

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

Thus,

$$((GW)^{-1}\delta^{\delta(\delta+D)-S^{-1}})^{QT} \equiv U(\hat{\boldsymbol{G}}Q^{S^{-1}})^{-QT}\delta^{-(\delta Q-WH)Tr} \ (\% \ M).$$

Likewise, if $\langle Q, M-1\rangle = 1$, $\langle T, (M-1)/T\rangle = 1$, and the element $\delta^{-(\delta Q-WH)Tr}$ with a small order is hit in sub-exponential time, then the trivial solution to $x^{QT} \equiv U(\hat{\boldsymbol{G}}Q^{S^{-1}})^{-QT}\delta^{-(\delta Q-WH)Tr} \ (\% \ M)$ can be found. However, the probability that it matches to $(GW)^{-1}\delta^{\delta(\delta+D)-S^{-1}}$ is equal to or less than $1/T \leq 1/2^n$. Moreover, neither $(GW)^{-1}$ nor $\delta^{\delta(\delta+D)-S^{-1}}$ can be determined respectively.

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

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

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 $\{y \mid y \equiv g^x \ (\% \ p), x = 1, \ldots, p-1\} = \mathbb{Z}_p^*$ [17].

Assume that k satisfying $(k, p-1) = 1$ is an integer, then also $\{y \mid y \equiv x^k \ (\% \ p), x = 1, \ldots, p-1\} = \mathbb{Z}_p^*$ [16].

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, as $p = 11$, $\{y \mid y \equiv x^x \ (\%\ p), x = 1, \ldots, p-1\} = \{1, 3, 4, 5, 6\}$, 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, $\varphi^{-1}(y) = y f^{-1}(y) / ((y' - y) \log_g e)$.

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

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

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

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

Further,

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

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

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

$$\lambda S \equiv ((\delta+Đ+r)US + \sum_{i=0}^{n-1} (\delta Q)^{n-1-i} (HW)^i)(\delta Q - WH) \ (\% -1).$$

Besides, let $a\,c \equiv (\alpha \delta^{-1})^{1/S}$, and $b\,c \equiv \hat{\boldsymbol{G}}$ (% M).

When δ is found, if suppose a value of a, the c, b can be figured out. Further, when W, $Đ$ are found, and r guessed, the attacker may compute the values of Q and U which make

$$Đ \mid \delta Q - WH, \text{ and } đ \mid (\delta+Đ+r)US + \sum_{i=0}^{n-1} (\delta Q)^{n-1-i} (HW)^i \ (\% -1).$$

Clearly, to seek δ, W and $Đ$, the attacker has to try to solve

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

Can δ, W and $Đ$ be found from the above equation system? If δ exists, there is

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

Notice 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 [15], calculate q, u, and v such that $g^q \equiv \alpha$, $g^u \equiv \beta$, and $g^v \equiv \gamma$ (% M), obtain

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

Accordingly,

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

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 the computation of n^{-1} is referred to theorem 1, theorem 2 and bibliography [16]. Further, due to $W = 0$ is not a solution to the equations, there is

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

Transparently, the above equation is a true polynomial of degree $n-1$ with two variables, which can not be solved via discrete logarithms even though $Đ$ as a factor of $(M-1)$ is found in sub-exponential time. Of course, to enhance intractability, a user may substitute the $Đ$ involved in β with a new variable in the key generation algorithm.

If the Euler phi function $\phi(M-1)$ is worked out, the attacker may seem to resort to the probabilistic algorithm in reference [25]. However, the running time of this algorithm is at least $\max(O(2^{n-2}), O(\phi(M-1) / (n-1)))$. When $n > 80$ or $\phi(M-1) / (n-1) > 2^{80}$, seeking δ is infeasible at present. Furthermore, when $\langle u(q^{-1}v)^{(n-1)n^{-1}}, M-1 \rangle \nmid v$, 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

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

it is a signature fraud on F'.

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

If let $Q' = Q$, Q' does not necessarily satisfy $Đ \mid (\delta Q' - WH')$, and computing U' is 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 Q and R involved in U are random, interrelated exponentially, and the value of U changes intensely between 1 and M, there is no polynomial function or statistic regularity among different U, which means that 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 Adaptive-chosen-message Attack Being Intractable

Conforming to section 3.4, Q satisfies $Đ \mid \delta Q - WH$, namely $Q \equiv (\kappa Đ - WH)\delta^{-1}$ (% –1), where κ is a stochastic integer and $đ \nmid \kappa$.

The randomness of κ leads Q and U to be stochastic, where $Q \equiv (RG_0)^S\delta$ (% M), and $U \equiv (RW^{k_1-1})^{QT}\delta^{(\delta Q(\delta+Đ+r)-WHr)T}$ (% M).

Hence, for an identical file F, there are different signatures on F, that is, (Q, U) has indistinguishability, which is equivalent to the fact that $H = hash(F)$ is a random oracle even if *hash* is substituted with a concrete one-way function.

In terms of reference [19], the signature (Q, U) on F is secure against adaptive-chosen-message attack.

5.5 Chosen-signature Attack Being Intractable

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

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

Assume that the values of Q and U are chosen. If let $\hat{\boldsymbol{G}} = f(H)$, the above congruence is an equation in H. Evidently, it is harder than the DLP to seek H from the congruence. If let $H = f^{-1}(\hat{\boldsymbol{G}})$, similarly, it is harder than the DLP to infer $\hat{\boldsymbol{G}}$ from the above congruence.

In fact, due to $H = b_1 \ldots b_n$, $\hat{\boldsymbol{G}} \equiv \prod_{i=1}^{n} C_i^{b_i}$ (% M), there is no linear or polynomial relation between H and $\hat{\boldsymbol{G}}$. That is, a linear or polynomial function f is not existent (see section 4.3), and neither is a linear or polynomial inverse f^{-1}.

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 Academician Jiren Cai, the chair of the academic committee of SKLOIS, and Academician Changxiang Shen, the director of Trusted Computing Laboratory, Beijing University of Technology for their important guidance, suggestions and helps.

The authors would like to thank the Professors Mulan Liu, Huanguo Zhang, Dingyi Pei, Xuejia Lai, Dengguo Feng, Lequan Min, Bingru Yang, Zhiying Wang, Xiulin Zheng 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*, vol. 21, no. 2, pp. 120–126, Feb. 1978.
[2] T. ElGamal, "A Public-key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.
[3] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.
[4] Oded Goldreich, *Foundations of Cryptography: Basic Tools*. Cambridge, UK: Cambridge University Press, 2001, pp. 31–35.
[5] A. C. Yao, "Theory and Applications of Trapdoor Functions," in *Proc. the 23rd Annual Symposium on the Foundations of Computer Science*, IEEE, 1982, pp. 80–91.
[6] Salil Vadhan. (2004, Jul.). *Computational Complexity*. [Online]. Avail- able: http://eecs.harvard.edu/~salil/papers/encyc.pdf.
[7] M. Bellare and P. Rogaway, "The Exact Security of Digital Signatures — How to Sign with RSA and Rabin," in *Proc. Advance in Cryptology: Eurocrypt '96*, Springer-Verlag, 1996, pp. 399–416.
[8] I. F. Blake, G. Seroussi, and N. P. Smart, *Elliptic Curves in Cryptography*. Cambridge, UK: Cambridge University Press, 1999, ch. 1, 6.
[9] R. C. Merkle and M. E. Hellman, "Hiding information and Signatures in Trapdoor Knapsacks," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 525–530, Sep. 1978.
[10] A. Shamir, "A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem," in *Proc. the 23th IEEE Symposium on the Foundations of Computer Science*, 1982, pp. 145–152.
[11] E. F. Brickell, "Solving Low Density Knapsacks," in *Proc. Advance in Cryptology: CRYPTO '83*, Plenum Press, 1984, pp. 25–37.
[12] M. J. Coster, A. Joux and B. A. LaMacchia etc., "Improved Low-Density Subset Sum Algorithms," *Computational Complexity*, vol. 2, issue 2, pp. 111–128, Dec. 1992.
[13] C. P. Schnorr and M. Euchner, "Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems," *Mathematical Programming*: Series A and B, vol. 66, issue 2, pp. 181–199, Sep. 1994.
[14] Song Y. Yan, *Number Theory for Computing*. (2nd ed.) New York: Springer-Verlag, 2002, ch. 1.
[15] A. J. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. London: CRC Press, 1997, ch. 2, 3, 8.
[16] Paul Garrett, *Making, Breaking Codes: An Introduction to Cryptology*. New Jersey: Prentice-Hall, 2001, ch. 12.
[17] Thomas W. Hungerford, *Algebra*. New York: Springer-Verlag, 1998, ch. 1–3.
[18] D. Z. Du and K. Ko, *Theory of Computational Complexity*. New York: John Wiley & Sons, 2000, ch.3-4.
[19] M. Bellare and P. Rogaway, "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols," in *Proc.* the 1st *ACM Conference on Computer and Communications Security*, New York: ACM Press, 1993, pp. 62–73.
[20] Ran Canetti, Oded Goldreich and Shai Halevi, "The Random Oracle Methodology Revisited, " in *Proc. the 30th Annual ACM Symposium on Theory of Computing*, New York: ACM Press, 1998, pp. 209–218.
[21] Kenneth H. Rosen, *Elementary Number Theory and Its Applications*. (5th ed.) Boston: Addison-Wesley, 2005, ch. 12.
[22] Daniel Bleichenbacher, "Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1," in *Proc. Advance in Cryptology: Crypto '98*, Springer-Verlag, 1998, pp. 1–12.
[23] Ronald Cramer and Victor Shoup, "A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack," in *Proc. Advance in Cryptology: Crypto '98*, Springer-Verlag, 1998, pp. 13–25.
[24] Victor Shoup, "OAEP Reconsidered," in *Proc. Advance in Cryptology: Crypto '01*, Springer-Verlag, 2001, pp. 239–259.
[25] Henri Cohen, *A Course in Computational Algebraic Number Theory*. New York: Springer-Verlag, 2000, ch. 1, 3.