A Public Key Cryptoscheme Using Bit-pair Shadows*

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

[1] College of Computers, Beijing University of Technology, Beijing 100124
[2] Graduate School, Chinese Academy of Sciences, Beijing 100039
[3] School of Mathematics, Peking University, Beijing 100871

Abstract: The authors give the definition and property of a bit-pair shadow, and design the three algorithms of a public key cryptoscheme that is based on a multivariate permutation problem (MPP) and an anomalous subset product problem (ASPP) to which no subexponential time solutions are found so far, and regards a bit-pair as an operation unit. Further, demonstrate that the decryption algorithm is correct, deduce the probability that a plaintext solution is nonunique is nearly zero, dissect the running times of the three algorithms, analyze the security of the new scheme against extracting a private key from a public key and recovering a related plaintext from a ciphertext by LLL lattice basis reduction, meet-in-the-middle dichotomy, and adaptive-chosen-ciphertext approach on the assumption that an integer factorization problem, a discrete logarithm problem, and a low-density subset sum problem can be solved efficiently, and prove that new scheme using random both padding and permutation is semantically secure. Meantime, give a conversion from an ASPP to an anomalous subset sum problem (ASSP). The analysis shows that the bit-pair method increases the density of a related ASSP knapsack to $D > 1$, and decreases the modulus length of the new scheme to $\lceil \lg M \rceil = 464$, 544, or 640 corresponding to $n = 80$, 96, or 112 separately.

Keywords: Public key cryptoscheme; Semantical security; Bit-pair shadow; Random padding Anomalous subset sum problem; Compact sequence

1 Introduction

In [1], we propose a prototypal public key cryptosystem called REESSE1+ which is based on the three new provable problems, contains the five algorithms, and is used for data encryption and digital signing.

In REESSE1+, a ciphertext is defined as $\bar{G} \equiv \prod_{i=1}^{n} C_i^{\dot{b}_i} \ (\% M)$, an anomalous subset product problem (ASPP), where $\dot{b}_i$ is the bit shadow of a bit b_i [1] (also see Section 2.2), and n is the bit-length of a plaintext block.

Let $C_1 \equiv g^{u_1} \ (\% M)$, …, $C_n \equiv g^{u_n} \ (\% M)$, and $\bar{G} \equiv g^{v} \ (\% M)$, where g is a generator of $(\mathbb{Z}_M^*, \cdot)$ which can be found in tolerable subexponential time when the modulus $M < 2^{1024}$ can be factorized [2]. Then solving $\bar{G} \equiv \prod_{i=1}^{n} C_i^{\dot{b}_i} \ (\% M)$ for $\dot{b}_1 \ldots \dot{b}_n$ is equivalent to solving

$$\dot{b}_1 u_1 + \ldots + \dot{b}_n u_n \equiv v \ (\% \ \bar{M}). \tag{1}$$

where v may be substituted with $v + k\bar{M}$ along with $k \in [0, n-1]$ [3].

Equation (1) is called an anomalous subset sum problem (ASSP) due to every $\dot{b}_i \in [0, n]$ [1]. Likewise, due to every $\dot{b}_i \in [0, n]$, $\{u_1, \ldots, u_n\}$ is called a compact sequence [4].

May convert an ASSP into a subset sum problem (SSP) through converting u_i to a binary number, and thus the density of an ASSP knapsack is defined as

$$\begin{aligned} D &= \textstyle\sum_{i=1}^{n} \lceil \lg n \rceil / \lceil \lg M \rceil \\ &= n \lceil \lg n \rceil / \lceil \lg M \rceil. \end{aligned} \tag{2}$$

Evidently, the parameters $\lceil \lg M \rceil$ and n have an important influence on the value of D.

In REESSE1+, there are $n = 80$, 96, 112, or 128, and correspondingly $\lceil \lg M \rceil = 696$, 864, 1030, or 1216. Substituting the parameters with concrete values yields

$D = 80 \times 7 / 696 \approx 0.8046 < 1$ for $n = 80$ and $\lceil \lg M \rceil = 696$,
$D = 96 \times 7 / 864 \approx 0.7778 < 1$ for $n = 96$ and $\lceil \lg M \rceil = 864$,
$D = 112 \times 7 / 1030 \approx 0.7612 < 1$ for $n = 112$ and $\lceil \lg M \rceil = 1030$,
$D = 128 \times 8 / 1216 \approx 0.8421 < 1$ for $n = 128$ and $\lceil \lg M \rceil = 1216$.

The above values mean that the original solution to an ASSP may possibly be found through the LLL lattice basis reduction algorithm [5][6]. However, it is uncertain to find the original solution to the

* This work is supported by MOST with Project 2007CB311100 and 2009AA01Z441. Corresponding email: reesse@126.com.

ASSP since the density $D < 1$ of an ASSP only assure that the shortest vector is unique in a related lattice, and cannot assure that the vector of the original solution is just the shortest vector or an approximately shortest vector which will occur in the final reduced lattice basis.

The LLL reduction algorithm is famous for it has a fatal threat to the classical MH knapsack cryptosystem [7] which produces a ciphertext in the form of a subset sum problem.

To avoid the low density of a knapsack from an ASPP and to decrease the modulus length of a cryptoscheme, on the basis of REESSE1+, we propose a new cryptoscheme called JUNA which treats a bit-pair as an operation unit and introduces random ingredients when a bit string is encrypted, and moreover it is proved to be semantically secure.

Throughout this paper, unless otherwise specified, $n \geq 80$ is the bit-length of a plaintext block, $\tilde{n} \geq 144$ is the item-length of a public key sequence, the sign % denotes "modulo", $\bar{M}$ does "$M-1$" with M prime, $\lg x$ means the logarithm of x to the base 2, $\neg$ does the opposite value of a bit, $\dot{P}$ does the maximal prime allowed in coprime sequences, $|x|$ does the absolute value of a number x, $\|x\|$ does the order of an element x % M, $\|S\|$ does the size of a set S, and gcd(a, b) represents the greatest common divisor of two integers. Without ambiguity, "% M" is usually omitted in expressions.

2 Several Definitions

The following definitions lay the stone foundation for the new public key encryption scheme.

2.1 A Coprime Sequence

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

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

For example, {13, 2, 23, 11, 17, 19, 21, 15} and {29, 7, 11, 23, 19, 13, 5, 17} are two coprime sequences separately.

Property 1: Let $\{A_1, \ldots, A_n\}$ be a coprime sequence. If randomly select $k \in [1, n]$ elements $A_{x_1}, \ldots, A_{x_k}$ from the sequence, then the mapping from a subset $\{A_{x_1}, \ldots, A_{x_k}\}$ to a subset product $G = \prod_{i=1}^{k} A_{x_i}$ is one-to-one, namely the mapping from $b_1 \ldots b_n$ to $G = \prod_{i=1}^{n} A_i^{b_i}$ is one-to-one, where $b_1 \ldots b_n$ is a bit string.

Refer to [1] for its proof.

2.2 A Bit Shadow

Definition 2: Let $b_1 \ldots b_n \neq 0$ be a bit string. Then $\dot{b}_i$ with $i \in [1, n]$ is called a bit shadow if it comes from such a rule: ① $\dot{b}_i = 0$ if $b_i = 0$, ② $\dot{b}_i = 1$ + the number of successive 0-bits before b_i if $b_i = 1$, or ③ $\dot{b}_i = 1$ + the number of successive 0-bits before b_i + the number of successive 0-bits after the rightmost 1-bit if b_i is the leftmost 1-bit.

Notice that ③ of this definition is slightly different from that in [1].

For example, let $b_1 \ldots b_{16}$ = 10 01 00 00 01 00 11 00 or 00 10 01 00 1100 01 00, then $\dot{b}_1 \ldots \dot{b}_{16}$ = 3003000006003100 or 0050030031000400.

Fact 1: Let $\dot{b}_1 \ldots \dot{b}_n$ be the bit shadow string of $b_1 \ldots b_n \neq 0$. Then there is $\sum_{i=1}^{n} \dot{b}_i = n$.

Proof:

According to Definition 2, every bit of $b_1 \ldots b_n$ is considered into $\sum_{i=1}^{k} \dot{b}_{x_i}$, where $k \leq n$, and $\dot{b}_{x_1}, \ldots, \dot{b}_{x_k}$ are 1-bit shadows in the string $\dot{b}_1 \ldots \dot{b}_n$, and thus there is $\sum_{i=1}^{k} \dot{b}_{x_i} = n$.

On the other hand, there is $\sum_{j=1}^{n-k} \dot{b}_{y_j} = 0$, where $\dot{b}_{y_1}, \ldots, \dot{b}_{y_{n-k}}$ are 0-bit shadows.

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

Property 2: Let $\{A_1, \ldots, A_n\}$ be a coprime sequence, and $\dot{b}_1 \ldots \dot{b}_n$ be the bit shadow string of $b_1 \ldots b_n \neq 0$. Then the mapping from $b_1 \ldots b_n$ to $G = \prod_{i=1}^{n} A_i^{\dot{b}_i}$ is one-to-one.

Proof:

Let $b_1 \ldots b_n$ and $b'_1 \ldots b'_n$ be two different nonzero bit strings, and $\dot{b}_1 \ldots \dot{b}_n$ and $\dot{b}'_1 \ldots \dot{b}'_n$ be the two corresponding bit shadow strings.

①. If $\dot{b}_1 \ldots \dot{b}_n = \dot{b}'_1 \ldots \dot{b}'_n$, then by Definition 2, there is $b_1 \ldots b_n = b'_1 \ldots b'_n$.

Again, for any arbitrary bit shadow string $\underline{b}_1 \ldots \underline{b}_n$, there always exists a preimage $b_1 \ldots b_n$, namely the mapping from $b_1 \ldots b_n$ to $\underline{b}_1 \ldots \underline{b}_n$ is surjective.

Thus, the mapping from $b_1 \ldots b_n$ to $\underline{b}_1 \ldots \underline{b}_n$ is one-to-one.

②. Obviously the mapping from $\underline{b}_1 \ldots \underline{b}_n$ to $\prod_{i=1}^{n} A_i^{\underline{b}_i}$ is surjective.

Again, presuppose that $\prod_{i=1}^{n} A_i^{\underline{b}_i} = \prod_{i=1}^{n} A_i^{\underline{b}'_i}$ for $\underline{b}_1 \ldots \underline{b}_n \neq \underline{b}'_1 \ldots \underline{b}'_n$.

Since $\{A_1, \ldots, A_n\}$ is a coprime sequence, and $A_i^{\underline{b}_i}$ either equals 1 with $\underline{b}_i = 0$ or contains the same prime factors as those of A_i with $\underline{b}_i \neq 0$, we can obtain $\underline{b}_1 \ldots \underline{b}_n = \underline{b}'_1 \ldots \underline{b}'_n$ from $\prod_{i=1}^{n} A_i^{\underline{b}_i} = \prod_{i=1}^{n} A_i^{\underline{b}'_i}$. It is in direct contradiction to $\underline{b}_1 \ldots \underline{b}_n \neq \underline{b}'_1 \ldots \underline{b}'_n$, which indicates that the mapping from $\underline{b}_1 \ldots \underline{b}_n$ to $\prod_{i=1}^{n} A_i^{\underline{b}_i}$ is injective [8].

Thus, the mapping from $\underline{b}_1 \ldots \underline{b}_n$ to $\prod_{i=1}^{n} A_i^{\underline{b}_i}$ is one-to-one.

By transitivity, the mapping from $b_1 \ldots b_n$ to $\prod_{i=1}^{n} A_i^{\underline{b}_i}$ is also one-to-one. □

2.3 A Bit-pair Shadow

It is well understood that a public key cryptosystem is mainly used for transmitting a symmetric key. Assume that $b_1 \ldots b_n$ is a symmetric key. At present, to guard against exhaustive search namely brute force attack, n should be no less than 80 [9].

To make the modulus M of the new cryptoscheme comparatively small, we will utilize the idea of a bit-pair string with 2 bits to 3 items.

In this wise, the length of a coprime sequence is changed to $3n/2$, namely $\{A_1, \ldots, A_n\}$ is substituted with $\{A_1, A_2, A_3, \ldots, A_{3n/2-2}, A_{3n/2-1}, A_{3n/2}\}$ that may be logically orderly partitioned into $n/2$ triples of which each comprises 3 elements: $A_{3j-2}, A_{3j-1}, A_{3j}$ with $j \in [1, n/2]$. Likewise, a non-coprime sequence $\{C_1, \ldots, C_n\}$ is substituted with $\{C_1, C_2, C_3, \ldots, C_{3n/2-2}, C_{3n/2-1}, C_{3n/2}\}$, where $(C_{3j-2}, C_{3j-1}, C_{3j})$ with $j \in [1, n/2]$ is acquired from $(A_{3j-2}, A_{3j-1}, A_{3j})$ and other private parameters.

Definition 3*:* Let $\{A_{3j-2}, A_{3j-1}, A_{3j} \mid j = 1, \ldots, n/2\}$ be a coprime sequence. Orderly partition a bit string $b_1 \ldots b_n$ into $n/2$ pairs $B_1, \ldots, B_{n/2}$, where B_j with $j \in [1, n/2]$ has four states: 00, 01, 10, and 11 which correspond to 1, A_{3j-2}, A_{3j-1}, and A_{3j} respectively. Then $B_1, \ldots, B_{n/2}$ is called a bit-pair string, shortly $B_1 \ldots B_{n/2}$.

Property 3*:* Let $\{A_{3j-2}, A_{3j-1}, A_{3j} \mid j = 1, \ldots, n/2\}$ be a coprime sequence, and $B_1 \ldots B_{n/2}$ be a nonzero bit-pair string. Then the mapping from $B_1 \ldots B_{n/2}$ to $G' = \prod_{i=1}^{n/2} (A_{3(i-1)+B_i})^{\lceil B_i/3 \rceil}$ with $A_0 = 1$ is one-to-one, where $\lceil B_i / 3 \rceil = 0$ or 1, and G' is called a coprime subsequence product.

Its proof is parallel to that of Property 1 in [1].

For example, let $n = 8$, $\{A_1, A_2, A_3, \ldots, A_{10}, A_{11}, A_{12}\} = \{13, 7, 23, 6, 11, 29, 17, 31, 37, 41, 19, 5\}$, and $b_1 \ldots b_8 = 11011000$, then $B_1 \ldots B_4 = 11\ 01\ 10\ 00$, and $G' = \prod_{i=1}^{8/2} (A_{3(i-1)+B_i})^{\lceil B_i/3 \rceil} = A_3 A_4 A_8 1 = 23 \times 6 \times 3\,1 \times 1 = 4278$.

Definition 4*:* Let $B_1 \ldots B_{n/2}$ be a nonzero bit-pair string. Then $\underline{B}_i$ with $i \in [1, n/2]$ is called a bit-pair shadow if it comes from such a rule: ① $\underline{B}_i = 0$ if $B_i = 00$, ② $\underline{B}_i = 1$ + the number of successive 00-pairs before B_i if $B_i \neq 00$, or ③ $\underline{B}_i = 1$ + the number of successive 00-pairs before B_i + the number of successive 00-pairs after the rightmost non-00-pair if B_i is the leftmost non-00-pair.

For example, let $n = 16$, and $B_1 \ldots B_8 = 1001000001001100$ or 0010010011000100, then $\underline{B}_1 \ldots \underline{B}_8 = 21003020$ or 03102020.

Fact 2: Let $\underline{B}_1 \ldots \underline{B}_{n/2}$ be the bit-pair shadow string of $B_1 \ldots B_{n/2} \neq 0$. Then there is $\sum_{i=1}^{n/2} \underline{B}_i = n/2$.

Proof:

According to Definition 4, every pair of $B_1 \ldots B_{n/2}$ is considered into $\sum_{i=1}^{k} \underline{B}_{x_i}$, where $k \leq n/2$, and $\underline{B}_{x_1}, \ldots, \underline{B}_{x_k}$ are non-00-pair shadows in the string $\underline{B}_1 \ldots \underline{B}_{n/2}$, and thus there is $\sum_{i=1}^{k} \underline{B}_{x_i} = n/2$.

On the other hand, there is $\sum_{j=1}^{n/2-k} \underline{B}_{y_j} = 0$, where $\underline{B}_{y_1}, \ldots, \underline{B}_{y_{n/2-k}}$ are 00-pair shadows.

In total, there is $\sum_{i=1}^{n/2} \underline{B}_i = n/2$. □

Property 4*:* Let $\{A_{3j-2}, A_{3j-1}, A_{3j} \mid j = 1, \ldots, n/2\}$ be a coprime sequence, and $\underline{B}_1 \ldots \underline{B}_{n/2}$ be the bit-pair shadow string of $B_1 \ldots B_{n/2} \neq 0$. Then the mapping from $B_1 \ldots B_{n/2}$ to $G = \prod_{i=1}^{n/2} (A_{3(i-1)+B_i})^{\underline{B}_i}$ with $A_0 = 1$ is one-to-one, where G is called an anomalous coprime subsequence product.

Its proof is parallel to that of Property 2 in Section 2.2.

For example, let $n = 8$, $\{A_1, A_2, A_3, \ldots, A_{10}, A_{11}, A_{12}\} = \{13, 7, 23, 6, 11, 29, 17, 31, 37, 41, 19, 5\}$,

and $B_1 \ldots B_4$ = 11 01 10 00, then $\beta_1 \ldots \beta_4 = 2110$, and $G = \prod_{i=1}^{8/2} (A_{3(i-1)+B_i})^{\beta_i} = (A_3)^2 A_4 A_8 1 = 23^2 \times 6 \times 31 \times 1 = 98394$.

Property 3 and 4 manifest that G' or G may act as a component of a trapdoor function under bit-pair string circumstances.

2.4 A Lever Function

Definition 5*:* The secret parameter $\ell(i)$ in the key transform of a public key cryptoscheme is called a lever function, if it has the following features:

- $\ell(.)$ is an injection from the domain $\{1, \ldots, n\}$ to the codomain $\Omega \subset \{5, \ldots, \bar{M}\}$, where $\bar{M}$ is large;
- the mapping between i and $\ell(i)$ is established randomly without an analytical expression;
- an attacker has to be faced with all the arrangements of n elements in Ω when extracting a related private key from a public key;
- the owner of a private key only needs to consider the accumulative sum of n elements in Ω when recovering a related plaintext from a ciphertext.

The latter two points manifest that if n is large enough, it is infeasible for the attacker to search all the permutations of elements in Ω exhaustively while the decryption of a normal ciphertext is feasible in some time being polynomial in n. Thus, there are the large amount of calculation on $\ell(.)$ at "a public terminal", and the small amount of calculation on $\ell(.)$ at "a private terminal".

Notice that ① in modular $\bar{M}$ arithmetic, $-x$ represents $\bar{M} - x$; ② the number of elements of Ω is not less than $\tilde{n}$; ③ considering the speed of decryption, the absolute values of all the elements should be comparatively small; ④ the lower limit 5 will make seeking the root W from $W^{\ell(i)} \equiv A_i^{-1} C_i$ (% M) face an unsolvable Galois group when the value of $A_i \le 1201$ is guessed [10].

Concretely to the new cryptoscheme, $\ell(i)$ in the key transform $C_i \equiv (A_i W^{\ell(i)})^{\delta}$ (% M) with $i \in [1, n]$ is an exponent.

Property 5 **(Indeterminacy of $\ell(.)$)**: Let $\delta = 1$ and $C_i \equiv (A_i W^{\ell(i)})^{\delta}$ (% M) with $\ell(i) \in \Omega = \{5, 6, \ldots, n + 4\}$ and $A_i \in \Lambda = \{2, 3, \ldots, \dot{P} \mid \dot{P} \le 1201\}$ for $i = 1, \ldots, n$. Then $\forall W (\|W\| \ne \bar{M}) \in (1, \bar{M})$ and $\forall x, y, z (x \ne y \ne z) \in [1, n]$,

① when $\ell(x) + \ell(y) = \ell(z)$, there is $\ell(x) + \|W\| + \ell(y) + \|W\| \ne \ell(z) + \|W\|$ (% $\bar{M}$);

② when $\ell(x) + \ell(y) \ne \ell(z)$, there always exist

$$C_x \equiv A'_x W'^{\ell'(x)} \text{ (\% } M\text{)}, \; C_y \equiv A'_y W'^{\ell'(y)} \text{ (\% } M\text{)}, \text{ and } C_z \equiv A'_z W'^{\ell'(z)} \text{ (\% } M\text{)}$$

such that $\ell'(x) + \ell'(y) \equiv \ell'(z)$ (% $\bar{M}$) with $A'_z \le \dot{P}$.

Refer to [1] for its proof.

Notice that according to the proof of Property 5 in [1], it is not difficult to understand that when $\Omega = \{5, 6, \ldots, n + 4\}$ is substituted with $\Omega = \{+/-5, +/-7, \ldots, +/-(2n + 3)\}$, where "+/–" means the selection of the "+" or "–" sign, Property 5 still holds.

3 Design of the New Cryptoscheme

Due to $L_{\dot{p}}[1/3, 1.923] = 2^{80}$ with $\dot{p}$ prime and $\lceil \lg \dot{p} \rceil = 1024$ [11], the shortest bit-length of a plaintext block should be 80 which makes the time complexity of an exhaustive search attack on the plaintext block reach 2^{80}.

In the new scheme, to acquire provable semantical security, 16 random bits are appended the terminal of a plaintext block of n bits when it is encrypted.

Let $\tilde{n} = n + 16$ with n = 80, 96, or 112.

Additionally, two adjacent bits are orderly treated as a unit, namely a bit-pair string $B_1 \ldots B_{\tilde{n}/2}$ is used to represent a plaintext block $b_1 \ldots b_{\tilde{n}} \ne 0$.

3.1 Key Generation Algorithm

Considering decryption speed, the absolute values of elements of Ω should be as small as possible, and every three successive elements of Ω should be in a triple conforming with 2 bits to 3 items.

Let $\Lambda = \{2, \ldots, \dot{P}\}$, where $\dot{P} = 937$, 991, or 1201 corresponds to $\tilde{n} = 96$, 112, or 128 separately.

Let $\tilde{t} = \lceil \lg M \rceil = 464$, 544, or 640 also corresponds to $\tilde{n} = 96$, 112, or 128 separately.

Assume that $\bar{A}_j$ is the maximum in $(A_{3j-2}, A_{3j-1}, A_{3j})$ $\forall j \in [1, \tilde{n}/2]$.

The following algorithm is generally employed by the owner of a key pair.

INPUT: the integer n; the integer $\tilde{t}$; the prime $\dot{P}$.

S1: Let $\tilde{n} \leftarrow n + 16$, $\Lambda \leftarrow \{2, \ldots, \dot{P}\}$.
Yield the first $\tilde{n}$ primes in the natural number set $p_1, \ldots, p_{\tilde{n}}$.
Yield $\Omega \leftarrow \{(+/-(6j-1), +/-(6j+1), +/-(6j+3))_P \lfloor j=1, \ldots, \tilde{n}/2\}$.

S2: Produce an odd coprime sequence $\{A_1, \ldots, A_{3\tilde{n}/2} \mid A_i \in \Lambda\}$.
Formally $\{A_1, \ldots, A_{3\tilde{n}/2}\} = \{A_{3j-2}, A_{3j-1}, A_{3j} \mid j = 1, \ldots, \tilde{n}/2\}$.
Arrange $\bar{A}_1, \ldots, \bar{A}_{\tilde{n}/2}$ to $\bar{A}_{x_1}, \ldots, \bar{A}_{x_{\tilde{n}/2}}$ in descending order.

S3: Find a prime $M > \bar{A}_{x_1}^{\tilde{n}/4+1} \prod_{i=2}^{\tilde{n}/4} \bar{A}_{x_i}$ making $\lceil \lg M \rceil = \tilde{t}$ and
$\prod_{i=1}^{k} p_i^{e_i} \mid \bar{M}$, where k meets $\prod_{i=1}^{k} e_i \geq 2^{10}$ and $p_k < \tilde{n}$.

S4: Produce pairwise distinct $(\ell(3j-2), \ell(3j-1), \ell(3j)) \in \Omega$
for $j = 1, \ldots, \tilde{n}/2$.

S5: Stochastically pick $W \in (1, \bar{M})$ making $\|W\| \geq 2^{n-20}$.
Stochastically pick $\delta \in (1, \bar{M})$ making $\gcd(\delta, \bar{M}) = 1$.

S6: Compute $C_i \leftarrow (A_i W^{\ell(i)})^{\delta} \% M$ for $i = 1, \ldots, 3\tilde{n}/2$.

OUTPUT: a public key $(\{C_1, \ldots, C_{3\tilde{n}/2}\}, M)$; a private key $(\{A_1, \ldots, A_{3\tilde{n}/2}\}, W, \delta, M)$.

The lever function $\{\ell(1), \ldots, \ell(\tilde{n}/2)\}$ is discarded but must not be divulged.

Notice that

① at S1, $\Omega = \{(+/-(6j-1), +/-(6j+1), +/-(6j+3))_P \mid j = 1, \ldots, \tilde{n}/2\}$ indicates that Ω is one of $(3!)^{\tilde{n}/2} 2^{3\tilde{n}/2}$ potential sets consisting of 3-tuple elements, where "+/–" means the selection of the "+" or "–" sign, and the subscript P means that $(+/-(6j-1), +/-(6j+1), +/-(6j+3))_P$ is a permutation of $(+/-(6j-1), +/-(6j+1), +/-(6j+3))$;

② at S2, $\gcd(A_{3i-2}, A_{3i-1}, A_{3i}) \neq 1$ ($i \in [1, \tilde{n}/2]$) is allowed — $(3^3, 3^2, 3)$ for example since only one of three elements will occur in the product G;

③ at S3, the inequation $M > \bar{A}_{x_1}^{\tilde{n}/4+1} \prod_{i=2}^{\tilde{n}/4} \bar{A}_{x_i}$ assures that a ciphertext can be decrypted correctly;

④ at S5, let $W \equiv g^{\bar{M}/F} (\% M)$, then $\|W\| = \bar{M} / \gcd(\bar{M}, \bar{M}/F)$ [10], where $F \geq 2^{n-20}$ is a factor of $\bar{M}$, and g is a generator by Algorithm 4.80 in Section 4.6 of [11].

Definition 6*:* Given $(\{C_1, \ldots, C_{3\tilde{n}/2}\}, M)$, seeking the original $(\{A_1, \ldots, A_{3\tilde{n}/2}\}, \{\ell(1), \ldots, \ell(3\tilde{n}/2)\}, W, \delta)$ from $C_i \equiv (A_i W^{\ell(i)})^{\delta} (\% M)$ with $A_i \in \Lambda = \{2, \ldots, \dot{P} \mid \dot{P} \leq 1201\}$ and $\ell(i)$ from $\Omega = \{(+/-(6j-1), +/-(6j+1), +/-(6j+3))_P \mid j = 1, \ldots, \tilde{n}/2\}$ for $i = 1, \ldots, 3\tilde{n}/2$ is referred to as a multivariate permutation problem (MPP).

3.2 Encryption Algorithm

This algorithm is employed by a person who wants to encrypt plaintexts.

INPUT: a public key $(\{C_1, \ldots, C_{3\tilde{n}/2}\}, M)$;
the bit-pair string $B_1 \ldots B_{n/2}$ of a plaintext block $b_1 \ldots b_n \neq 0$.

Notice that if the number of 00-pairs in $B_1 \ldots B_{n/2}$ is larger than $n/4$, let $b_1 \ldots b_n = \neg b_1 \ldots \neg b_n$ in order that a related ciphertext can be decrypted conforming to the constraint on M.

S1: Yield a random bit string $b_{n+1} \ldots b_{\tilde{n}}$ appended to $b_1 \ldots b_n$,
and form $B_1 \ldots B_{\tilde{n}/2}$ with the number of 00-pairs $\leq \tilde{n}/4$.

S2: Set $C_0 \leftarrow 1$, $k \leftarrow 0$, $i \leftarrow 1$, $\bar{s} \leftarrow 0$.

S3: If $B_i = 00$ then
S3.1: let $k \leftarrow k + 1$, $\dot{B}_i \leftarrow 0$
else
S3.2: let $\dot{B}_i \leftarrow k + 1$, $k \leftarrow 0$;
S3.3: if $\bar{s} = 0$ then $\bar{s} \leftarrow i$ else null.

S4: Let $i \leftarrow i + 1$.
If $i \leq \tilde{n}/2$ then goto S3.

S5: If $k \neq 0$ then let $\dot{B}_{\bar{s}} \leftarrow \dot{B}_{\bar{s}} + k$.

S6: Stochastically produce $r_1 \ldots r_{\tilde{n}/2} \in \{0, 1\}^{\tilde{n}/2}$,
and set $r_{\bar{s}} \leftarrow 1$.

S7: Compute $\bar{G} \leftarrow \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i) + \neg r_i(3(i-\dot{B}_i)+B_i)})^{\dot{B}_i} \% M$.

OUTPUT: a ciphertext $\bar{G}$.

Obviously, a different ciphertext will be outputted every time an identical plaintext is inputted repeatedly. The identical plaintext may correspond to at most $2^{\tilde{n}/4}2^{\tilde{n}-n}$ different ciphertexts because $\tilde{n}/2$ bit-pairs may be interlaced by a 00-pair and a non-00-pair, and $b_{n+1}\ldots b_{\tilde{n}}$ is produced randomly. It will take the running time of $O(\tilde{n}2^{\tilde{n}/2}2^{\tilde{n}-n}\lg^2 M)$ bit operations exhaustively to search all the possible ciphertexts of a plaintext.

Notice that a JUNA ciphertext $\prod_{i=1}^{\tilde{n}/2}(C_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{\underline{B}_i}$ (% M) is different from a Naccache-Stern ciphertext $c \equiv \prod_{i=1}^{n} v_i^{b_i}$ (% M) [12], where $v_i \equiv \acute{\iota}_i^{1/\delta}$ (% M) with $\acute{\iota}_i$ prime is a public key.

Definition 7: Given ($\{C_1, \ldots, C_{3\tilde{n}/2}\}$, M) and $\bar{G}$, seeking $B_1\ldots B_{\tilde{n}/2}$ from $\bar{G}' \equiv \prod_{i=1}^{\tilde{n}/2}(C_{3(i-1)+B_i})^{\lceil B_i/3\rceil}$ (% M) with $C_0 = 1$ is referred to as a subset product problem (SPP), where $B_1\ldots B_{\tilde{n}/2}$ is the bit-pair string of $b_1\ldots b_{\tilde{n}} \neq 0$.

Definition 8: Given ($\{C_1, \ldots, C_{3\tilde{n}/2}\}$, M) and $\bar{G}$, seeking $B_1\ldots B_{\tilde{n}/2}$ from $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2}(C_{3(i-1)+B_i})^{\underline{B}_i}$ (% M) or $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2}(C_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{\underline{B}_i}$ (% M) with $C_0 = 1$ and $r_1\ldots r_{\tilde{n}/2}$ a random bit string is referred to as an anomalous subset product problem (ASPP), where $\underline{B}_1\ldots\underline{B}_{\tilde{n}/2}$ is the bit-pair shadow string of $B_1\ldots B_{\tilde{n}/2}$ corresponding to $b_1\ldots b_{\tilde{n}} \neq 0$.

3.3 Decryption Algorithm

This algorithm is employed by a person who wants to decrypt ciphertexts.

INPUT: a private key ($\{A_1, \ldots, A_{3\tilde{n}/2}\}$, W, δ, M); a ciphertext $\bar{G}$.

It should be noted that due to $2 \mid \sum_{i=1}^{\tilde{n}/2}\underline{B}_i$ and $2 \nmid \ell(r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i))$ for $i \in [1, \tilde{n}/2]$ with $\ell(0) = 0$, $\underline{k} = \sum_{i=1}^{\tilde{n}/2}\underline{B}_i\ell(r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i))$ must be even.

S1: Compute $Z_0 \leftarrow \bar{G}^{\delta^{-1}}$ % M.
  Set $Z_1 \leftarrow Z_0$, $h \leftarrow 0$.
S2: If $2 \mid Z_h$ then do $Z_h \leftarrow Z_h W^{2(-1)^h}$ % M, goto S2.
S3: Set $B_1\ldots B_{n/2} \leftarrow 0$, $j \leftarrow 0$, $k \leftarrow 0$, $l \leftarrow 0$, $i \leftarrow 1$, $G \leftarrow Z_h$.
S4: If $(A_{3i-j})^{l+1} \mid G$ then let $l \leftarrow l + 1$, goto S4.
S5: Let $j \leftarrow j + 1$.
  If $l = 0$ and $j \leq 2$ then goto S4.
S6: If $l = 0$ then
    S6.1: let $k \leftarrow k + 1$, $i \leftarrow i + 1$
  else
    S6.2: compute $G \leftarrow G/(A_{3i-j})^l$;
    S6.3: if $k > 0$ or $l \geq i$ then
        let $B_i \leftarrow 3 - j$, $i \leftarrow i + 1$
    S6.4: else
        let $B_{i+l-1} \leftarrow 3 - j$, $i \leftarrow i + l$;
    S6.5: set $l \leftarrow 0$, $k \leftarrow 0$.
S7: If $i \leq \tilde{n}/2$ and $G \neq 1$ then set $j \leftarrow 0$, goto S4.
S8: If $G \neq 1$ then set $h \leftarrow \neg h$, do $Z_h \leftarrow Z_h W^{2(-1)^h}$ % M, goto S2.
S9: Separate $B_1\ldots B_{n/2}$ from $B_1\ldots B_{\tilde{n}/2}$.
OUTPUT: a related plaintext $B_1\ldots B_{n/2}$, namely $b_1\ldots b_n$.

Notice that only if $\bar{G}$ is a true ciphertext, can the algorithm always terminate normally, and $b_1\ldots b_n$ will be original although $r_1\ldots r_{\tilde{n}/2}$ is brought into an encryption process.

4 Correctness, Uniqueness, and Complexity

In this section, we will discuss whether a ciphertext can be decrypted correctly, a plaintext solution is unique, and a decryption process can be finished in polynomial time.

4.1 Correctness of the Decryption Algorithm

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

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

where $W \in [1, \bar{M}]$ is any arbitrary integer.

Fact 3: Let $\underline{k} = \sum_{i=1}^{\tilde{n}/2} \underline{B}_i \ell(r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)) \bmod \bar{M}$ with $\ell(0) = 0$, where $\underline{B}_1 \ldots \underline{B}_{\tilde{n}/2}$ is the bit-pair shadow string of $B_1 \ldots B_{\tilde{n}/2}$ corresponding to $b_1 \ldots b_{\tilde{n}} \neq 0$, and $r_1 \ldots r_{\tilde{n}/2}$ is a random bit string. Then $\bar{G}^{\delta^{-1}}(W^{-1})^{\underline{k}} \equiv \prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{B_i} (\bmod M)$.

Proof:

Let $b_1 \ldots b_{\tilde{n}}$, namely $B_1 \ldots B_{\tilde{n}/2}$ be a plaintext of $\tilde{n}$ bits.

Additionally, let $A_0 = 1$.

According to the key generator, the encryption algorithm, and $\sum_{i=1}^{\tilde{n}/2} \underline{B}_i = \tilde{n}/2$, there is

$$\begin{aligned}\bar{G} &\equiv \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{B_i} \\ &\equiv \prod_{i=1}^{\tilde{n}/2} ((A_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)} W^{\ell(r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i))})^{\delta})^{B_i} \\ &\equiv W^{(\sum_{i=1}^{\tilde{n}/2} B_i \ell(r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)))\delta} \prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{\delta B_i} \\ &\equiv W^{\underline{k}\delta} \prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{\delta B_i} (\bmod M).\end{aligned}$$

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

$$\begin{aligned}\bar{G}^{\delta^{-1}} &\equiv (W^{\underline{k}\delta} \prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{\delta B_i})^{\delta^{-1}} \\ &\equiv W^{\underline{k}} \prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{B_i} (\bmod M).\end{aligned}$$

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

$$\begin{aligned}\bar{G}^{\delta^{-1}}(W^{-1})^{\underline{k}} &\equiv W^{\underline{k}} \prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{B_i} (W^{-1})^{\underline{k}} \\ &\equiv \prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{B_i} \\ &\equiv G \ (\bmod M).\end{aligned}$$

Clearly, the above process also gives a method of seeking G at one time. □

Notice that in practice, $\underline{k}$ is unknowable in advance.

However, because $|\underline{k}| < \tilde{n}(2(3\tilde{n}/2)+3)/2 = 3\tilde{n}(\tilde{n}+1)/2$ is comparatively small, we may search $\underline{k}$ heuristically by multiplying W^{-2} or $W^2 \bmod M$ and judging whether $G = 1$ after it is divided exactly by some $(A_{3i-j})^l$. It is known from the decryption algorithm that the original $B_1 \ldots B_{\tilde{n}/2}$ will be acquired at the same time the condition $G = 1$ is satisfied.

4.2 Uniqueness of a Plaintext Solution

Because the public key $\{C_1, \ldots, C_{3\tilde{n}/2}\}$ is a non-coprime sequence, the mapping from $B_1 \ldots B_{\tilde{n}/2}$ to $\bar{G} = \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{B_i} \bmod M$ is theoretically many-to-one. It might possibly result in the nonuniqueness of a plaintext solution $B_1 \ldots B_{\tilde{n}/2}$ when $\bar{G}$ is being unveiled.

Suppose that a ciphertext $\bar{G}$ can be obtained respectively from two different bit-pair strings $B_1 \ldots B_{\tilde{n}/2}$ and $B'_1 \ldots B'_{\tilde{n}/2}$. Then,

$$\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{B_i} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B'_i)+\neg r_i(3(i-\underline{B}'_i)+B'_i)})^{B'_i} (\bmod M).$$

That is,

$$\begin{aligned}&\prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)} W^{\ell(r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i))})^{\delta B_i} \\ \equiv &\prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B'_i)+\neg r_i(3(i-\underline{B}'_i)+B'_i)} W^{\ell(r_i(3(i-1)+B'_i)+\neg r_i(3(i-\underline{B}'_i)+B'_i))})^{\delta B'_i} (\bmod M).\end{aligned}$$

Further, there is

$$W^{\underline{k}\delta} \prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{\delta B_i} \equiv W^{\underline{k}'\delta} \prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B'_i)+\neg r_i(3(i-\underline{B}'_i)+B'_i)})^{\delta B'_i} (\bmod M),$$

where $\underline{k} = \sum_{i=1}^{\tilde{n}/2} \underline{B}_i \ell(r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i))$, and $\underline{k}' = \sum_{i=1}^{\tilde{n}/2} \underline{B}'_i \ell(r_i(3(i-1)+B'_i)+\neg r_i(3(i-\underline{B}'_i)+B'_i)) \bmod \bar{M}$ with $\ell(0) = 0$.

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

$$W^{\underline{k}} \prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{B_i} \equiv W^{\underline{k}'} \prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B'_i)+\neg r_i(3(i-\underline{B}'_i)+B'_i)})^{B'_i} (\bmod M).$$

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

$$W^{\underline{k}-\underline{k}'} \equiv \prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B'_i)+\neg r_i(3(i-\underline{B}'_i)+B'_i)})^{B'_i} (A_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{-B_i} (\bmod M).$$

Let $\theta \equiv \prod_{i=1}^{\tilde{n}/2} (A_{r_i(3(i-1)+B'_i)+\neg r_i(3(i-\underline{B}'_i)+B'_i)})^{B'_i} (A_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\underline{B}_i)+B_i)})^{-B_i} (\bmod M)$, namely $\theta \equiv W^{\underline{k}-\underline{k}'} (\bmod M)$.

This congruence signifies when the plaintext $B_1 \ldots B_{\tilde{n}/2}$ is not unique, the value of W must be relevant to θ. The contrapositive assertion equivalent to it is that if the value of W is irrelevant to θ, $B_1 \ldots B_{\tilde{n}/2}$ will be unique. Thus, we need to consider the probability that W takes a value relevant to θ.

If an adversary tries to attack an 80-bit symmetric key through the exhaustive search, and a computer

can verify trillion values per second, then it will take 38334 years for the adversary to verify all the potential values. Hence, currently 80 bits are quite enough for the security of a symmetric key.

$B_1 \ldots B_{\tilde{n}/2}$ contains $\tilde{n}$ bits which indicates $\prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\beta_i)+B_i)})^{\beta_i}$ has $2^{\tilde{n}}$ potential values, and thus the number of potential values of θ is at most $2^{\tilde{n}} \times 2^{\tilde{n}}$. Notice that because A_1^{-1}, …, $A_{3\tilde{n}/2}^{-1}$ are not necessarily coprime, some values of θ may possibly occur repeatedly.

Because $|\underline{k} - \underline{k}'| < 3\tilde{n}(\tilde{n}+1) \le 47601 \approx 2^{16}$ with $\tilde{n} \le 128$, and W has at most 2^{16} solutions to every θ, the probability that W takes a value relevant to θ is at most $2^{16}2^{2\tilde{n}}/M$.

When $\tilde{n} \ge 96$, there is $2^{16}2^{2\tilde{n}}/M \le 2^{208}/2^{464} = 1/2^{256}$ (notice that when $\tilde{n} = 96$, 112, or 128, there is $\lceil \lg M \rceil = 464$, 544, or 640), which is close to zero. The probability will further decrease when W is a prime since the solutions to θ lean to being composite integers in the average case.

In addition, if you please, resorting to $\sum_{i=1}^{\tilde{n}/2} \beta_i = \tilde{n}/2$, may exclude some unoriginal plaintext solutions.

4.3 Running Times of the Algorithms

The running time of an algorithm is measured in the number of bit operations [11], and it has an asymptotic implication. According to [11], a modular addition will take $O(\lg M)$ bit operations (shortly, bos), and a modular multiplication will take $O(2\lg^2 M)$ bos.

4.3.1 Running Time of the Key Generator

In the key generator, the steps which exert dominant effects on running time are S5 and S6.

At S5, seeking W namely seeking g will take $O(6(\ln\ln \bar{M})\lg^3 M)$ bos [11]. For every i, S6 contains one modular power $(A_i W^{\ell(i)})^{\delta}$, where $W^{\ell(i)}$ will take $2\lg(3\tilde{n}/2)$ modular multiplications which is subject to one modular power since $\tilde{n}$ is far smaller than M. Thus, the running time of the key generator is about $O((6\ln\ln \bar{M} + 4(3\tilde{n}/2))\lg^3 M) = O(6(\ln\ln \bar{M} + \tilde{n})\lg^3 M)$ bos.

4.3.2 Running Time of the Encryption Algorithm

In the encryption algorithm, the dominant step is S7.

Due to $\sum_{i=1}^{\tilde{n}/2} \beta_i = \tilde{n}/2$, S7 contains at most $\tilde{n}/2$ modular multiplications. Thus, the running time of the encryption algorithm is $O(\tilde{n}\lg^2 M)$ bos.

4.3.3 Running Time of the Decryption Algorithm

In the decryption algorithm, the dominant step is S2 which pairs with S8 to form a loop. S1 of a modular power is subject to the loop S2 ↔ S8.

It is easy to see that the number of times of executing the loop S2 ↔ S8 which mainly contains a modular multiplication is $\underline{k} = \sum_{i=1}^{\tilde{n}/2} \beta_i \ell(r_i(3(i-1)+B_i)+\neg r_i(3(i-\beta_i)+B_i))$, where β_i is relevant to a plaintext block, and $\ell(r_i(3(i-1)+B_i)+\neg r_i(3(i-\beta_i)+B_i))$ is relevant to the set $\Omega = \{(+/-(6j-1), +/-(6j+1), +/-(6j+3))_P \mid j=1, \ldots, \tilde{n}/2\}$ which is indeterminate. Therefore, it is very difficult accurately to know the value of $\underline{k}$.

When a plaintext block $B_1 \ldots B_{\tilde{n}/2}$ contains $\tilde{n}/4$ successive 00-pairs, we may obtain the maximal value of $\underline{k}$ on condition that every $\ell(r_i(3(i-1)+B_i)+\neg r_i(3(i-\beta_i)+B_i))$ has the same operating sign.

The maximal value of $\underline{k}$ is

$$\begin{aligned}\underline{k} &= (\tilde{n}/4+1)(2(3\tilde{n}/2)+3) + (2(3\tilde{n}/2)-3) + (2(3\tilde{n}/2)-9)+\ldots+(2(3\tilde{n}/2)+3-6(\tilde{n}/4-1)) \\ &= (3/4)(\tilde{n}+4)(\tilde{n}+1) + (3/16)(3\tilde{n}+4)(\tilde{n}-4) \\ &= (3/16)\tilde{n}(7\tilde{n}+12).\end{aligned}$$

Considering that $\ell(r_i(3(i-1)+B_i)+\neg r_i(3(i-\beta_i)+B_i))$ may be positive or negative, the minimal value of $\underline{k}$ will be 0 with a suitable plaintext block $B_1 \ldots B_{\tilde{n}/2}$.

Thus, the simply expected value of $\underline{k}$ is $(3/32)\tilde{n}(7\tilde{n}+12) \approx (21/32)\tilde{n}^2$.

Again considering that W^{-2} and W^2 is multiplied alternately every time, the simply expected value of $\underline{k}$ should be $2(21/32)\tilde{n}^2 \approx \tilde{n}^2$.

In summary, the simply expected time of the decryption algorithm is $O(2\tilde{n}^2\lg^2 M)$ bos.

In practice, because the possible values of $\underline{k}$ (including the repeated) will distribute at $(3/32)\tilde{n}(7\tilde{n}+12)$ integral points 0, 2, 4, …, $(3/16)\tilde{n}(7\tilde{n}+12)$ which is the maximal possible range, and the probability that $\underline{k}$ takes large integers is comparatively small, the concrete running time of a decryption process will be far smaller than $O(2\tilde{n}^2\lg^2 M)$ bos.

5 Analysis of Security of a Private Key

In this section, we will analyze the security of the new cryptoscheme against extracting a related private key from a public key.

In cryptanalysis, we suppose that the integer factorization problem (IFP) $N = \acute{p}q$ with $\lceil \lg N \rceil < 1024$ [2], the discrete logarithm problem (DLP) $y \equiv g^x$ (% $\acute{p}$) with $\lceil \lg \acute{p} \rceil < 1024$ [13][14], and the subset sum problem of low density (SSP) $s \equiv \sum_{i=1}^{n} c_i b_i$ (% M) with $D \approx n / \lceil \lg M \rceil < 1$ and $n < \lceil \lg M \rceil < 1024$ [7] can be solved in tolerable subexponential time or in polynomial time [15].

5.1 A Property of the MPP

In the new cryptoscheme, there is the MPP $C_i \equiv (A_i W^{\ell(i)})^{\delta}$ (% M) with $A_i \in \Lambda$ and $\ell(i)$ from $\Omega = \{(+/-(6j-1), +/-(6j+1), +/-(6j+3))_{\mathrm{P}} \mid j=1, \ldots, \tilde{n}/2\}$ for $i = 1, \ldots, 3\tilde{n}/2$.

According to Definition 6, the MPP has the following property.

Property 6: The MPP $C_i \equiv (A_i W^{\ell(i)})^{\delta}$ (% M) with $A_i \in \Lambda = \{2, \ldots, \grave{P} \mid \grave{P} \le 1201\}$ and $\ell(i)$ from $\Omega = \{(+/-(6j-1), +/-(6j+1), +/-(6j+3))_{\mathrm{P}} \mid j=1, \ldots, \tilde{n}/2\}$ for $i = 1, \ldots, 3\tilde{n}/2$ is computationally at least equivalent to the DLP in the same prime field.

Refer to Section 4.1 of [1] for its proof.

Notice that the structure of the set Ω consisting of triples has no change in essence compared with the Ω in [1].

5.2 Attack by Interaction of the Key Transform Items

In the key transform $C_i \equiv (A_i W^{\ell(i)})^{\delta}$ (% M), the parameters $A_i \in \Lambda = \{2, 3, \ldots, \grave{P} \mid \grave{P} = 937, 991, \text{or } 1201\}$ and $\ell(i)$ from $\Omega = \{(+/-(6j-1), +/-(6j+1), +/-(6j+3))_{\mathrm{P}} \mid j = 1, \ldots, \tilde{n}/2\}$ seem vulnerable.

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

$\forall\, x_1, x_2, y_1, y_2 \in [1, 3\tilde{n}/2]$, assume that $\ell(x_1) + \ell(x_2) = \ell(y_1) + \ell(y_2)$.

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

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

If an adversary divines the values of A_{x_1}, A_{x_2}, A_{y_1}, $A_{y_2} \in \Lambda$, he may compute δ through a discrete logarithm in $L_M[1/3, 1.923]$ time, where $M < 2^{640}$.

However, a concrete Ω is one of $(2^3 3!)^{\tilde{n}/2}$ potential sets, indeterminate, and unknown due to $|\Omega| = \tilde{n}/2$ and $\Omega = \{(+/-(6j-1), +/-(6j+1), +/-(6j+3))_{\mathrm{P}} \mid j=1, \ldots, \tilde{n}/2\}$.

For example, assume that $\ell(x_1) + \ell(x_2) = 5 + 11$, and $\ell(y_1) + \ell(y_2) = -7 + 9$, then there is $\ell(x_1) + \ell(x_2) \ne \ell(y_1) + \ell(y_2)$. Therefore, among $\ell(1)$, …, and $\ell(3\tilde{n}/2)$, there does not necessarily exist $\ell(x_1) + \ell(x_2) = \ell(y_1) + \ell(y_2)$.

The above example illustrates that to determinate the existence of $\ell(x_1) + \ell(x_2) = \ell(y_1) + \ell(y_2)$, the adversary must first determinate the constitution of the set Ω, which will have $O((2^3 3!)^{\tilde{n}/2})$ time complexity.

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

Owing to $\lceil \lg M \rceil = 512 \text{ or } 640$, $\bar{M}$ can be factorized in tolerable subexponential time. Again owing to $\prod_{i=1}^{k} p_i^{e_i} \mid \bar{M}$ and $\prod_{i=1}^{k} e_i \ge 2^{10}$ with $p_k < \tilde{n}$, $\|W\|$ can be divined in the time of about the 2^{10} magnitude.

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

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

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

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

for $i = 1, \ldots, 3\tilde{n}/2$. Notice that $u_i \ne v_i \delta$ (% $\bar{M}$) due to $\|W\| \mid \bar{M}$.

The above congruence looks to be the MH transform [7]. Actually, $\{v_1\|W\|, \ldots, v_{3\tilde{n}/2}\|W\|\}$ is not a super increasing sequence, and moreover there is not necessarily $\lg(u_i\|W\|) = \lg \bar{M}$.

Because $v_i\|W\| \in [1, \bar{M}]$ is stochastic, the inverse δ^{-1} % $\bar{M}$ not need be close to the minimum $\bar{M}/(u_i\|W\|)$, $2\bar{M}/(u_i\|W\|)$, …, or $(u_i\|W\| - 1)\bar{M}/(u_i\|W\|)$. Namely δ^{-1} may lie at any integral position in the interval $[k\bar{M}/(u_i\|W\|), (k+1)\bar{M}/(u_i\|W\|)]$, where $k = 0, 1, \ldots, u_i\|W\| - 1$, which illustrates the accumulation points of minima do not exist. Further observing, in this case, when i traverses the

interval [2, $3\tilde{n}/2$], the number of intersections of the intervals including δ^{-1} is likely the max of ($u_2\|W\|$, …, $u_{3\tilde{n}/2}\|W\|$) which is promisingly close to $\bar{M}$. Therefore, the Shamir attack by the accumulation point of minima is fully ineffectual [16].

Even though find out δ^{-1} by the Shamir attack method, because each of v_i has $\|W\|$ solutions, the number of potential sequences $\{g^{y_1}, \ldots, g^{y_{3\tilde{n}/2}}\}$ is up to $\|W\|^{3\tilde{n}/2}$. Because of needing to verify whether $\{g^{y_1}, \ldots, g^{y_{3\tilde{n}/2}}\}$ is a coprime sequence for each different sequence $\{v_1, \ldots, v_{3\tilde{n}/2}\}$, the number of coprime sequences is in direct proportion to $\|W\|^{3\tilde{n}/2}$. Hence, the initial $\{A_1, \ldots, A_{3\tilde{n}/2}\}$ cannot be determined in subexponential time. Further, the value of W cannot be computed, and the values of $\|W\|$ and δ^{-1} cannot be verified in subexponential time, which indicates that MPP can also be resistant to the attack by the accumulation point of minima.

Additionally, an adversary may divine value of A_i in about $\|\Lambda\|$ time, where $i \in [1, 3\tilde{n}/2]$, and compute δ by $u_i\|W\| \equiv v_i\|W\|\delta$ (% $\bar{M}$).

However, because of $\|W\| \mid \bar{M}$, the equation will have $\|W\|$ solutions. Therefore, the time complexity of finding the original δ is at least

$$\begin{aligned} T_e &= (3\tilde{n}/2)\|\Lambda\|L_M[1/3, 1.923] + 2^{10}\|\Lambda\|\|W\| \\ &= 2^9(3\tilde{n})L_M[1/3, 1.923] + 2^{10}2^{10}2^{n-20} \\ &\approx 2^9(3\tilde{n})L_M[1/3, 1.923] + 2^n > 2^n. \end{aligned}$$

It is exponential in n with n = 80, 96, or 112.

5.3 Attack by a Certain Single C_i

Assume that there is only a solitary $C_i = (A_i W^{\ell(i)})^{\delta}$ % M — $i = 1$ for example, and other C_i's (i = 2, …, $3\tilde{n}/2$) are unknown for adversaries.

Through divining the values of $A_1 \in \Lambda$, $\ell(1)$ from Ω, and δ coprime to $\bar{M}$, the secrete parameters $W \in (1, \bar{M})$ can be computed. Thus, the number of possible values of W will be larger than $\|\Omega\|\|\Lambda\|(\bar{M}/\ln\bar{M}) > 2^{\tilde{n}}$, which manifests that the original (A_1, $\ell(1)$, W, δ) cannot be determined in subexponential time.

Evidently, if $g_1 \equiv A_1 W^{\ell(1)}$ (% M) is a constant, solving $C_1 = g_1^{\delta}$ % M for δ is equivalent to the DLP. Factually, g_1 is not a constant, and at present, the time complexity of seeking the original g_1, namely $A_1 W^{\ell(1)}$ will be $O(M) > O(2^{\tilde{n}})$.

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

6 Analysis of Security of a Plaintext

In this section, we will analyze the security of the new cryptoscheme against recovering a related plaintext from a ciphertext.

The security of a plaintext depends on the ASPP $\tilde{G} \equiv \prod_{i=1}^{\tilde{n}/2}(C_{r_i(3(i-1)+B_i)+\neg r_i(3(i-B_i)+B_i)})^{B_i}$ (% M) with C_0 = 1 and $r_1 \ldots r_{\tilde{n}/2}$ a random bit string, but we need also to understand the SPP $\tilde{G}' \equiv \prod_{i=1}^{\tilde{n}/2}(C_{3(i-1)+B_i})^{\lceil B_i/3 \rceil}$ (% M) with $C_0 = 1$.

Definition 9*:* Let A and B be two computational problems. A is said to reduce to B in polynomial time, written as $A \leq_T^P B$, if there is an algorithm for solving A which calls, as a subroutine, a hypothetical algorithm for solving B, and runs in polynomial time, excluding the time of the algorithm for B [11][17].

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

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

Definition 10*:* Let A and B be two computational problems. If $A \leq_T^P B$ and $B \leq_T^P A$, then A and B are said to be computationally equivalent, written as $A =_T^P B$ [11][17].

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

Definition 9 and 10 suggest a reductive proof method called polynomial time Turing reduction (PTR)

[17]. Provable security by PTR is substantially relative and asymptotic just as a one-way function is. Relative security implies that the security of a cryptosystem based on intractabilities is not absolute. Asymptotic security implies that even if a cryptosystem based on intractabilities is proven to be secure, it is practically secure only on condition that the dominant parameter is large enough.

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

Let $\hat{H}(y = f(x))$ represent the complexity or hardness of solving the problem $y = f(x)$ for x [15].

6.1 Three Properties

According to Definition 7, the SPP has the following property.

Property 7: The SPP $\bar{G}' \equiv \prod_{i=1}^{\tilde{n}/2} (C_{3(i-1)+B_i})^{\lceil B_i/3 \rceil} (\% M)$ with $C_0 = 1$ is computationally at least equivalent to the DLP in the same prime field, where $B_1 \ldots B_{\tilde{n}/2} \neq 0$ is a bit-pair string.

Proof:

For clear explanations, we extend $B_1 \ldots B_{\tilde{n}/2}$ to a bit string $b'_1 \ldots b'_{3\tilde{n}/2}$ by the following rule for $i = 1, \ldots, \tilde{n}/2$:

① when $B_i = 0$, let $b'_{3(i-1)+1} = b'_{3(i-1)+2} = b'_{3(i-1)+3} = 0$;

② when $B_i \neq 0$, let $b'_{3(i-1)+1} = b'_{3(i-1)+2} = b'_{3(i-1)+3} = 0$, and $b'_{3(i-1)+B_i} = 1$.

Hence, we have the equivalent

$$\bar{G}' \equiv \prod_{i=1}^{3\tilde{n}/2} C_i^{b'_i} (\% M).$$

The form of $\bar{G}'$ here is similar to that of $\bar{G}'$ in [1].

Especially, define $\bar{G}' \equiv \prod_{i=1}^{3\tilde{n}/2} C^{2^{3\tilde{n}/2-i} b'_i} \equiv \prod_{i=1}^{3\tilde{n}/2} (C^{2^{3\tilde{n}/2-i}})^{b'_i} (\% M)$ when $C_1 = \ldots = C_{3\tilde{n}/2} = C$.

Obviously, $\prod_{i=1}^{3\tilde{n}/2} C_i^{b'_i} = L M + \bar{G}'$. Owing to $L \in [1, \overline{M}]$, deriving the non-modular product $\prod_{i=1}^{3\tilde{n}/2} C_i^{b'_i}$ from $\bar{G}'$ is infeasible, which means inferring $b'_1 \ldots b'_{3\tilde{n}/2}$ from $\bar{G}'$ is not a factorization problem.

Assume that $\bar{O}_{\mathrm{s}}(\bar{G}', C_1, \ldots, C_{3\tilde{n}/2}, M)$ is an oracle on solving $\bar{G}' \equiv \prod_{i=1}^{3\tilde{n}/2} C_i^{b'_i} (\% M)$ for $b'_1 \ldots b'_{3\tilde{n}/2}$.

Let $y \equiv g^x (\% M)$ be of the DLP, where g is a generator of $(\mathbb{Z}_M^*, \cdot)$, and the binary form of x is $\dot{b}_1 \ldots \dot{b}_{3\tilde{n}/2}$, namely $y \equiv \prod_{i=1}^{3\tilde{n}/2} (g^{2^{3\tilde{n}/2-i}})^{\dot{b}_i} (\% M)$.

Then, by calling $\bar{O}_{\mathrm{s}}(y, g^{2^{3\tilde{n}/2-1}}, \ldots, g, M)$, $\dot{b}_1 \ldots \dot{b}_{3\tilde{n}/2}$ namely x can be found.

According to Definition 9, there is

$$\hat{H}(y \equiv g^x (\% M)) \leq_{\mathrm{T}}^{\mathrm{P}} \hat{H}(\bar{G}' \equiv \prod_{i=1}^{\tilde{n}/2} (C_{3(i-1)+B_i})^{\lceil B_i/3 \rceil} (\% M)),$$

namely the SPP is at least equivalent to the DLP in the same prime field in complexity. □

According to Definition 8, and the ASPP has the following property.

Property 8: The ASPP $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{3(i-1)+B_i})^{B_i} (\% M)$ with $C_0 = 1$ is computationally at least equivalent to the DLP in the same prime field, where $B_1 \ldots B_{\tilde{n}/2}$ is the bit-pair shadow string of $B_1 \ldots B_{\tilde{n}/2} \neq 0$.

Proof:

For clear explanations, we extend $B_1 \ldots B_{\tilde{n}/2}$ to a nonrigid shadow string $b'_1 \ldots b'_{3\tilde{n}/2}$ by the following rule for $i = 1, \ldots, \tilde{n}/2$:

① when $B_i = 0$, let $b'_{3(i-1)+1} = b'_{3(i-1)+2} = b'_{3(i-1)+3} = 0$;

② when $B_i \neq 0$, let $b'_{3(i-1)+1} = b'_{3(i-1)+2} = b'_{3(i-1)+3} = 0$, and $b'_{3(i-1)+B_i} = B_i$.

Hence, we have the equivalent

$$\bar{G} \equiv \prod_{i=1}^{3\tilde{n}/2} C_i^{b'_i} (\% M).$$

The form of $\bar{G}$ here is similar to that of $\bar{G}$ in [1].

Assume that $\bar{O}_{\mathrm{a}}(\bar{G}, C_1, \ldots, C_{3\tilde{n}/2}, M)$ is an oracle on solving $\bar{G} \equiv \prod_{i=1}^{3\tilde{n}/2} C_i^{b'_i} (\% M)$ for $b'_1 \ldots b'_{3\tilde{n}/2}$, where $b'_1 \ldots b'_{3\tilde{n}/2}$ is a nonrigid shadow string corresponding to $B_1 \ldots B_{\tilde{n}/2}$.

Especially, define $\bar{G} \equiv \prod_{i=1}^{3\tilde{n}/2} C^{\tilde{n}^{3\tilde{n}/2-i} b'_i} \equiv \prod_{i=1}^{\tilde{n}} (C^{\tilde{n}^{3\tilde{n}/2-i}})^{b'_i} (\% M)$ when $C_1 = \ldots = C_{3\tilde{n}/2} = C$. Notice that due to $b'_i \leq \tilde{n}/2$, there must be $b'_i < \tilde{n}$.

Let $\bar{G}' \equiv \prod_{i=1}^{3\tilde{n}/2} C_i^{b'_i} (\% M)$ be of the SPP, where $b'_1 \ldots b'_{3\tilde{n}/2}$ corresponds to $B_1 \ldots B_{\tilde{n}/2}$.

Because $\bar{G}' \equiv \prod_{i=1}^{3\tilde{n}/2} C_i^{b'_i} (\% M)$ and $\bar{G} \equiv \prod_{i=1}^{3\tilde{n}/2} C_i^{b'_i} (\% M)$ with $0 \leq b'_i \leq b'_i$ have the same structure, by calling $\bar{O}_{\mathrm{a}}(\bar{G}', C_1, \ldots, C_{3\tilde{n}/2}, M)$, $b'_1 \ldots b'_{3\tilde{n}/2}$ can be found.

According to Definition 9, there is $\hat{H}(\bar{G}' \equiv \prod_{i=1}^{3\tilde{n}/2} C_i^{b'_i} (\% M)) \leq_{\mathrm{T}}^{\mathrm{P}} \hat{H}(\bar{G} \equiv \prod_{i=1}^{3\tilde{n}/2} C_i^{b'_i} (\% M))$.

Further by transitivity, there is

$$\hat{H}(y \equiv g^x (\% M)) \leq_{\mathrm{T}}^{\mathrm{P}} \hat{H}(\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{3(i-1)+B_i})^{B_i} (\% M)),$$

namely the ASPP $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{3(i-1)+B_i})^{\beta_i} (\% M)$ is at least equivalent to the DLP in the same prime field in computational complexity. □

Property 9: The ASPP $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i) + \neg r_i(3(i - \beta_i) + B_i)})^{\beta_i} (\% M)$ with $C_0 = 1$ and $r_1 \ldots r_{\tilde{n}/2}$ a random bit string is computationally at least equivalent to the DLP in the same prime field, where $\beta_1 \ldots \beta_{\tilde{n}/2}$ is the bit-pair shadow string of $B_1 \ldots B_{\tilde{n}/2} \neq 0$.

Proof:

Let $r_1 \ldots r_{\tilde{n}/2} = 1 \ldots 1$, then the ASPP $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i) + \neg r_i(3(i - \beta_i) + B_i)})^{\beta_i} (\% M)$ is reduced to the ASPP $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{3(i-1)+B_i})^{\beta_i} (\% M)$.

By Property 8 and the transitivity, there exists

$$\hat{H}(y \equiv g^x (\% M)) \leq_{\mathrm{T}}^{\mathrm{p}} \hat{H}(\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i) + \neg r_i(3(i - \beta_i) + B_i)})^{\beta_i} (\% M)),$$

namely the ASPP is at least equivalent to the DLP in the same prime field in complexity. □

6.2 Resisting LLL Lattice basis Reduction

We know that after a lattice basis is reduced through the LLL algorithm, the final reduced base will contain the shortest or approximately shortest vectors, but among them does not necessarily exist the original solution to a subset sum problem because only if

① the solution vector for the SSP is the shortest,

② the shortest vector is unique in the lattice,

will the original solution vector appear in the reduced base with large probability.

In the new cryptoscheme, there are $\tilde{n} = 96$, 112, or 128 and $\lceil \lg M \rceil = 464$, 544, or 640. Under the circumstances, the DLP and IFP can be solved in tolerable subexponential time, namely the DLP and IFP cannot resist the attack of adversaries.

We first consider the ASPP $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{3(i-1)+B_i})^{\beta_i} (\% M)$.

For convenience, extend $\beta_1 \ldots \beta_{\tilde{n}/2}$ to $\beta'_1 \ldots \beta'_{3\tilde{n}/2}$ by the following rule for $i = 1, \ldots, \tilde{n}/2$:

① when $\beta_i = 0$, let $\beta'_{3(i-1)+1} = \beta'_{3(i-1)+2} = \beta'_{3(i-1)+3} = 0$;

② when $\beta_i \neq 0$, let $\beta'_{3(i-1)+1} = \beta'_{3(i-1)+2} = \beta'_{3(i-1)+3} = 0$, and $\beta'_{3(i-1)+B_i} = \beta_i$.

For example, suppose that $B_1 \ldots B_4 = 00\,10\,01\,00$, then $\beta_1 \ldots \beta_4 = 0310$, and $\beta'_1 \ldots \beta'_{12} = 000\,030\,100\,000$.

In this way, the ASPP $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{3(i-1)+B_i})^{\beta_i} (\% M)$ is converted into

$$\bar{G} \equiv \prod_{i=1}^{3\tilde{n}/2} C_i^{\beta'_i} (\% M).$$

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

Let $C_1 \equiv g^{u_1} (\% M)$, …, $C_{3\tilde{n}/2} \equiv g^{u_{3\tilde{n}/2}} (\% M)$, and $\bar{G} \equiv g^v (\% M)$.

Then, through a conversion in subexponential time, seeking $B_1 \ldots B_{\tilde{n}/2}$ from $\bar{G}$ is equivalent to seeking $\beta'_1 \ldots \beta'_{3\tilde{n}/2}$ from the congruence

$$u_1 \beta'_1 + \ldots + u_{3\tilde{n}/2} \beta'_{3\tilde{n}/2} \equiv v (\% \bar{M}), \tag{3}$$

where v may be substituted with $v + k\bar{M}$ along with $k \in [0, 3\tilde{n}/2]$ [3].

Similar to Section 1, $\{u_1, \ldots, u_{3\tilde{n}/2}\}$ is called a compact sequence due to every $\beta'_i \in [0, \tilde{n}/4 + 1]$ [4], and solving Equation (3) for $\beta'_1 \ldots \beta'_{3\tilde{n}/2}$ is called an ASSP [1].

May also convert this ASSP into a SSP through splitting u_i into bits, and thus according to $\beta'_i \in [0, \tilde{n}/4 + 1]$, the density of the related ASSP knapsack is defined as

$$\begin{aligned} D &= \textstyle\sum_{i=1}^{3\tilde{n}/2} \lceil \lg(\tilde{n}/4 + 1) \rceil / \lceil \lg M \rceil \\ &= (3\tilde{n}/2) \lceil \lg(\tilde{n}/4 + 1) \rceil / \lceil \lg M \rceil. \end{aligned}$$

Namely,

$$D = 3\tilde{n} \lceil \lg(\tilde{n}/4 + 1) \rceil / (2 \lceil \lg M \rceil). \tag{4}$$

which is slightly different from Formula (2).

Concretely speaking, in the new cryptoscheme, there are

$D = 144 \times 5 / 464 \approx 1.5517 > 1$ for $\tilde{n} = 96$ and $\lceil \lg M \rceil = 464$;

$D = 168 \times 5 / 544 \approx 1.5441 > 1$ for $\tilde{n} = 112$ and $\lceil \lg M \rceil = 544$;

$D = 192 \times 6 / 640 \approx 1.8000 > 1$ for $\tilde{n} = 128$ and $\lceil \lg M \rceil = 640$.

Therefore, Equation (3) does represent an ASSP of high density, which indicates that many different subsets will have the same sum, and probability that the original solution vector will occur in the final reduced lattice basis is nearly zeroth. Meanwhile, our experiment demonstrates that the original solution vector does not occur in the final reduced base [18].

Because $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{3(i-1)+B_i})^{B_i}$ (% M) is only a special case of $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i) + \neg r_i(3(i - B_i) + B_i)})^{B_i}$ (% M), the latter is also able to resist the LLL lattice basis reduction.

6.3 Avoiding Meet-in-the-middle Attack

Meet-in-the-middle dichotomy was first developed in 1977 [19]. Section 3.10 of [11] puts forward a meet-in-the-middle attack on a subset sum problem.

INPUT: a set of positive integers $\{c_1, \ldots, c_n\}$ and a positive integer s.

S1: Set $t \leftarrow \lfloor n / 2 \rfloor$.

S2: Construct a table with entries $(\sum_{i=1}^{t} c_i b_i, (b_1, \ldots, b_t))$ for $(b_1, \ldots, b_t) \in (\mathbb{Z}_2)^t$.
Sort this table by the first component.

S3: For each $(b_{t+1}, b_{t+2}, \ldots, b_n) \in (\mathbb{Z}_2)^{n-t}$, do the following:

S3.1: Compute $r = s - \sum_{i=t+1}^{n} c_i b_i$ and check, using a binary search,
whether r is the first component of some entry in the table;

S3.2: If $r = \sum_{i=1}^{t} c_i b_i$, then return (a solution is $(b_1, \ldots, b_n)$).

S4: Return (no solution exists).

OUTPUT: $b_i \in \{0, 1\}$, $1 \le i \le n$, such that $\sum_{i=1}^{n} c_i b_i = s$, provided such b_i exist.

It is not difficult to understand that the time complexity of the above algorithm is $O(n2^{n/2})$.

Likewise, currently the versatile meet-in-the-middle dichotomy may be used to assault the ASSP $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i) + \neg r_i(3(i - B_i) + B_i)})^{B_i}$ (% M) with entries $(\prod_{i=1}^{\tilde{n}/4} (C_{r_i(3(i-1)+B_i) + \neg r_i(3(i - B_i) + B_i)})^{B_i}$, $(r_1, \ldots, r_{\tilde{n}/4})$, $(B_1, \ldots, B_{\tilde{n}/4}))$ for $(r_1, \ldots, r_{\tilde{n}/4}) \in \{0, 1\}^{\tilde{n}/4}$ and $(B_1, \ldots, B_{\tilde{n}/4}) \in \{00, 01, 10, 11\}^{\tilde{n}/4}$ when $B_{\tilde{n}/4} \ne 00$ and $B_{\tilde{n}/2} \ne 00$ which occurs with probability 9 / 16 = 0.5625. Obviously, the random bit string $r_1 \ldots r_{\tilde{n}/4}$ extends the scope of exhaustive search. Further, It is easy to see that the running time of this attack task is $O(\tilde{n}2^{\tilde{n}/2}2^{\tilde{n}/4}\lg^2 M) = O(\tilde{n}2^{3\tilde{n}/4}\lg^2 M)$ bit operations.

Concretely speaking,

when $\tilde{n} = 96$ namely $n = 80$ with $\lceil \lg M \rceil = 464$, $\mathcal{T}_m = 2^7 2^{3 \times 96/4} (2^9)^2 = 2^{97} > 2^{80}$ bos;

when $\tilde{n} = 112$ namely $n = 96$ with $\lceil \lg M \rceil = 544$, $\mathcal{T}_m = 2^7 2^{3 \times 112/4} (2^{10})^2 = 2^{111} > 2^{96}$ bos;

when $\tilde{n} = 128$ namely $n = 112$ with $\lceil \lg M \rceil = 640$, $\mathcal{T}_m = 2^8 2^{3 \times 128/4} (2^{10})^2 = 2^{124} > 2^{112}$ bos.

Therefore, the new cryptoscheme can resist the meet-in-the-middle attack.

6.4 Avoiding Adaptive-chosen-ciphertext Attack

Most of public key cryptoschemes may probably be faced with adaptive-chosen-ciphertext attack [20]. However, It is lucky the Cramer-Shoup asymmetric encryption scheme is very indistinguishable and nonmalleable [21], and proven to be secure against the adaptive-chosen-ciphertext attack under the cryptographic assumptions [22]. So is the OAEP+ scheme [23].

6.4.1 Indistinguishability of Ciphertexts

In the encryption process of a JUNA plaintext, ①a random padding string of $\tilde{n} - n$ bits is appended to the terminal of the JUNA plaintext, which changes the original plaintext to an extended plaintext, and ② a random permutation string of $\tilde{n}/2$ bits is introduced into the arrangement of bit-pairs of the extended plaintext, which is equivalent to the thing that the order of triple items of a public key is varied along with every encryption.

Due to the interlacement of 00-pairs and non-00-pairs and the randomicity of bit string generation, the padding string and the permutation string make an identical original plaintext be able to correspond to at most $2^{\tilde{n}/4}2^{\tilde{n}-n}$, which is exponential in n, different ciphertexts. It will take the running time of $O(\tilde{n}2^{\tilde{n}/2}2^{\tilde{n}-n}\lg^2 M)$ bit operations exhaustively to search all the possible ciphertexts of an original plaintext. Therefore, the correspondence between any arbitrary ciphertext and a related original plaintext are indistinguishable in subexponential time.

Concretely speaking, the running time of searching all the ciphertexts of an original plaintext is

$\mathcal{T}_s = (96)2^{96/2}2^{96-80}(464)^2 \approx 2^{88} > 2^{80}$ for $n = 80$, $\tilde{n} = 96$, and $\lceil \lg M \rceil = 464$;

$\mathcal{T}_s = (112)2^{112/2}2^{112-96}(544)^2 \approx 2^{98} > 2^{96}$ for $n = 96$, $\tilde{n} = 112$, and $\lceil \lg M \rceil = 544$;

$\mathcal{T}_s = (128)2^{128/2}2^{128-112}(644)^2 \approx 2^{108} \approx 2^{112}$ for $n = 112$, $\tilde{n} = 128$, and $\lceil \lg M \rceil = 640$.

6.4.2 Nonmalleability of Ciphertexts

An encryption scheme is said to be malleable if it is possible for an adversary to transform a ciphertext into another ciphertext which reverts to a related plaintext. That is, given an ciphertext of a plaintext $\underline{m}$, it is possible to generate another ciphertext which reverts to $f(\underline{m})$ without necessarily knowing or learning $\underline{m}$, where f is a known function [21].

By way of example, let a RSA ciphertext $\bar{C} = \underline{m}^e \% N$, then $z^e \bar{C} = (z\underline{m})^e \% N$ is a malleation of $\bar{C}$ which reverts to $f(\underline{m}) = z\underline{m} \% N$. Again let an ElGamal ciphertext $\bar{C} = (g^r, \underline{m}y^r \% \dot{p})$, then $(g^r, z\underline{m}y^r \% \dot{p})$ is a malleation of $(g^r, \underline{m}y^r \% \dot{p})$ which reverts to $f(\underline{m}) = z\underline{m} \% \dot{p}$.

In the new cryptoscheme, there is the ciphertext $\bar{G} = f'(\dot{B}) = \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i) + \neg r_i(3(i-\bar{B}_i)+B_i)})^{\bar{B}_i} \% M$ which takes a bit-pair as an operation unit, where $\dot{B} = B_1 \ldots B_{\tilde{n}/2}$ is a related extended plaintext, and thus, evidently the plaintext function $f(\dot{B}) = z\dot{B} \% M$ is not suitable for $\bar{G}$. Again considering that $\dot{B} = B_1 \ldots B_{\tilde{n}/2}$ occurs in the subscript of $\{C_1, \ldots, C_{3\tilde{n}/2}\}$, and moreover is relevant to the random bit string $r_1 \ldots r_{\tilde{n}/2}$ that can be guessed only in exponential time, it is impossible to exist other $f(\dot{B})$ which corresponds to a malleation of $f'(\dot{B})$ namely $\bar{G}$, where f is such a function of which the inverse can be computed in subexponential time.

6.4.3 Proof of the Semantical Security

If the security requirement of a cryptoscheme can be stated formally in an antagonistic model, as opposed to heuristically, with clear assumptions that certain computational problems are intractable, and an adversary has access to he algorithms of the cryptoscheme as well as enough computational resources, the cryptoscheme possesses provable security [24][25].

Definition 11*:* A cryptoscheme is said to be semantically secure if an adversary who knows the encryption algorithm of the cryptoscheme and is in possession of a ciphertext is unable to determine any information about the related plaintext [24].

It is subsequently demonstrated that semantic security is equivalent to another definition of security called ciphertext indistinguishability [26]. If a cryptoscheme has the property of indistinguishability, then an adversary will be unable to distinguish a pair of ciphertexts based on the two plaintexts encrypted by a challenger.

A chosen plaintext attack (CPA) is an attack model for cryptanalysis which presumes that the attacker has the capability to choose arbitrary plaintexts to be encrypted and obtain the corresponding ciphertexts that are expected to decrease the security of an encryption scheme [25].

Definition 12*:* A cryptoscheme is said to be IND-CPA (indistinguishable under chosen plaintext attack), namely semantically secure against chosen plaintext attack, if the adversary cannot determine which of the two plaintexts was chosen by a challenger, with probability significantly greater than $1/2$, where $1/2$ means the success rate of random guessing [25][27].

For a probabilistic asymmetric cryptoscheme based on computational security, indistinguishability under chosen plaintext attack is illuminated by a game between an adversary and a challenger, where the adversary is regarded as a probabilistic polynomial time Turing machine, which means that it must complete the game and output a guess within a polynomial number of operation steps.

Notice that for the JUNA cryptoscheme, the adversary may be also regarded as a probabilistic subexponential time Turing machine since no subexponential time solution to the MPP or ASPP is found so far.

Theorem 1*:* The JUNA cryptoscheme is semantically secure against chosen plaintext attack on the assumption that the MPP and ASPP cannot be solved in subexponential time.

Proof:

Let $E(k_p, \underline{m})$ represents the encryption of a message (plaintext) $\underline{m}$ under the public key k_p.

A game between an adversary and a challenger is given as follows.

① The challenger calls the key generation algorithm with the parameters n, $\tilde{t}$, and $\dot{P}$, obtains a key pair (k_p, k_s), publishes $k_p = (\{C_1, \ldots, C_{3\tilde{n}/2}\}, M)$ to the adversary, and retains k_s for himself.

② The adversary may perform any number of encryptions or other compatible operations.

③ Eventually, the adversary chooses any two distinct n-bit plaintexts $(\underline{m}_0, \underline{m}_1)$, and submits them to the challenger.

④ The challenger selects a bit $x \in \{0, 1\}$ uniformly at random, and sends the challenge ciphertext $\bar{G} = E(k_p, \underline{m}_x) = \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i) + \neg r_i(3(i-\bar{B}_i)+B_i)})^{\bar{B}_i} \% M$ back to the adversary, where $\underline{m}_x = b_1 \ldots b_n$.

⑤ The adversary is free to perform any number of additional computations or encryptions. Finally, it outputs a guess for the value of x. Therefore need to analyze the probability of hitting x.

Because the MPP and ASPP have no subexponential time solutions, neither can the adversary decrypt $\bar{G}$ for $\underline{m}_x$ with a private key, nor can directly solve $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\bar{B}_i)+B_i)})^{B_i}$ (% M) for $\underline{m}_x$ (= $b_1 \ldots b_n = B_1 \ldots B_{n/2}$).

It is known from the encryption algorithm that an identical plaintext may correspond to at most $2^{\tilde{n}/4}2^{\tilde{n}-n}$ different ciphertexts, where $\tilde{n} = n + 16$, and it will need the running time of $O(\tilde{n}2^{\tilde{n}/2}2^{\tilde{n}-n}\lg^2 M)$ bit operations to verify all the possible ciphertexts of a plaintext. Thus the probability that the adversary hits x with guessing is only $1/2+1/2^{\tilde{n}/4}2^{\tilde{n}-n}$, where $2^{\tilde{n}/4}2^{\tilde{n}-n}$ is exponential in n, which means that $1/2^{\tilde{n}/4}2^{\tilde{n}-n}$ is a negligible function of n, and for every (nonzero) polynomial function $poly(n)$ (notice that in the JUNA cryptoscheme, it may be also a subexponential function), there exists n_0 such that $1/2^{\tilde{n}/4}2^{\tilde{n}-n} < 1 / poly(n)$ for all $n > n_0$.

In summary, the JUNA cryptoscheme is semantically secure, namely IND-CPA. □

7 Conclusion

In the paper, a new public key cryptoscheme which includes the key generator, encryption algorithm, and decryption algorithm is proposed.

The new cryptoscheme builds its security firmly on two intractabilities: the MPP $C_i = (A_i W^{\ell(i)})^{\delta}$ % M with $A_i \in \Lambda$ and $\ell(i)$ from Ω and the ASPP $\bar{G} \equiv \prod_{i=1}^{\tilde{n}/2} (C_{r_i(3(i-1)+B_i)+\neg r_i(3(i-\bar{B}_i)+B_i)})^{B_i}$ (% M) to which no subexponential time solutions are found, and there exist only exponential time solutions so far [28], utilizes a bit-pair string to decrease the bit-length of the modulus M, exploits a bit-pair shadow string to guard against the LLL lattice basis reduction attack, and adopts the approaches of introducing a random bit string into an encryption and appending a random bit string to a plaintext to avoiding the adaptive-chosen-ciphertext attack and the meet-in-the-middle dichotomy.

As $\tilde{n}$ = 96, 112, or 128, there exists $\lceil \lg M \rceil$ = 464, 544, or 640, which assures that when a JUNA ciphertext $\bar{G}$ with $r_1 \ldots r_{\tilde{n}/2}$ = 1…1 is converted into an ASSP through a discrete logarithm, the density of a related ASSP knapsack is pretty high, and larger than 1.

There always exists contradiction between time and security, so does between space and security, and so does between time and space. We attempt to find a balance which is none other than a delicate thing among time, space, and security.

Acknowledgment

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

The authors also would like to thank the Professors Dingyi Pei, Jie Wang, Ronald L. Rivest, Moti Yung, Adi Shamir, Dingzhu Du, Mulan Liu, Huanguo Zhang, Dengguo Feng, Yixian Yang, Hanliang Xu, Xuejia Lai, Yongfei Han, Yupu Hu, Dongdai Lin, Chuankun Wu, Rongquan Feng, Ping Luo, Jianfeng Ma, Lusheng Chen, Tao Xie, Wenbao Han, Bogang Lin, Lequan Min, Qibin Zhai, Hong Zhu, Renji Tao, Zhiying Wang, Quanyuan Wu, and Zhichang Qi for their important counsel, suggestions, and corrections.

References

[1] S. Su and S. Lü, A Public Key Cryptosystem Based on Three New Provable Problems, *Theoretical Computer Science*, vol. 426-427, Apr. 2012, pp. 91-117.

[2] 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(2), 1978, pp. 120-126.

[3] V. Niemi, A New Trapdoor in Knapsacks, *Proc. of Advances in Cryptology: EUROCRYPT '90*, LNCS 473, Springer-Verlag, Berlin, 1991, pp. 405-411.

[4] G. Orton, A Multiple-Iterated Trapdoor for Dense Compact Knapsacks, *Proc. of Advance in Cryptology: EUROCRYPT '94*, Springer-Verlag, 1994, pp. 112-130.

[5] E. F. Brickell, Solving Low Density Knapsacks, *Proc. of Advance in Cryptology: CRYPTO '83*, Plenum Press, 1984, pp. 25-37.

[6] M. J. Coster, A. Joux, B. A. LaMacchia etc, Improved Low-Density Subset Sum Algorithms, *Computational Complexity*, vol. 2(2), 1992, pp. 111-128.

[7] R. C. Merkle and M. E. Hellman, Hiding information and Signatures in Trapdoor Knapsacks, *IEEE Transactions on Information Theory*, vol. 24(5), 1978, pp. 525-530.

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

[9] L. Fibíková and J. Vyskoč, Practical Cryptography - The Key Size Problem: PGP after Years, *http://www.vaf.sk/download/keysize.pdf*, Dec. 2001.

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

[11] A. J. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, London, UK, 2001, ch. 2, 3, 8.

[12] D. Naccache and J. Stern, A new public key cryptosystem, *Proc. of Advances in Cryptology: EUROCRYPT '97*, Springer-Verlag, 1997, pp. 27-36.
[13] T. ElGamal, A Public-key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, *IEEE Transactions on Information Theory*, vol. 31(4), 1985, pp. 469-472.
[14] I. F. Blake, G. Seroussi, and N. P. Smart, *Elliptic Curves in Cryptography*, Cambridge University Press, Cambridge, UK, 1999.
[15] M. Davis, *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions*, Dover Publications, Mineola, 2004.
[16] A. Shamir, A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem, *Proc. of the 23th IEEE Symposium on the Foundations of Computer Science*, IEEE, 1982, pp. 145-152.
[17] D. Z. Du and K. Ko, *Theory of Computational Complexity*, John Wiley & Sons, New York, 2000, ch. 2, 3.
[18] T. Li and S. Su, Analysis of Success Rate of Attacking Knapsacks from JUNA Cryptosystem by LLL Lattice Basis Reduction, *Proc. Of Computational Intelligence and Security (CIS) 2013*, IEEE Computer, Dec. 2013, pp. 454-458.
[19] W. Diffie and M. E. Hellman, Exhaustive Cryptanalysis of the NBS Data Encryption Standard, *Computer*, vol. 10 (6), 1977, pp. 74-84.
[20] D. Bleichenbacher, Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1, *Proc. of Advance in Cryptology: Crypto '98*, Springer-Verlag, 1998, pp. 1-12.
[21] D. Dolev, C. Dwork, and M. Naor, Nonmalleable Cryptography, *SIAM Journal on Computing*, vol. 30(2), 2000, pp. 391-437.
[22] R. Cramer and V. Shoup, A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack, *Proc. of Advance in Cryptology: Crypto '98*, Springer-Verlag, 1998, pp. 13-25.
[23] V. Shoup, OAEP Reconsidered, *Proc. of Advance in Cryptology: Crypto '01*, Springer-Verlag, 2001, pp. 239-259.
[24] S. Goldwasser and S. Micali, Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information, *Proc. Of the 14th Annual ACM Symposium on Theory of Computing*, ACM, 1982, pp. 365-377.
[26] S. Goldwasser and S. Micali, Probabilistic Encryption, *Journal of Computer and System Sciences*, vol.28, 1984, pp. 270-299.
[25] M. Bellare, Practice-oriented Provable Security, *Proc. of First International Workshop on Information Security (ISW 97)*, LNCS 1396, Springer, 1998, pp. 221-231.
[27] J. Katz and Y. Lindell, *Introduction to Modern Cryptography: Principles and Protocols*, Chapman & Hall / CRC, 2007, ch. 1, 5.
[28] S. Su and S. Lü, REESSE1+ · Reward · Proof by Experiment · A New Approach to Proof of P != NP, *Cornell University Library*, http://arxiv.org/pdf/0908.0482, Aug. 2009 (revised Aug. 2014).