A Public Key Cryptoscheme Using the Bit-pair Method [*]

Shenghui Su [1, 2], Maozhi Xu [3], and Shuwang Lü [4]

[1] College of Computers, Beijing University of Technology, Beijing
[2] College of Info. Engineering, Yangzhou University, Yangzhou
[3] School of Mathematics, Peking University, Beijing
[4] Graduate School, Chinese Academy of Sciences, Beijing

Abstract: The authors give the definition 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 the correctness of the decryption algorithm, deduce the probability that a plaintext solution is nonunique is nearly zero, and analyze the security of the cryptoscheme against extracting a private key from a public key, and recovering a plaintext from a ciphertext on the assumption that IFP, DLP, and SSP 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, or 640 corresponding to n = 80, 96, 112, 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, and used for data encryption and digital signing.

In REESSE1+, a ciphertext is defined as $\bar{G} \equiv \prod_{i=1}^{n} C_i^{\underline{b}_i}$ (% M), where $\underline{b}_i$ 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 $\bar{G} \equiv g^{v}$ (% 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 $\bar{G} \equiv \prod_{i=1}^{n} C_i^{\underline{b}_i}$ (% M) for $\underline{b}_1 \ldots \underline{b}_n$ is equivalent to solving

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

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 due to every $\underline{b}_i \in [0, n]$, shortly ASSP [1]. Likewise, due to every $\underline{b}_i \in [0, n]$, $\{u_1, \ldots, u_n\}$ 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

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

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 $\lceil \lg M \rceil$ = 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 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 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.

[*] This research is supported by MOST with Project 2007CB311100 and 2009AA01Z441. Email: sheenway@126.com

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-lengthof 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, Ᵽ 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 whose elements are pairwise coprime is a coprime sequence.

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

Refer to [1] for its proof.

2.2 A Bit Shadow

Definition 2: Let $b_1 \ldots b_n \neq 0$ be a bit string. Then $\underline{b}_i$ with $i \in [1, n]$ is called a bit shadow if it comes from such a rule: ① $\underline{b}_i = 0$ if $b_i = 0$, ② $\underline{b}_i = 1$ + the number of successive 0-bits before b_i if $b_i = 1$, or ③ $\underline{b}_i = 1$ + 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 $b_1 \ldots b_{16}$ = 1001000001001100 or 0010010011000100, $\underline{b}_1 \ldots \underline{b}_{16}$ = 3003000006003100 or 0050030031000400.

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

Proof.

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

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

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

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

Proof.

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

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

In addition, for any arbitrary bit shadow $\underline{b}_1 \ldots \underline{b}_n$, there always exists a preimage $b_1 \ldots b_n$. Thus, the mapping from $b_1 \ldots b_n$ to $\underline{b}_1 \ldots \underline{b}_n$ is one-to-one.

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

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 $\prod_{i=1}^{n} A_i^{\underline{b}_i} = \prod_{i=1}^{n} A_i^{\underline{b}'_i}$, which is in direct contradiction to $\underline{b}_1 \ldots \underline{b}_n \neq \underline{b}'_1 \ldots \underline{b}'_n$.

Therefore, the mapping from $\underline{b}_1 \ldots \underline{b}_n$ to $\prod_{i=1}^{n} A_i^{\underline{b}_i}$ is injective [8].

In summary, the mapping from $\underline{b}_1 \ldots \underline{b}_n$ to $\prod_{i=1}^{n} A_i^{\underline{b}_i}$ is one-to-one, and further 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 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 $\{A_1, \ldots, A_n\}$ is substituted with $\{A_1, A_2, A_3, \ldots, A_{3n/2-2}, A_{3n/2-1}, A_{3n/2}\}$ that may be 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 $\{C_1, \ldots, C_n\}$ is substituted with $\{C_1, C_2, C_3, \ldots, C_{3n/2-2}, C_{3n/2-1}, C_{3n/2}\}$, where C_i with $i \in [1, 3n/2]$ is acquired from A_i and other private parameters.

***Definition 3**:* Let $\{A_1, \ldots, A_{3n/2}\}$ 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_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 $B_1, \ldots, B_{n/2}$ is called a bit-pair string, shortly $B_1 \ldots B_{n/2}$.

***Property 3**:* Let $\{A_1, \ldots, A_{3n/2}\}$ 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].

***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 $n = 16$, then when $B_1 \ldots B_8 = 1001000001001100$ or 0010010011000100, $\underline{B}_1 \ldots \underline{B}_8 = 21003020$ 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 $\underline{B}_{y_1}, \ldots, \underline{B}_{y_{n-k}}$ are 00-pair shadows.

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

***Property 4**:* Let $\{A_1, \ldots, A_{3n/2}\}$ 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 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:

- $\ell(.)$ is an injection from the domain $\{1, \ldots, \tilde{n}\}$ to the codomain $\Omega \subset \{5, \ldots, \bar{M}\}$, where $\bar{M}$ is a large positive integer;
- 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.

Obviously, 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 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 $W^{\ell(i)} \equiv A_i^{-1} C_i$ (% M) face an unsolvable Galois group when $A_i \le 1201$ is guessed [10].

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

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

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

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

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

Refer to [1] for its proof.

Notice that according to the proof in [1], it is not difficult to understand that if $\Omega = \{5, \ldots, \tilde{n} + 4\}$ is substituted with $\Omega \subset \{\pm 5, \pm 7, \ldots, \pm(2\tilde{n} + 3)\}(x + y \ne 0\ \forall x, y \in \Omega)$, 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 $B_1 \ldots B_{n/2}$ represent a related plaintext block $b_1 \ldots b_n \ne 0$.

3.1 The Key Generation Algorithm

Let $p_1, \ldots, p_{\tilde{n}}$ be the first $\tilde{n}$ 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 $\Omega \subset \{(\pm 5, \pm 7, \pm 9), \ldots, (\pm(2\tilde{n} - 1), \pm(2\tilde{n} + 1), \pm(2\tilde{n} + 3))\}$ with $|x_1||x_2||x_3| \ne |y_1||y_2||y_3|\ \forall (x_1, x_2, x_3), (y_1, y_2, y_3) \in \Omega$, and $|\Omega| = n / 2$. Then a concrete Ω is one of $(3!)^{n/2} 2^{\tilde{n}}$ sets consisting of 3-tuple elements.

Additionally, let $\Lambda = \{2, \ldots, Ᵽ\}$, where $Ᵽ = 863, 937, 991$, or 1201 as $n = 80, 96, 112$, or 128.

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

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

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

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 $\prod_{i=1}^{k} e_i \ge 2^{10}$ and $p_k \approx \tilde{n}/2 + 1$.

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

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

S4: Randomly pick δ, $W \in [1, \bar{M}]$ making $\|W\| \ge 2^{n-30}$ and $\gcd(\delta, \bar{M}) = 1$.

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

At last, obtain a public key $(\{C_i\}, M)$, and a private key $(\{A_i\}, W, \delta, M)$. $\{\ell(i)\}$ may be discarded.

Notice that

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

② $\gcd(A_{3i-2}, A_{3i-1}, A_{3i}) \ne 1$ 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 $n = 80, 96, 112$, or 128, there exists $\lceil \lg M \rceil = 384, 464, 544$, or 640.

***Definition 6**:* Seeking $\{A_i\}$, $\{\ell(i)\}$, W, δ from the key transform $C_i = (A_i W^{\ell(i)})^{\delta}$ % M with $\ell(i) \in (\ell(j + 1), \ell(j+2), \ell(j+3)) \in \Omega$ and $A_i \in \Lambda$ for $i = 1, \ldots, \tilde{n}$ is referred to as the multivariate permutation problem, where $j = 3\lfloor (i-1) / 3 \rfloor$, and shortly MPP.

3.2 The Encryption Algorithm

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

Notice that if the number of successive 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 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 $k \leftarrow k + 1$, $\underline{B}_i \leftarrow 0$
    else let $\underline{B}_i \leftarrow k + 1$, $k \leftarrow 0$, 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 $\underline{B}_s \leftarrow \underline{B}_s + k$.
S5: Compute $\bar{G} \leftarrow \prod_{i=1}^{n/2} (C_{3(i-1)+B_i})^{\underline{B}_i}$ % $M$.
```

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

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

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

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

3.3 The Decryption Algorithm

Assume that $(\{A_i\}, W, \delta, M)$ is a related private key, and $\bar{G}$ is a ciphertext.

Notice that due to $\sum_{i=1}^{n/2} \underline{B}_i = n/2$ and $\ell(3(i-1)+B_i)$ being odd (except $\ell(0) = 0$), $\underline{k} = \sum_{i=1}^{n/2} \underline{B}_i \ell(3(i-1)+B_i)$ 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 $B_1 \ldots B_{n/2} \leftarrow 0, j \leftarrow 0, k \leftarrow 0, i \leftarrow 1, s \leftarrow 0, G \leftarrow Z_h$.
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 $h \leftarrow \neg h$, do $Z_h \leftarrow Z_h W^{2(-1)^h}$ % $M$, goto S2 else end.
```

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

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

4 Correctness and Uniqueness

In this section, we discuss whether a ciphertext can be decrypted correctly.

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

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

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

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

Proof:

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

Additionally, let $A_0 = 1$.

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

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

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

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

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

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

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

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

However, because $|\underline{k}| < n(2\tilde{n} + 3)/2 = 3n(n+1)/2$ is comparatively small, we may search $\underline{k}$ 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 $B_1 \ldots B_{n/2}$ will be acquired at the same time the condition $G = 1$ is satisfied.

4.2 Uniqueness of a Plaintext Solution

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

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

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

That is,

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

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

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

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

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

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

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

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

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

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

$B_1 \ldots B_{n/2}$ contains n bits which indicates $\prod_{i=1}^{n/2} (A_{3(i-1)+B_i})^{\underline{B}_i}$ 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 $A_1{}^{-1}, \ldots, A_{\tilde{n}}{}^{-1}$ are not necessarily coprime, some values of θ may possibly occur repeatedly.

Because $|\underline{k} - \underline{k}'| < 3n(n + 1) \leq 47601 \approx 2^{16}$ 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, or 128, there is $\lceil \lg M \rceil$ = 384, 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 $\sum_{i=1}^{n/2} \underline{B}_i = n/2$, may exclude some unoriginal plaintext solutions.

5 Security of the JUNA Cryptoscheme

We will analyze the security of the new cryptoscheme against extracting a private key from a public key and recovering a plaintext from a ciphertext.

5.1 Security of a JUNA Private Key

The security of a JUNA private key depends on the MPP $C_i = (A_i W^{\ell(i)})^{\delta}$ % M with $A_i \in \Lambda$ and $\ell(i) \in (x, y, z) \in \Omega$ for $i = 1, \ldots, \tilde{n}$.

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

Its proof is parallel to that of Property 4 in [1].

Other analysis is parallel to Section 4 in [1].

5.2 Security of a JUNA Plaintext

The security of a JUNA plaintext depends on the ASPP $\bar{G} \equiv \prod_{i=1}^{n/2} (C_{3(i-1)+\underline{B}_i})^{\underline{B}_i}$ (% M) with $C_0 = 1$.

5.2.1 Two Properties

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

Its proof is parallel to that of Property 5 in [1].

Property 8: The ASPP $\bar{G} \equiv \prod_{i=1}^{n/2} (C_{3(i-1)+\underline{B}_i})^{\underline{B}_i}$ (% M) with $C_0 = 1$ is computationally at least equivalent to DLP in the same prime field, where $\underline{B}_1 \ldots \underline{B}_{n/2}$ is the bit-pair shadow string of $B_1 \ldots B_{n/2} \neq 0$.

Its proof is parallel to that of Property 6 in [1].

5.2.2 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 solution vector appear in the reduced base with large probability.

In the JUNA cryptoscheme, there are n = 80, 96, 112, or 128 and $\lceil \lg M \rceil$ = 384, 464, 544, 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 $\underline{B}_1 \ldots \underline{B}_{n/2}$ to $\underline{b}'_1 \ldots \underline{b}'_{\tilde{n}}$ by the following rule for $i = 1, \ldots, n/2$:

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

For example, suppose that $B_1 \ldots B_4$ = 00 10 01 00, then $\underline{B}_1 \ldots \underline{B}_4$ = 0310, and $\underline{b}'_1 \ldots \underline{b}'_{12}$ = 000 030 100 000.

In this way, there is

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

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

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

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

$$u_1 \underline{b}'_1 + \ldots + u_{\tilde{n}} \underline{b}'_{\tilde{n}} \equiv v \text{ (\% } \bar{M}), \tag{3}$$

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

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

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

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

Namely,

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

which is slightly different from Formula (2).

Concretely speaking, in the JUNA cryptoscheme, there are

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

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

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

$D = 196 \times 6 / 640 \approx 1.8375 > 1$ for $n = 128$ 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 the original solution vector will not occur in the final reduced lattice base in general.

5.2.3 Avoiding Adaptive-chosen-ciphertext Attack

To avoid adaptive-chosen-ciphertext attack [13], a random bit string $r_1 \ldots r_{n/2}$ is introduced into the encryption algorithm, which makes the algorithm be able to produce many different ciphertexts to the identical plaintext. Correspondingly, the decryption algorithm needs some adjusting.

The concrete method is parallel to Section 5.3 in [1].

5.2.4 Avoiding Meet-in-the-middle Attack

Meet-in-the-middle dichotomy was first developed in 1977 [14]. Section 3.10 of [11] brings forth a meet-in-the-middle attack on the subset sum problem.

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 $B_{n/2} \neq 00$ with the probability of success $9 / 16 = 0.5625$ and the time complexity of attack task $O(n2^{n/2})$.

Hence, to avoid probable meet-in-the-middle attack, parallel to the above section, a random bit string $r_1 \ldots r_{n/2}$ should be brought into the encryption algorithm, which makes the algorithm be able to output many different ciphertexts to the identical plaintext so as to extend the scope of exhaustive search. The concrete method is similar to Section 5.3 in [1].

6 Conclusion

The paper proposes a new public key cryptoscheme that is based on the two problems MPP and ASPP to which no subexponential time solutions are found so far [15], includes the key generator, encryption algorithm, and decryption algorithm, utilizes a bit-pair string to decrease the bit-length of the modulus M, and exploits a bit-pair shadow string to prevent attack by LLL lattice base reduction.

As n = 80, 96, 112, or 128, there exists $\lceil \lg M \rceil$ = 384, 464, 544, 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 among time, space, and security which is none other than a delicate thing.

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. 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*, 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] 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.

[14] W. Diffie and M. E. Hellman, Exhaustive Cryptanalysis of the NBS Data Encryption Standard, *Computer*, v10 (6), 1977, pp. 74-84.

[15] 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).