

A Public Key Cryptoscheme Using the Bit-pair Method^{*}

Shenghui Su^{1,2}, Maozhi Xu³, Tao Xie⁴, and Shuwang Lü⁵

¹ College of Computers, Beijing University of Technology, Beijing 100124

² College of Info. Engineering, Yangzhou University, Yangzhou 225009

³ School of Mathematics, Peking University, Beijing 100871

⁴ School of Computers, National University of Defense Technology, Changsha 410073

⁵ Graduate School, Chinese Academy of Sciences, Beijing 100039

Abstract: The authors give the definition and property of a bit-pair shadow, and design the three algorithms of a public key cryptoscheme called JUNA which regards a bit-pair as an operation unit, and is based on the multivariate permutation problem (MPP) and the anomalous subset product problem (ASPP). Then, demonstrate that the decryption algorithm is correct, deduce the probability that a plaintext solution is nonunique is nearly zeroth, dissect the time complexities of the algorithms, and analyze the security of the cryptoscheme against extracting a private key from a public key, and recovering a plaintext from a ciphertext by LLL lattice base reduction, adaptive-chosen-ciphertext measure, and meet-in-the-middle dichotomy on the assumption that the IFP, DLP, and SSP of low density can be solved efficiently. Besides, give the conversion from the ASPP to the anomalous subset sum problem (ASSP) through a discrete logarithm. The facts show the bit-pair method increases the density of a related ASSP knapsack with $D > 1$, and decreases the length of modulus of the cryptoscheme with $\lceil \lg M \rceil = 384, 464, 544, \text{ or } 640$ corresponding to $n = 80, 96, 112, \text{ or } 128$.

Keywords: Public key cryptoscheme; Anomalous subset sum problem; Bit-pair shadow string; Compact sequence; Lever function

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 $\tilde{G} \equiv \prod_{i=1}^n C_i^{b_i} (\% M)$, where b_i is the bit shadow of a bit b_i [1], 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 $\tilde{G} \equiv g^v (\% M)$, 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 $\tilde{G} \equiv \prod_{i=1}^n C_i^{b_i} (\% M)$ for $b_1 \dots b_n$ is equivalent to solving

$$b_1 u_1 + \dots + b_n u_n \equiv v (\% \bar{M}). \quad (1)$$

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 due to every $b_i \in [0, n]$, shortly ASSP [1]. Likewise, due to every $b_i \in [0, n]$, $\{u_1, \dots, u_n\}$ is called a compact sequence [4].

It is not difficult to understand that an ASSP may be converted into a subset sum problem (SSP), and thus the density of an ASSP knapsack is defined as

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

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

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

$$D = 80 \times 7 / 696 \approx 0.8046 < 1 \text{ for } n = 80 \text{ and } \lceil \lg M \rceil = 696,$$

$$D = 96 \times 7 / 864 \approx 0.7778 < 1 \text{ for } n = 96 \text{ and } \lceil \lg M \rceil = 864,$$

$$D = 112 \times 7 / 1030 \approx 0.7612 < 1 \text{ for } n = 112 \text{ and } \lceil \lg M \rceil = 1030,$$

$$D = 128 \times 8 / 1216 \approx 0.8421 < 1 \text{ for } n = 128 \text{ and } \lceil \lg M \rceil = 1216.$$

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 ASSP since $D < 1$ only assure that the shortest vector is unique, and it cannot assure that the vector of the original solution is just the shortest vector or an approximately shortest vector occurring in the final

^{*} This research is supported by MOST with Project 2007CB311100 and 2009AA01Z441. Correspondence email: sheenway@126.com.

reduced basis.

The LLL 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 it that the original solution may possibly be found through LLL and to decrease the length of modulus of a cryptoscheme, on the basis of REESSE1+, we propose a new cryptoscheme called JUNA which treats a bit-pair as an operation unit when a bit string is encrypted in this paper.

Throughout the paper, unless otherwise specified, $n \geq 80$ is the bit-length of a plaintext block or the item-length of a sequence, the sign % means “modulo”, \bar{M} means “ $M-1$ ” with M prime, $\lg x$ denotes the logarithm of x to the base 2, \neg does the opposite value of a bit, 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, \dots, 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 $\{A_1, \dots, A_n\}$, shortly $\{A_i\}$.

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

Property 1: Let $\{A_1, \dots, A_n\}$ be a coprime sequence. If we randomly select $m \in [1, n]$ elements from $\{A_1, \dots, A_n\}$, and construct a subset $\{A_{x_1}, \dots, A_{x_m}\}$, then the subset product $G = \prod_{i=1}^m A_{x_i} = A_{x_1} \dots A_{x_m}$ is uniquely determined, namely the mapping from $\{A_{x_1}, \dots, A_{x_m}\}$ to G is one-to-one.

Refer to [1] for its proof.

2.2 A Bit Shadow

Definition 2: Let $b_1 \dots b_n \neq 0$ be a bit string. Then b_i with $i \in [1, n]$ is called a bit shadow if it comes from such a rule: ① $b_i = 0$ if $b_i = 0$, ② $b_i = 1$ + the number of successive 0-bits before b_i if $b_i = 1$, or ③ $b_i = 1$ + 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 $n = 16$, then when $b_1 \dots b_{16} = 1001000001001100$ or 0010010011000100 , $b_1 \dots b_{16} = 3003000006003100$ or 0050030031000400 .

Fact 1: Let $b_1 \dots b_n$ be the bit shadow string of $b_1 \dots b_n \neq 0$. Then there is $\sum_{i=1}^n b_i = n$.

Proof.

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

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

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

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

Proof.

Firstly, let $b_1 \dots b_n$ and $b'_1 \dots b'_n$ be two different nonzero bit strings, and $b_1 \dots b_n$ and $b'_1 \dots b'_n$ be the two corresponding bit shadow strings.

If $b_1 \dots b_n = b'_1 \dots b'_n$, then by Definition 2, there is $b_1 \dots b_n = b'_1 \dots b'_n$.

In addition, for any arbitrary bit shadow $b_1 \dots b_n$, there always exists a preimage $b_1 \dots b_n$. Thus, the mapping from $b_1 \dots b_n$ to $b_1 \dots b_n$ is one-to-one.

Secondly, obviously the mapping from $b_1 \dots b_n$ to $\prod_{i=1}^n A_i^{b_i}$ is surjective.

Presuppose that $\prod_{i=1}^n A_i^{b_i} = \prod_{i=1}^n A_i^{b'_i}$ for $b_1 \dots b_n \neq b'_1 \dots b'_n$.

Since $\{A_1, \dots, A_n\}$ is a coprime sequence, and $A_i^{b_i}$ either equals 1 with $b_i = 0$ or contains the same

prime factors as those of A_i with $b_i \neq 0$, we can obtain $b_1 \dots b_n = b'_1 \dots b'_n$ from $\prod_{i=1}^n A_i^{b_i} = \prod_{i=1}^n A_i^{b'_i}$, which is in direct contradiction to $b_1 \dots b_n \neq b'_1 \dots b'_n$.

Therefore, the mapping from $b_1 \dots b_n$ to $\prod_{i=1}^n A_i^{b_i}$ is injective [8].

In summary, the mapping from $b_1 \dots b_n$ to $\prod_{i=1}^n A_i^{b_i}$ is one-to-one, and further the mapping from $b_1 \dots b_n$ to $\prod_{i=1}^n A_i^{b_i}$ is also one-to-one. \square

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 $b_1 \dots b_n$ is a symmetric key. At present, to prevent 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 and 2 to 3.

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

Definition 3: Let $\{A_1, \dots, A_{3n/2}\}$ be a coprime sequence. Orderly partition a bit string $b_1 \dots b_n$ into $n/2$ pairs $B_1, \dots, B_{n/2}$, where B_i with $i \in [1, n/2]$ has four state: 00, 01, 10, and 11 which correspond to 1, A_{3i-2}, A_{3i-1} , and A_{3i} respectively. Then $B_1, \dots, B_{n/2}$ is called a bit-pair string, shortly $B_1 \dots B_{n/2}$.

Property 3: Let $\{A_1, \dots, A_{3n/2}\}$ be a coprime sequence, and $B_1 \dots B_{n/2}$ be a nonzero bit-pair string. Then the mapping from $B_1 \dots B_{n/2}$ 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].

Definition 4: Let $B_1 \dots B_{n/2}$ be a nonzero bit-pair string. Then \mathcal{B}_i with $i \in [1, n/2]$ is called a bit-pair shadow if it comes from such a rule: ① $\mathcal{B}_i = 0$ if $B_i = 00$, ② $\mathcal{B}_i = 1 +$ the number of successive 00-pairs before B_i if $B_i \neq 00$, or ③ $\mathcal{B}_i = 1 +$ 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 $n = 16$, then when $B_1 \dots B_8 = 1001000001001100$ or 0010010011000100 , $\mathcal{B}_1 \dots \mathcal{B}_8 = 21003020$ or 03102020 .

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

Proof.

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

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

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

Property 4: Let $\{A_1, \dots, A_{3n/2}\}$ be a coprime sequence, and $\mathcal{B}_1 \dots \mathcal{B}_{n/2}$ be the bit-pair shadow string of $B_1 \dots B_{n/2} \neq 0$. Then the mapping from $B_1 \dots B_{n/2}$ to $G = \prod_{i=1}^{n/2} (A_{3(i-1)+B_i})^{\mathcal{B}_i}$ 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 this text.

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

2.4 A Lever Function

Considering a bit-pair string, in the following text, let $\tilde{n} = 3n/2$, where $n \leq 128$.

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:

- $\ell(\cdot)$ is an injection from the domain $\{1, \dots, \tilde{n}\}$ to the codomain $\Omega \subset \{5, \dots, \overline{M}\}$, where \overline{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 time being polynomial in n . Thus, there are the large amount of calculation on $\ell(\cdot)$ at “a public terminal”, and the small amount of calculation on $\ell(\cdot)$ at “a private terminal”.

Notice that ① in modular \bar{M} arithmetic, $-x$ represents $\bar{M} - x$; ② the number of elements in Ω is not less than 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 $W^{\ell(i)} \equiv A_i^{-1} C_i \pmod{M}$ face an unsolvable Galois group when $A_i \leq 1201$ is guessed [10].

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

Property 5 (Indeterminacy of $\ell(\cdot)$): Let $\delta = 1$ and $C_i \equiv A_i W^{\ell(i)} \pmod{M}$ with $\ell(i) \in \Omega = \{5, \dots, \bar{n} + 4\}$ and $A_i \in A = \{2, \dots, \bar{P}\}$ for $i = 1, \dots, \bar{n}$, where $\bar{P} \leq 1201$. Then $\forall W \in (1, \bar{M})$, and $\forall x, y, z \in [1, \bar{n}]$ with $z \neq x, y$,

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

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

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

such that $\ell'(x) + \ell'(y) = \ell'(z) \pmod{\bar{M}}$ with $A'_z \leq \bar{P}$.

Refer to [1] for its proof.

Notice that according to the proof in [1], it is not difficult to understand that if $\Omega = \{5, \dots, \bar{n} + 4\}$ is substituted with $\Omega \subset \{\pm 5, \pm 7, \dots, \pm(2\bar{n} + 3)\} (x + y \neq 0 \forall x, y \in \Omega)$, where “ $\pm x$ ” means the coexistence of the numbers “ $+x$ ” and “ $-x$ ”, Property 5 still holds.

3 Design of the JUNA Cryptoscheme

In this new scheme, two adjacent bits are treated as a unit, namely a bit-pair string $B_1 \dots B_{n/2}$ represent a related plaintext block $b_1 \dots b_n \neq 0$.

3.1 The Key Generation Algorithm

Let $p_1, \dots, p_{\bar{n}}$ be the first \bar{n} primes in the set \mathbb{N} which can constitute a smallest coprime sequence.

Considering 2 to 3, the elements of Ω should be of 3-tuple, and again considering the promptness of decryption, the absolute values of elements of Ω should be as small as possible.

Thus, Let $\Omega \subset \{\pm 5, \pm 7, \pm 9, \dots, (\pm(2\bar{n} - 1), \pm(2\bar{n} + 1), \pm(2\bar{n} + 3))\}$ with $|x_1||x_2||x_3| \neq |y_1||y_2||y_3| \forall (x_1, x_2, x_3), (y_1, y_2, y_3) \in \Omega$, and $|\Omega_i| = n/2$. Then a concrete Ω is one of $(3!)^{n/2} 2^{\bar{n}}$ potential sets consisting of 3-tuple elements.

Additionally, let $A = \{2, \dots, \bar{P}\}$, where $\bar{P} = 863, 937, 991, \text{ or } 1201$ as $n = 80, 96, 112, \text{ or } 128$.

Assume that \bar{A}_i is the maximum in a triple $(A_{3i-2}, A_{3i-1}, A_{3i})$ for $i = 1, \dots, n/2$. Arrange $\bar{A}_1, \dots, \bar{A}_{n/2}$ in descending order, and obtain $\bar{A}_{x_1}, \dots, \bar{A}_{x_{n/2}}$.

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

S1: Randomly produce an odd coprime sequence $\{A_1, \dots, A_{\bar{n}}\}$.

S2: Find a prime $M > \bar{A}_{x_1}^{n/4+1} \prod_{i=2}^{n/4} \bar{A}_{x_i}$ making $\prod_{i=1}^k p_i^{e_i} \mid \bar{M}$,

where k meets $\prod_{i=1}^k e_i \geq 2^{10}$ and $p_k < \bar{n}/2$.

S3: Generate pairwise distinct $(\ell(3i-2), \ell(3i-1), \ell(3i))$

of which each belongs to Ω for $i = 1, \dots, n/2$.

S4: Randomly pick $\delta, W \in [1, \bar{M}]$ making $\|W\| \geq 2^{n-20}$ and $\gcd(\delta, \bar{M}) = 1$.

S5: Compute $C_i \leftarrow (A_i W^{\ell(i)})^\delta \pmod{M}$ for $i = 1, \dots, \bar{n}$.

At last, obtain a public key $(\{C_1, \dots, C_{\bar{n}}\}, M)$, and a private key $(\{A_1, \dots, A_{\bar{n}}\}, W, \delta)$. $\{\ell(i)\}$ may be discarded.

Notice that

① for seeking a fit W , let $W \equiv g^{\bar{M}/F} \pmod{M}$ since $\|W\| = \bar{M} / \gcd(\bar{M}, \bar{M}/F)$ [10], where $F \geq 2^{n-30}$ is a factor of \bar{M} , and g is a generator by Algorithm 4.80 in Section 4.6 of [11];

② $\gcd(A_{3i-2}, A_{3i-1}, A_{3i}) \neq 1$ is allowed — $(3^3, 3^2, 3)$ for example since only one of three elements will occur in G ;

③ the inequation $M > \bar{A}_{x_1}^{n/4+1} \prod_{i=2}^{n/4} \bar{A}_{x_i}$ assures that when $n = 80, 96, 112, \text{ or } 128$, there exists $\lceil \lg M \rceil = 384, 464, 544, \text{ or } 640$.

Definition 6: Seeking $\{A_i\}$, $\{\ell(i)\}$, W , δ from the key transform $C_i = (A_i W^{\ell(i)})^\delta \pmod{M}$ with $\ell(i) \in (\ell(j +$

1), $\ell(j+2), \ell(j+3)) \in \Omega$ and $A_i \in \mathcal{A}$ for $i = 1, \dots, \tilde{n}$ is referred to as a multivariate permutation problem (MPP), where $j = 3\lfloor(i-1)/3\rfloor$.

3.2 The Encryption Algorithm

Assume that $(\{C_1, \dots, C_{\tilde{n}}\}, M)$ is a public key, and $B_1 \dots B_{n/2}$ is the bit-pair string of a plaintext block $b_1 \dots b_n \neq 0$.

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

S1: Set $C_0 \leftarrow 1, k \leftarrow 0, i \leftarrow 1, s \leftarrow 0$.

S2: If $B_i = 00$ then

 let $k \leftarrow k + 1, \mathcal{B}_i \leftarrow 0$

else

 let $\mathcal{B}_i \leftarrow k + 1, k \leftarrow 0$;

 if $s \neq 0$ then $s \leftarrow i$.

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

 If $i \leq n/2$ then goto S2.

S4: If $k \neq 0$ then let $\mathcal{B}_s \leftarrow \mathcal{B}_s + k$.

S5: Compute $\tilde{\mathcal{G}} \leftarrow \prod_{i=1}^{n/2} (C_{3(i-1)+B_i})^{\mathcal{B}_i} \% M$.

At last, a related ciphertext $\tilde{\mathcal{G}}$ is obtained.

We see that a JUNA ciphertext $\tilde{\mathcal{G}} \equiv \prod_{i=1}^{n/2} (C_{3(i-1)+B_i})^{\mathcal{B}_i} (\% M)$ is different from an Naccache-Stern ciphertext $c \equiv \prod_{i=1}^n v_i^{b_i} (\% M)$ [12], where $v_i \equiv p_i^{1/\delta} (\% M)$ is a public key.

Definition 7: Let $B_1 \dots B_{n/2}$ be the bit-pair string of $b_1 \dots b_n \neq 0$. Given $\{C_1, \dots, C_{3n/2}\}, M$, and $\tilde{\mathcal{G}}$, then seeking $B_1 \dots B_{n/2}$ from $\tilde{\mathcal{G}}_1 \equiv \prod_{i=1}^{n/2} (C_{3(i-1)+B_i})^{\lceil B_i/3 \rceil} (\% M)$ with $C_0 = 1$ is referred to as a subset product problem (SPP).

Definition 8: Let $B_1 \dots B_{n/2}$ be the bit-pair string of $b_1 \dots b_n \neq 0$, and $\mathcal{B}_1 \dots \mathcal{B}_{n/2}$ be the bit-pair shadow string. Given $\{C_1, \dots, C_{3n/2}\}, M$, and $\tilde{\mathcal{G}}$, then seeking $\mathcal{B}_1 \dots \mathcal{B}_{n/2}$ from $\tilde{\mathcal{G}} \equiv \prod_{i=1}^{n/2} (C_{3(i-1)+B_i})^{\mathcal{B}_i} (\% M)$ with $C_0 = 1$ is referred to as an anomalous subset product problem (ASPP).

3.3 The Decryption Algorithm

Assume that $(\{A_1, \dots, A_{\tilde{n}}\}, W, \delta)$ is a related private key, and $\tilde{\mathcal{G}}$ is a ciphertext.

Notice that due to $\sum_{i=1}^{n/2} \mathcal{B}_i = n/2$ and $\ell(3(i-1)+B_i)$ odd (excluding $\ell(0) = 0$), $k = \sum_{i=1}^{n/2} \mathcal{B}_i \ell(3(i-1)+B_i)$ must be even.

S1: Compute $Z_0 \leftarrow \tilde{\mathcal{G}}^{\delta^{-1}} \% M$.

 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 $B_1 \dots B_{n/2} \leftarrow 0, j \leftarrow 0, k \leftarrow 0, i \leftarrow 1, s \leftarrow 0, G \leftarrow Z_h$.

S4: If $A_{3i-j}^{k+1} \mid G$ then

 do $G \leftarrow G / A_{3i-j}^{k+1}, B_i \leftarrow 3 - j, k \leftarrow 0$;

 if $s \neq 0$ then $s \leftarrow 3i - j$ else null

else

 let $j \leftarrow j + 1$;

 if $j \leq 2$ then goto S4 else let $k \leftarrow k + 1$.

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

 If $i \leq n/2$ and $G \neq 1$ then set $j \leftarrow 0$, goto S4.

S6: If $k \neq 0$ and $A_s^k \mid G$ then do $G \leftarrow G / A_s^k$.

S7: If $G \neq 1$ then

 set $h \leftarrow -h$, do $Z_h \leftarrow Z_h W^{2(-1)^h} \% M$, goto S2.

At last, the original plaintext block $B_1 \dots B_{n/2}$, namely $b_1 \dots b_n$ is recovered.

Only if $\tilde{\mathcal{G}}$ is a true ciphertext, can the algorithm terminate normally.

Notice that considering that it is possible that the number of 00-pairs in $B_1 \dots B_{n/2}$ is larger than $n/4$, a satisfactory plaintext is either $b_1 \dots b_n$ or $-b_1 \dots -b_n$.

4 Correctness, Uniqueness, and Complexity

In this section, we 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, $\forall k \in [1, \bar{M}]$, there is

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

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

Fact 3: Let $k = \sum_{i=1}^{n/2} \mathcal{B}_i \ell(3(i-1) + B_i) \% \bar{M}$ with $\ell(0) = 0$, where $\mathcal{B}_1 \dots \mathcal{B}_{n/2}$ is the bit-pair shadow string of $B_1 \dots B_{n/2}$ corresponding to $b_1 \dots b_n \neq 0$. Then $\bar{G}^{\delta^{-1}}(W^{-1})^k \equiv \prod_{i=1}^{n/2} (A_{3(i-1)+B_i})^{\mathcal{B}_i} \pmod{M}$.

Proof:

Let $b_1 \dots b_n$, namely $B_1 \dots B_{n/2}$ be an n -bit plaintext block.

Additionally, let $A_0 = 1$.

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

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

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

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

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

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

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

Notice that in practice, k is unknowable in advance.

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

4.2 Uniqueness of a Plaintext Solution

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

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

$$\bar{G} \equiv \prod_{i=1}^{n/2} (C_{3(i-1)+B_i})^{\mathcal{B}_i} \equiv \prod_{i=1}^{n/2} (C_{3(i-1)+B'_i})^{\mathcal{B}'_i} \pmod{M}.$$

That is,

$$\prod_{i=1}^{n/2} (A_{3(i-1)+B_i} W^{\ell(3(i-1)+B_i)})^{\delta \mathcal{B}_i} \equiv \prod_{i=1}^{n/2} (A_{3(i-1)+B'_i} W^{\ell(3(i-1)+B'_i)})^{\delta \mathcal{B}'_i} \pmod{M}.$$

Further, owing to $\sum_{i=1}^{n/2} \mathcal{B}_i = \sum_{i=1}^{n/2} \mathcal{B}'_i = n/2$, there is

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

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

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

$$W^k \prod_{i=1}^{n/2} (A_{3(i-1)+B_i})^{\mathcal{B}_i} \equiv W^{k'} \prod_{i=1}^{n/2} (A_{3(i-1)+B'_i})^{\mathcal{B}'_i} \pmod{M}.$$

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

$$W^{k-k'} \equiv \prod_{i=1}^{n/2} (A_{3(i-1)+B_i})^{\mathcal{B}_i} (\prod_{i=1}^{n/2} (A_{3(i-1)+B'_i})^{\mathcal{B}'_i})^{-1} \pmod{M}.$$

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

This congruence signifies when the plaintext $B_1 \dots B_{n/2}$ 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 , $B_1 \dots B_{n/2}$ 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, 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.

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

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

When $n \geq 80$, there is $2^{16} 2^{2n} / M \leq 2^{176} / 2^{384} = 1 / 2^{208}$ (notice that when $n = 80, 96, 112, \text{ or } 128$, there is $\lceil \lg M \rceil = 384, 464, 544, \text{ or } 640$), 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 $\sum_{i=1}^{n/2} B_i = n/2$, may exclude some unoriginal plaintext solutions.

4.3 Time Complexities of the Algorithms

In the paper, the running time, time complexity of an algorithm on an input 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, and a modular multiplication will take $O(2 \lg^2 M)$ bit operations.

4.3.1 Running Time of the Key Generation Algorithm

In the key generation algorithm, the step which exerts a dominant effect on the running time is S5.

For every i , S5 contains a modular power and the computing of $A_i W^{\ell(i)}$ which is equivalent to \tilde{n} modular multiplications. Thus, the running time of the key generation algorithm is $O(\tilde{n}(\tilde{n} \lg^2 M + \lg^3 M))$.

4.3.2 Running Time of the Encryption Algorithm

In the encryption algorithm, the dominant step is S5.

Due to $\sum_{i=1}^{n/2} B_i = n/2$, S5 contains $n/2$ modular multiplications. Thus, the running time of the encryption algorithm is $O(n \lg^2 M)$.

4.3.3 Running Time of the Decryption Algorithm

In the decryption algorithm, the dominant step is S2 which pairs with S7 to construct a loop.

It is easy to see that the number of times of executing the loop $S2 \leftrightarrow S7$ which mainly contain a modular multiplication is $k = \sum_{i=1}^{n/2} B_i \ell(3(i-1) + B_i)$, where B_i is relevant to a plaintext block, and $\ell(3(i-1) + B_i)$ is relevant to the set Ω which is indeterminate. Therefore, it is very difficult accurately to know the value of k .

When a plaintext block $B_1 \dots B_{n/2}$ contains $n/4$ successive 00-pairs, we may obtain the maximal value of k , where k is considered only as a positive number.

The maximal value of k is

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

Similarly, when a plaintext block $B_1 \dots B_{n/2}$ contains suitable bit-pairs, we may obtain the minimal value of k which equals 0.

In summary, the simply expected value of k is $(3/32)n(7n + 12) \approx (21/32)n^2$.

Again considering that k is possibly negative, and $W^{2^{(-1)^h}}$ is multiplied every time, the simply expected value of k should be $2(1/2)(21/32)n^2$ which is still $(21/32)n^2$.

Thus, the simply expected running time of the decryption algorithm is $O((21/32)n^2 \lg^2 M)$.

However, in practice, because 2^n k -values will distribute at $(3/32)n(7n + 12)$ integral points 0, 2, 4, ..., $(3/16)n(7n + 12)$ which is the maximal possible range, and the probability that k takes large integers is comparatively small, the concrete running time of a decryption process will be far smaller than $O((21/32)n^2 \lg^2 M)$.

5 Analysis of Security of a JUNA 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) [2], the discrete logarithm problem (DLP) [13][14], and the subset sum problem of low density (SSP) [7] can all be solved in tolerable subexponential time [15].

5.1 A Property of the MPP

In the new cryptoscheme, there is the MPP $C_i = (A_i W^{\ell(i)})^\delta \% M$ with $A_i \in \mathcal{A}$ and $\ell(i) \in (x, y, z) \in \Omega \subset \{(\pm 5, \pm 7, \pm 9), \dots, (\pm(2\bar{n}-1), \pm(2\bar{n}+1), \pm(2\bar{n}+3))\}$ for $i = 1, \dots, \bar{n}$. Notice that the structure of the set Ω consisting of triples has no change compared with that in [1] in essence.

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

Property 6: The MPP $C_i = (A_i W^{\ell(i)})^\delta \% M$ with $A_i \in \mathcal{A}$ and $\ell(i) \in (\ell(j+1), \ell(j+2), \ell(j+3)) \in \Omega$ for $i = 1, \dots, \bar{n}$ is computationally at least equivalent to DLP in the same prime field, where $j = 3\lfloor (i-1)/3 \rfloor$.

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

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 $A_i \in \mathcal{A} = \{2, 3, \dots, \bar{P} \mid \bar{P} = 863, 937, 991, \text{ or } 1201\}$ and $\ell(i) \in (x, y, z) \in \Omega \subset \{(\pm 5, \pm 7, \pm 9), \dots, (\pm(2\bar{n}-1), \pm(2\bar{n}+1), \pm(2\bar{n}+3))\}$ are vulnerable.

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

$\forall x_1, x_2, y_1, y_2 \in [1, n]$, 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

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

If an adversary divines the values of $A_{x_1}, A_{x_2}, A_{y_1}, A_{y_2} \in \mathcal{A}$, he may compute δ through a discrete logarithm in time of $L_M[1/3, 1.923]$ with $\lceil \lg \bar{M} \rceil \leq 640$.

However, due to $|\Omega| = n/2$ and $\Omega \subset \{(\pm 5, \pm 7, \pm 9), \dots, (\pm(2\bar{n}-1), \pm(2\bar{n}+1), \pm(2\bar{n}+3))\}$ with $|x_1||x_2| |x_3| \neq |y_1||y_2||y_3| \forall (x_1, x_2, x_3), (y_1, y_2, y_3) \in \Omega$, a concrete Ω is one of $(2^3 3!)^{n/2}$ potential sets consisting of triples, and indeterminate.

For example, assume that $\ell(x_1) + \ell(x_2) = 5 + 11$, and $\ell(y_1) + \ell(y_2) = -7 + 9$, then there is $\ell(x_1) + \ell(x_2) \neq \ell(y_1) + \ell(y_2)$. Therefore, among $\ell(1), \dots, \text{ and } \ell(\bar{n})$, 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 $\ell(x_1) + \ell(x_2) = \ell(y_1) + \ell(y_2)$, an adversary must first determinate the constitution of Ω , which will take the running time of $O((2^3 3!)^{n/2})$.

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

Due to $\lceil \lg \bar{M} \rceil = 384, 464, 544, \text{ or } 640$, \bar{M} can be factorized in tolerable subexponential time. Again due to $\prod_{i=1}^k p_i^{e_i} \mid \bar{M}$ and $\prod_{i=1}^k e_i \geq 2^{10}$ with $p_k < \bar{n}/2$, $\|W\|$ can be divined in the running time of about 2^{10} .

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

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

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

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

for $i = 1, \dots, \bar{n}$. Notice that $u_i \neq v_i \delta (\% \bar{M})$ owing to $\|W\| \mid \bar{M}$.

The above congruence looks to be the MH transform [7]. Actually, $\{v_1 \|W\|, \dots, v_{\bar{n}} \|W\|\}$ 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 $\delta^{-1} \% \bar{M}$ not need be close to the minimum $\bar{M}/(u_i \|W\|)$, $2\bar{M}/(u_i \|W\|)$, \dots , 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 $k = 0, 1, \dots, u_i \|W\| - 1$, which illustrates the accumulation points of minima do not exist. Further observing, in this case, when i traverses the interval $[2, \bar{n}]$, the number of intersections of the intervals including δ^{-1} is likely $\max_{2 \leq i \leq \bar{n}} \{u_i \|W\|\}$ 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 $\{g^{v_1}, \dots, g^{v_{\bar{n}}}\}$ is up to $\|W\|^{\bar{n}}$. Because of needing to verify whether $\{g^{v_1}, \dots, g^{v_{\bar{n}}}\}$ is a coprime sequence for each different sequence $\{v_1, \dots, v_{\bar{n}}\}$, the number of coprime sequences is in direct proportion to $\|W\|^{\bar{n}}$. Hence, the initial $\{A_1, \dots, A_{\bar{n}}\}$ 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 running time of about $\ell(A_i)$, where $i \in [1, \bar{n}]$, and compute δ by $u_i \|W\| \equiv v_i \|W\| \delta \pmod{\bar{M}}$. However, because of $\|W\| \mid \bar{M}$, the equation will have $\|W\|$ solutions. Therefore, the running time of finding the original δ is at least

$$\begin{aligned} F &= \bar{n}! A_i! L_M [1/3, 1.923] + 2^{10} \ell(A_i) \|W\| \\ &= \bar{n}! A_i! L_M [1/3, 1.923] + 2^{10} \ell(A_i) 2^{n-20} \\ &\approx \bar{n}! A_i! L_M [1/3, 1.923] + 2^n > 2^n. \end{aligned}$$

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

5.3 Attack by a Certain Single C_i

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

Through divining $A_1 \in \mathcal{A}$ and $\ell(1) \in \mathcal{Q}$, the parameters W and $\delta \in (1, \bar{M})$ can be computed. Thus, the number of solution $(A_1, \ell(1), W, \delta)$ will be up to $\ell(\mathcal{Q}) \ell(\mathcal{A}) \bar{M}^2 > 2^n$, which manifests that the original $(A_1, \ell(1), W, \delta)$ cannot be determined in time being subexponential in n .

Evidently, if $g_1 \equiv A_1 W^{\ell(1)} \pmod{M}$ is a constant, solving $C_1 = g_1^\delta \pmod{M}$ for δ is equivalent to DLP. Factually, g_1 is not a constant, and at present, seeking the original g_1 and δ will take at least $O(M) > O(2^n)$ steps.

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

6 Analysis of Security of a JUNA 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 JUNA plaintext depends on the ASPP $\bar{G} \equiv \prod_{i=1}^{n/2} (C_{3(i-1)+B_i})^{B_i} \pmod{M}$ with $C_0 = 1$, but we need also to understand the SPP $\bar{G}_1 \equiv \prod_{i=1}^{n/2} (C_{3(i-1)+B_i})^{\lceil B_i/3 \rceil} \pmod{M}$ with $C_0 = 1$.

6.1 Two Properties

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

Property 7: The SPP $\bar{G}_1 \equiv \prod_{i=1}^{n/2} (C_{3(i-1)+B_i})^{\lceil B_i/3 \rceil} \pmod{M}$ with $C_0 = 1$ is computationally at least equivalent to DLP in the same prime field, where $B_1 \dots B_{n/2} \neq 0$ is a bit-pair string.

Proof:

For the explanation of the problem, we extend $B_1 \dots B_{n/2}$ to $b'_1 \dots b'_{\bar{n}}$ by the following rule for $i = 1, \dots, n/2$:

- 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 $b'_{3(i-1)+B_i} = 1$.

Then, we have

$$\bar{G}_1 \equiv \prod_{i=1}^{\bar{n}} C_i^{b'_i} \pmod{M}.$$

The above form of \bar{G}_1 is similar to that of \bar{G}_1 in [1]. Therefore, the rest of this proof is parallel to the proof of Property 5 in [1]. \square

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

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

Proof:

For the explanation of the problem, we extend $B_1 \dots B_{n/2}$ to $b'_1 \dots b'_{\bar{n}}$ by the following rule for $i = 1, \dots, n/2$:

- 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 $b'_{3(i-1)+B_i} = B_i$.

Then, we have

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

The above form of \bar{G} is similar to that of \bar{G} in [1]. Therefore, the rest of this proof is parallel to the proof of Property 6 in [1]. \square

6.2 Resisting LLL Lattice Base Reduction

We know that after a lattice base 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, and

② the shortest vector is unique in the lattice,

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

In the JUNA cryptoscheme, there are $n = 80, 96, 112, \text{ or } 128$ and $\lceil \lg M \rceil = 384, 464, 544, \text{ or } 640$. Under this circumstances, DLP and IFP can be solved in tolerable subexponential time, namely DLP and IFP can not resist the attack of adversaries.

For convenience, extend $B_1 \dots B_{n/2}$ to $b'_1 \dots b'_{\tilde{n}}$ by the following rule for $i = 1, \dots, n/2$:

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 $b'_{3(i-1)+B_i} = B_i$.

For example, suppose that $B_1 \dots B_4 = 00100100$, then $B_1 \dots B_4 = 0310$, and $b'_1 \dots b'_{12} = 000\ 030\ 100\ 000$.

In this way, there is

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

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

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

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

$$u_1 b'_1 + \dots + u_{\tilde{n}} b'_{\tilde{n}} \equiv v (\% \bar{M}), \quad (3)$$

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

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

This ASSP may also be converted into a SSP, and thus according to $b'_i \in [0, n/4 + 1]$, the density of a related ASSP knapsack is defined as

$$D = \frac{\sum_{i=1}^{\tilde{n}} \lceil \lg(n/4 + 1) \rceil}{\lceil \lg M \rceil} \\ = \tilde{n} \lceil \lg(n/4 + 1) \rceil / \lceil \lg M \rceil.$$

Namely,

$$D = 3n \lceil \lg(n/4 + 1) \rceil / (2 \lceil \lg M \rceil). \quad (4)$$

which is slightly different from Formula (2).

Concretely speaking, in the JUNA cryptoscheme, there are

$$D = 120 \times 5 / 384 \approx 1.5625 > 1 \text{ for } n = 80 \text{ and } \lceil \lg M \rceil = 384,$$

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

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

$$D = 196 \times 6 / 640 \approx 1.8375 > 1 \text{ for } n = 128 \text{ and } \lceil \lg M \rceil = 640.$$

Therefore, Equation (3) represents 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 base is nearly zeroth. Meanwhile, our experiment demonstrates that the original solution vector does not occur in the final reduced base.

6.3 Avoiding Adaptive-chosen-ciphertext Attack

Most of public key cryptoschemes may probably be faced with adaptive-chosen-ciphertext attack [17]. The Cramer-Shoup asymmetric encryption scheme is the first efficient one which is extremely malleable, and proven to be secure against the adaptive-chosen-ciphertext attack using standard cryptographic assumptions [18]. In our scheme, to avoiding adaptive-chosen-ciphertext attack, we have two approaches of which each makes the algorithm be able to produce many different ciphertexts to an identical plaintext.

6.3.1 Introducing a Random Bit String into the Encryption

× The Adjusted Encryption Algorithm

Parallel to Section 3.2, assume that $(\{C_1, \dots, C_{n/2}\}, M)$ is a public key, and $B_1 \dots B_{n/2}$ is the bit-pair string of a plaintext block $b_1 \dots b_n \neq 0$.

S1: Set $C_0 \leftarrow 1, k \leftarrow 0, i \leftarrow 1, s \leftarrow 0$.

S2: If $B_i = 00$ then

 let $k \leftarrow k + 1, B_i \leftarrow 0$

 else

 let $B_i \leftarrow k + 1, k \leftarrow 0$;

 if $s \neq 0$ then $s \leftarrow i$.

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

 If $i \leq n / 2$ then goto S2.

S4: Randomly produce $r_1 \dots r_{n/2} \in \{0, 1\}^{n/2}$.

S5: $r_s \leftarrow 1$.

 If $k \neq 0$ then let $B_s \leftarrow B_s + k$.

S6: Compute $\bar{G} \leftarrow \prod_{i=1}^{n/2} (C_{r_i(3(i-1)+B_i) + -r_i(3(i-B_i) \% n/2 + B_i)})^{B_i} \% M$.

Clearly, when an identical plaintext is inputted repeatedly, a ciphertext different from one another will be returned by the above algorithm every time.

It is easily understood that contrarily a ciphertext can almost uniquely be decrypted in polynomial time in terms of Section 3.3 and 4.2.

× The Adjusted Decryption Algorithm

Parallel to Section 3.3, we redesign a corresponding decryption algorithm of which the running time is equivalent to that of the algorithm in Section 3.3.

Assume that $(\{A_1, \dots, A_{n/2}\}, W, \delta)$ is a related private key, and \bar{G} is a ciphertext.

Notice that due to $\sum_{i=1}^{n/2} B_i = n/2$ and $\ell(3(i-1)+B_i)$ odd (excluding $\ell(0) = 0$), $k = \sum_{i=1}^{n/2} B_i \ell(3(i-1)+B_i)$ 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 $B_1 \dots B_{n/2} \leftarrow 0, j \leftarrow 0, k \leftarrow 0, l \leftarrow 0, i \leftarrow 1, G \leftarrow Z_h$.

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

 let $k \leftarrow k + 1, i \leftarrow i + 1$

 else

 compute $G \leftarrow G / A_{3i-j}^l$;

 if $k > 0$ or $l \geq i$ then

 let $B_i \leftarrow 3 - j, i \leftarrow i + 1$

 else

 let $B_{i+l-1} \leftarrow 3 - j, i \leftarrow i + l$;

 set $l \leftarrow 0, k \leftarrow 0$.

S7: If $i \leq n / 2$ and $G \neq 1$ then set $j \leftarrow 0$, goto S4.

S8: If $G \neq 1$ then

 set $h \leftarrow -h$, do $Z_h \leftarrow Z_h W^{2(-1)^h} \% M$, goto S2.

It is easy to see that only if \bar{G} is a true ciphertext, can the decryption algorithm terminate normally.

In this wise, the original plaintext block $B_1 \dots B_{n/2}$, namely $b_1 \dots b_n$ is recovered although $r_1 \dots r_{n/2}$ is introduced into the encryption process.

6.3.2 Appending a Stochastic Bit String to a Plaintext

The second approach to avoiding adaptive-chosen-ciphertext attack is to append a stochastic fixed-length bit string to the terminal of a plaintext block when it is encrypted. For a concrete implementation, refer to the OAEP+ scheme [19].

Of course, we may combine the first approach with the second approach.

6.4 Avoiding Meet-in-the-middle Attack

Meet-in-the-middle dichotomy was first developed in 1977 [20]. Section 3.10 of [11] puts forward a meet-in-the-middle attack on a subset sum problem.

INPUT: a set of positive integers $\{C_1, C_2, \dots, C_n\}$ and a positive integer s .

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

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

S2: Construct a table with entries $(\sum_{i=1}^t C_i b_i, (b_1, b_2, \dots, b_t))$ for $(b_1, b_2, \dots, b_t) \in (\mathbb{Z}_2)^t$.

Sort this table by the first component.

S3: For each $(b_{t+1}, b_{t+2}, \dots, b_n) \in (\mathbb{Z}_2)^{n-t}$, 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, b_2, \dots, b_n)).

S4: Return (no solution exists).

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

Likewise, meet-in-the-middle dichotomy may be used to attack the ASSP $\bar{G} \equiv \prod_{i=1}^{n/2} (C_{3(i-1)+B_i})^{B_i} (\% M)$ when $B_{n/4} \neq 00$ and $B_{n/2} \neq 00$ which occur with the probability $9/16 = 0.5625$, and it will take the time complexity of $O(n2^{5n/8} \lg^2 M)$ bit operations.

Hence, to avoid meet-in-the-middle attack as efficiently as possible, parallel to Section 6.3.1, a random bit string $r_1 \dots r_{n/2}$ should be brought into the encryption algorithm so as to extend the scope of exhaustive search.

In this case, the ASPP is converted into $\bar{G} \equiv \prod_{i=1}^{n/2} (C_{r_i(3(i-1)+B_i) + \neg r_i(3(i-B_i) \% n/2 + B_i)})^{B_i} (\% M)$, and moreover, it is not difficult to understand that the time complexity of the attack task is increased to $O(n2^{5n/8} \lg^2 M)$ bit operations. Notice that the probability of success is still $9/16$.

Concretely speaking,

when $n = 80$ with $\lceil \lg M \rceil = 384$, $F = n2^{5n/8} \lg^2 M = 2^7 2^{50} 2^{18} = 2^{75}$ bit operations,

when $n = 96$ with $\lceil \lg M \rceil = 464$, $F = n2^{5n/8} \lg^2 M = 2^7 2^{60} 2^{18} = 2^{85}$ bit operations,

when $n = 112$ with $\lceil \lg M \rceil = 544$, $F = n2^{5n/8} \lg^2 M = 2^7 2^{70} 2^{20} = 2^{97}$ bit operations,

when $n = 128$ with $\lceil \lg M \rceil = 640$, $F = n2^{5n/8} \lg^2 M = 2^7 2^{80} 2^{20} = 2^{107}$ bit operations.

If the above time complexities do not satisfy users' requirement, the users need further appending a stochastic bit string to a plaintext.

7 Conclusion

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

The cryptoscheme builds its own security on the two problems $C_i = (A_i W^{\ell(i)})^\delta \% M$ with $A_i \in \mathcal{A}$ and $\ell(i) \in (x, y, z) \in \Omega$ and $\bar{G} \equiv \prod_{i=1}^{n/2} (C_{3(i-1)+B_i})^{B_i} (\% M)$ that are classified as MPP and ASPP to which no DLP subexponential time solutions are found [21], and there exist only exponential time solutions so far, utilizes a bit-pair string to decrease the bit-length of the modulus M , exploits a bit-pair shadow string to prevent attack by LLL lattice base reduction, and employs the two approaches of introducing a random bit string into the encryption and appending a stochastic bit string to a plaintext to avoid attack by adaptive-chosen-ciphertext measure and meet-in-the-middle dichotomy.

As $n = 80, 96, 112, \text{ or } 128$, there exists $\lceil \lg M \rceil = 384, 464, 544, \text{ or } 640$, which assures that when a JUNA ciphertext is converted into an ASSP through a discrete logarithm, the density of a related ASSP knapsack is pretty high, and larger than 1.

There 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, Changxiang Shen, Zhengyao Wei, Andrew C. Yao, Binxing Fang, Xicheng Lu, and Guangnan Ni for their important guidance, suggestions, and help.

The authors also would like to thank the Professors Dingyi Pei, Dengguo Feng, Jie Wang, Ronald L. Rivest, Moti Yung, Adi Shamir, Dingzhu Du, Mulan Liu, Huanguo Zhang, Maozhi Xu, Yixian Yang, Xuejia Lai, Xiaoyun Wang, Yupu Hu, Kefei Chen, Jiwu Jing, Rongquan Feng, Ping Luo, Jianfeng Ma, Xiao Chen, Dongdai Lin, Zhenfu Cao, Chao Li, Lei Hu, Lusheng Chen,

Wenbao Han, Xinchun Yin, Bogang Lin, Qibin Zhai, Dake He, Hong Zhu, Zhiying Wang, and Quanyuan Wu for their important advice, suggestions, and corrections.

References

- [1] S. Su and S. Lü, A Public Key Cryptosystem Based on Three New Provable Problems, *Theoretical Computer Science*, v426-427, 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*, v21(2), 1978, pp. 120-126.
- [3] V. Niemi, A New Trapdoor in Knapsacks, *Proc. 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. Advance in Cryptology: EUROCRYPT '94*, Springer-Verlag, 1994, pp. 112-130.
- [5] E. F. Brickell, Solving Low Density Knapsacks, *Proc. 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*, v2(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*, v24(5), 1978, pp. 525-530.
- [8] S. Y. Yan, *Number Theory for Computing* (2nd ed.), Springer-Verlag, Berlin, 2002, ch. 1.
- [9] L. Fibiková 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*, New York: Springer-Verlag, 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, *Proceedings 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, no. 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. 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.
- [18] 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.
- [19] V. Shoup, OAEP Reconsidered, *Proc. of Advance in Cryptology: Crypto '01*, Springer-Verlag, 2001, pp. 239-259.
- [20] W. Diffie and M. E. Hellman, Exhaustive Cryptanalysis of the NBS Data Encryption Standard, *Computer*, v10 (6), 1977, pp. 74-84.
- [21] S. Su and S. Lü, *REESSE1+ · Reward · Proof by Experiment on 80-bit Moduli*, <http://arxiv.org/pdf/0908.0482>, Aug. 2009 (revised Dec. 2012).