Knapsack Public-Key Cryptosystem Using Chinese Remainder Theorem

Yasuyuki MURAKAMI[*] Takeshi NASAKO [†]
yasuyuki@isc.osakac.ac.jp nasako@m.ieice.org

Abstract. The realization of the quantum computer will enable to break public-key cryptosystems based on factoring problem and discrete logarithm problem. It is considered that even the quantum computer can not solve *NP*-hard problem in a polynomial time. The subset sum problem is known to be *NP*-hard. Merkle and Hellman proposed a knapsack cryptosystem using the subset sum problem. However, it was broken by Shamir or Adleman because there exist the linearity of the modular transformation and the specialty in the secret keys. It is also broken with the low-density attack because the density is not sufficiently high. In this paper, we propose a new class of knapsack scheme without modular transformation. The specialty and the linearity can be avoidable by using the Chinese remainder theorem as the trapdoor. The proposed scheme has a high density and a large dimension to be sufficiently secure against a practical low-density attack.

Keywords: knapsack public-key cryptosystem, subset sum problem, low-density attack, Chinese remainder theorem.

1 Introduction

The realization of the quantum computer will enable to break public-key cryptosystems based on factoring problem and discrete logarithm problem [1]. Under this future threat, it is important to search for secure PKCs based on the other problems. The subset sum problem is known to be *NP*-hard. It is considered that even the quantum computer can not solve *NP*-hard problem in a polynomial time. This fact has motivated us to invent a public-key cryptosystem based on the subset sum problem.

The subset sum problem is to find the solution $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ such that

$$C = a_1x_1 + a_2x_2 + \cdots + a_nx_n$$

for given positive integers $a_1, a_2, \ldots, a_n$ and the subset sum C.

The public-key cryptosystems using the subset sum problem have been conventionally called knapsack cryptosystems. Merkle and Hellman proposed the first knapsack public-key cryptosystem(MH PKC) [2]. MH PKC has a remarkable feature that the encryption and the decryption can be performed very fast. However, it is known that the secret key of MH PKC can be disclosed

[*]Department of Telecommunications and Computer Networks, Osaka Electro-Communication University, 18-8, Hatsu-cho, Neyagawa-shi, Osaka, 572-8530 Japan.

[†]Department of Electronics and Applied Physics, Osaka Electro-Communication University, 18-8, Hatsu-cho, Neyagawa-shi, Osaka, 572-8530 Japan.

by Shamir's attack [3] or Adleman's attack [4] because of the specialty of the super-increasing sequence and the linearity of the modular transformation. The plaintext can be also disclosed with the low-destiny attack [5, 6] because the density is not sufficiently high. These attacks have given the impression that knapsack PKCs are insecure. It is, however, difficult to condemn that all knapsack PKCs can not be secure.

The density, an important parameter in knapsack schemes, is defined by

$$d = \frac{n}{\log_2\{\max(a_1, a_2, \ldots, a_n)\}}.$$

Lagarias and Odlyzko introduced the low-density attack (LDA) for solving the subset sum problem [5]. In LDA, the subset sum problem is converted into the problem of finding the shortest vector in a lattice (SVP). Coster el al. improved LDA so that it can solve almost all subset sum problems of the density less than 0.9408 [6]. However, a practical lattice reduction algorithm such as LLL [7] does not work as SVP oracle when the dimension n becomes larger.

In this paper, we shall propose a new class of knapsack scheme in which the modular transformation is not required. The proposed scheme uses the Chinese remainder theorem (CRT) as the trapdoor. By using CRT, the specialty among the secret keys and the linearity between the secret keys and the public keys can be avoidable. Thus, CRT provides the security against the attack of computing the secret key from the public key. Moreover, we show that a high density and a large dimension can be realized in the proposed scheme. Furthermore, we confirm that the proposed scheme is actually secure against the LDA with the computer experiment.

2 Message Pre-coding

In knapsack schemes, the message is often encoded before encryption. In this section, we shall describe the typical methods of message pre-coding. Any of the message encoding described below can be used as the pre-coding method in the proposed knapsack PKC. Throughout this paper, let the message be represented by $\boldsymbol{m} = (m_1, m_2, \ldots, m_n)$ and be already pre-coded with an appropriate message pre-coding.

2.1 Data Compression Coding

Data compression coding is preferable to remove the message redundancy. Any data compression coding can be used for the pre-coding. After an appropriate data compression coding, the original plaintext message can be divided into the pieces of n-bit messages.

2.2 Schalkwijk Algorithm

Define the set $\mathbb{Z}_\mu$ as $\mathbb{Z}_\mu = \{0, 1, \ldots, \mu - 1\}$ for an integer μ. Let B_{nw} be the set of n-dimensional binary vectors of weight w. Schalkwijk algorithm [8] gives a mapping from an integer $M \in \mathbb{Z}_{\binom{n}{w}}$ into a binary vector $\boldsymbol{m} = (m_1, m_2, \ldots, m_n) \in B_{nw}$. By using Schalkwijk algorithm as the message encoding, weak messages such as low-weight vectors can be avoidable.

The original plaintext message be $M \in \mathbb{Z}_{2^\nu}$. By deciding the system parameters ν, n, w such that $2^\nu < \binom{n}{w}$, it follows that $\mathbb{Z}_{2^\nu} \subset \mathbb{Z}_{\binom{n}{w}}$ and $B_{nw} \subset \{0,1\}^n$. Thus, Schalkwijk algorithms can be used as the mappings between $\mathbb{Z}_{2^\nu}$ and $\{0,1\}^n$.

We shall explain Schalkwijk algorithm as follows. In the following algorithms, we define $\binom{n}{w} = 0$ when $w > n$ for simplicity.

[Message Encoding with Schalkwijk Algorithm]

```
Input: $M \in \mathbb{Z}_{\binom{n}{w}}$
Output: $\boldsymbol{m} \in B_{nw}$
Parameters: $n$: dimension, $w$: weight
Enc($M$)
{
  $l = w$
  For $i = 1$ to $n$ do:
    If $M \geq \binom{n-i}{l}$ {
      $m_i = 1$
      $M \leftarrow M - \binom{n-i}{l}$
      $l \leftarrow l - 1$
    } else {
      $m_i = 0$
    }
  return($\boldsymbol{m}$)
}
```

[Message Decoding with Schalkwijk Algorithm]

```
Input: $\boldsymbol{m} \in B_{nw}$
Output: $M \in \mathbb{Z}_{\binom{n}{w}}$
Dec($\boldsymbol{m}$)
{
  $M = 0$
  $l = w - 1$
  For $i = 1$ to $n$ do:
    If $(m_i = 1)$ {
      $M \leftarrow M + \binom{n-i}{l}$
      $l \leftarrow l - 1$
    }
  return($M$)
}
```

3 Proposed Scheme

In this section, we shall propose a new class of knapsack PKC. In the proposed scheme, public keys are created with CRT instead of the modular transformations.

The keys of the proposed knapsack scheme are the followings:

Public key $\mathcal{PK}$: $\mathcal{PK} = \{\boldsymbol{a}\}$.

Secret key $\mathcal{SK}$: $\mathcal{SK} = \{\boldsymbol{s}_P, \boldsymbol{s}_Q, P, Q, N, \sigma\}$.

3.1 Definitions

Let us define the following functions which are used in the proposed algorithms.

Definition 1 ($f(\cdot)$) *A positive integer s can be uniquely represented as*

$$s = 2^r z,$$

where z be an odd number. Define the function $f(\cdot)$ which calculates r from s, i.e., $r = f(s)$.

Definition 2 ($\alpha(\cdot)$)

$$\alpha(i) = \#\{x \mid x < i, x \in \mathcal{L}\}.$$

3.2 Key Generation

Bob creates a public key $\mathcal{PK}$ and a corresponding secret key $\mathcal{SK}$ by doing the following:

[Algorithm $\mathcal{K}$]

1. Decide the dimension n and let $u = \lfloor n/2 \rfloor$.
2. Define the set $\mathcal{H}$ and $\mathcal{L}$ such that $\mathcal{H} \cup \mathcal{L} = \{1, 2, \ldots, n\}$ and $\mathcal{H} \cap \mathcal{L} = \phi$.
3. For $i = n$ downto $u + 1$ do:
   Generate a random integer $s_i^{(P)}$ such that
   $$f(s_i^{(P)}) = \alpha(u+1).$$
   Generate a random integer $s_i^{(Q)}$ such that
   $$\begin{cases} f(s_i^{(Q)}) \geq \alpha(i) - \alpha(u+1) \quad \wedge \quad s_i^{(Q)} > \sum_{k=i+1}^{n} s_k^{(Q)} & (i \in \mathcal{H}), \\ f(s_i^{(Q)}) = \alpha(i) - \alpha(u+1) & (i \in \mathcal{L}). \end{cases}$$
4. For $i = u$ downto 1 do:
   Generate a random integer $s_i^{(P)}$ such that
   $$\begin{cases} f(s_i^{(P)}) \geq \alpha(i) \quad \wedge \quad s_i^{(P)} > \sum_{k=i+1}^{n} s_k^{(P)} & (i \in \mathcal{H}), \\ f(s_i^{(P)}) = \alpha(i) & (i \in \mathcal{L}). \end{cases}$$
   Generate a random integer $s_i^{(Q)}$.
5. Choose integers P and Q such that
   $$P > \sum_{k=1}^{n} s_k^{(P)},$$
   $$Q > \sum_{k=1}^{n} s_k^{(Q)},$$
   $$\gcd(P, Q) = 1,$$
   and let $N = PQ$.

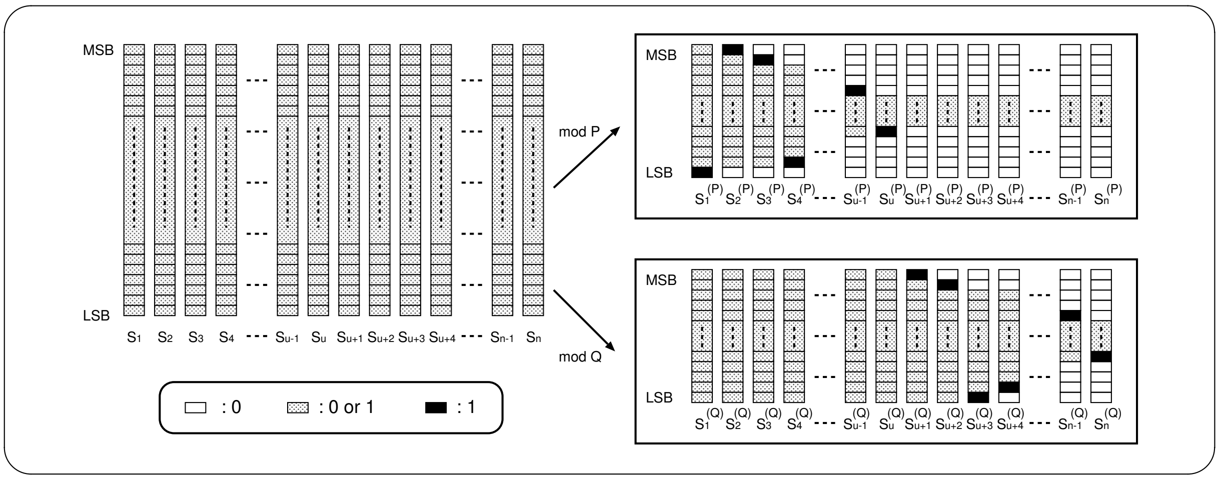

Figure 1: Concept of trapdoor using CRT

6. Compute $\boldsymbol{s} = (s_1, s_2, \ldots, s_n) \in \mathbb{Z}_N^n$ such that

$$s_i \equiv \begin{cases} s_i^{(P)} & (\text{mod } P), \\ s_i^{(Q)} & (\text{mod } Q) \end{cases}$$

   for $i = 1$ to n with the Chinese remainder theorem.

7. Select a random permutation $\sigma \in S_n$, where S_n denotes the set of permutations of integers $\{1, 2, \ldots, n\}$.

8. For $i = 1$ to n do:

$$a_i = s_{\sigma(i)}. \tag{1}$$

9. Publicize the public key $\boldsymbol{a}$.

3.3 Encryption

Alice encrypts a message $\boldsymbol{m} = (m_1, m_2, \ldots, m_n) \in \{0, 1\}^n$ into the ciphertext $C \in \mathbb{Z}$ by doing the following:

[Algorithm $\mathcal{E}$]

1. Obtain Bob's public key $\boldsymbol{a}$.

2. Compute the ciphertext $C \in \mathbb{Z}$ as follows:

$$C = \sum_{i=1}^{n} a_i m_i.$$

3. Send the ciphertext C to Bob.

3.4 Decryption

Bob decrypts the message $\boldsymbol{m} = (m_1, m_2, \ldots, m_n) \in \{0,1\}^n$ from the ciphertext $C \in \mathbb{Z}$ by doing the following:

[Algorithm $\mathcal{D}$]

1. Compute $C_P \in \mathbb{Z}_P$ and $C_Q \in \mathbb{Z}_Q$ as follows:

$$\begin{aligned} C_P &= C \bmod P, \\ C_Q &= C \bmod Q. \end{aligned}$$

2. For $i = 1$ to u do:

$$\begin{aligned} \widehat{m}_i &= \begin{cases} \begin{cases} 1 & (C_P \geq s_i^{(P)}) \\ 0 & (C_P < s_i^{(P)}) \end{cases} & (i \in \mathcal{H}), \\ \begin{cases} 1 & (f(C_P) = f(s_i^{(P)})) \\ 0 & (f(C_P) \neq f(s_i^{(P)})) \end{cases} & (i \in \mathcal{L}). \end{cases} \\ C_P &\leftarrow C_P - \widehat{m}_i s_i^{(P)} \\ C_Q &\leftarrow C_Q - \widehat{m}_i s_i^{(Q)}. \end{aligned}$$

3. For $i = u+1$ to n do:

$$\begin{aligned} \widehat{m}_i &= \begin{cases} \begin{cases} 1 & (C_Q \geq s_i^{(Q)}) \\ 0 & (C_Q < s_i^{(Q)}) \end{cases} & (i \in \mathcal{H}), \\ \begin{cases} 1 & (f(C_Q) = f(s_i^{(Q)})) \\ 0 & (f(C_Q) \neq f(s_i^{(Q)})) \end{cases} & (i \in \mathcal{L}). \end{cases} \\ C_P &\leftarrow C_P - \widehat{m}_i s_i^{(P)} \\ C_Q &\leftarrow C_Q - \widehat{m}_i s_i^{(Q)}. \end{aligned}$$

4. If $C_P \neq 0 \quad \vee \quad C_Q \neq 0$ then output "Invalid Ciphertext" and abort this Algorithm.

5. For $i = 1$ to n do:

$$m_i = \widehat{m}_{\sigma(i)}. \tag{2}$$

3.5 Correctness

The ciphertext C can be represented by

$$C = \sum_{i=1}^{n} s_i \widehat{m}_i.$$

That is,

$$C = \sum_{i=1}^{n} s_{\sigma(i)} \widehat{m}_{\sigma(i)}.$$

From Eqs.(4) and (2), we have

$$C = \sum_{i=1}^{n} a_i m_i.$$

Thus, the decryption works.

3.6 Density

The density d of the proposed knapsack scheme is estimated by

$$\begin{aligned} d &\simeq \frac{n}{\log_2 N} \\ &= \frac{n}{\log_2 P + \log_2 Q}. \end{aligned}$$

From $\log_2 P > u + \log_2 n$ and $\log_2 Q > n - u + \log_2 n$, we can let $\log_2 P = t_P + u + \log_2 n$ and $\log_2 Q = t_Q + n - u + \log_2 n$ by using appropriate parameters t_P and t_Q. Therefore, we have

$$d \simeq \frac{n}{n + t_P + t_Q + 2\log_2 n}.$$

We would like to recommend that t_P and t_Q takes on the value of 20 to 50 and that $u \simeq n-u \simeq 500$, $n > 1000$. In this case, it is seen that $d > 0.9$ can be realized.

4 Security Discussions

4.1 Computing Secret Keys from Public Keys

4.1.1 Non-linearity

Some attacks of computing secret keys from public keys are proposed on modular knapsack schemes [3, 4]. These attacks can compute secret keys by using a linearity between the secret keys and the public keys. However, the proposed scheme does not use the modular transformation. Thus, the weakness caused from the linearity does not exist. Therefore, we can conclude that this type of attacks can not work for the proposed scheme.

4.1.2 Non-specialty

It is considered that Shamir's attack and Adleman's attack specify the secret keys by using the specialty in the super-increasing sequence. However, in the proposed scheme, it seems that no specialty exists in the sequence $\boldsymbol{s}$.

Let $\widehat{\boldsymbol{s}}$ be the randomly generated sequence whose component sizes are the same as those of $\boldsymbol{s}$. If the random sequence $\widehat{\boldsymbol{s}}$ is used instead of $\boldsymbol{s}$, the subset sum problem is *NP*-hard because there is no trapdoor in $\widehat{\boldsymbol{s}}$. In the proposed scheme, the secret sequence $\boldsymbol{s}$ would not be distinguished from $\widehat{\boldsymbol{s}}$ unless the factors P and Q are known. Thus, the subset sum problem using $\boldsymbol{s}$ would be very hard.

4.1.3 High Confidentiality of P and Q

In the proposed scheme, it is seen that the secret key $\boldsymbol{s}^{(P)}$ can be computed when the factor P is disclosed. And conversely, the factor P can be found when $\boldsymbol{s}^{(P)}$ is disclosed with the following algorithm:

[Finding the Factor P]

```
For $i = 1$ to $n$ do:
  For $j = i$ to $n$ do:
    $P_{ij} = \gcd(a_1 - s_i^{(P)}, a_2 - s_j^{(P)})$.
```

The factor P can be disclosed from the multiple of the factor P which must be included the set $\{P_{ij}\}$.

A similar discussion can be given on Q. We think that P and Q are difficult to be disclosed because it is difficult to factor N even if N is opened. This is the reason that we think the proposed scheme is secure against the attacks for finding secret keys.

4.2 Computing Plaintext from Cipehrtext

It is known that LDA is effective to compute the plaintext from the ciphertext in knapsack PKCs. In LDA, the subset sum problem is converted into the problem of finding the shortest vector in a lattice (SVP).

Coster et al. proposed LDA which can solve almost all subset sum problems of the density less than 0.9408 [6] by using the following matrix with SVP oracle:

$$B = \begin{pmatrix} 1 & 0 & \dots & 0 & -\lambda a_1 \\ 0 & 1 & \dots & 0 & -\lambda a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -\lambda a_n \\ -1/2 & -1/2 & \dots & -1/2 & \lambda C \end{pmatrix},$$

where $\lambda > \sqrt{n}$.

LLL algorithm is known to be a practical method to compute the shortest vector in a lattice. However, it does not work as SVP oracle as the dimension becomes larger. We think that LLL algorithm does not work well when $n > 500$. Thus, there is no practical algorithm to solve the subset sum problem of large dimension. In the proposed scheme we recommend $n > 1000$. Therefore, we can say that the proposed scheme is practically secure against the LDA.

Figure 2 shows the density and the breaking rate for 10000 ciphertexts with LDA using LLL algorithm in NTL [9] for the dimension $n = 24$ to 200 step 2 and the parameters $t_P = t_Q = 43$. It is seen that no plaintext can be disclosed when $n > 100$. Thus, we can confirm that LDA with LLL never break the proposed scheme at the recommended parameters.

5 Conclusion

In this paper, we have proposed a new class of knapsack scheme using only the Chinese remainder theorem without the modular transformation as the trapdoor.

The proposed scheme can be considered secure against the attack of computing the secret key from the public key because the specialty among the secret keys the linearity between the secret keys and the public keys can be avoidable. Moreover, we have shown that a high density and a large dimension can be realized in the proposed scheme. We can conclude that the proposed scheme is secure against the low-density attack because a practical lattice reduction algorithm such as LLL can not solve subset sum problems when the message dimension is sufficiently large. Thus, the proposed scheme is actually secure against the low-density attack. Furthermore, we have confirmed this fact with the computer experiment.

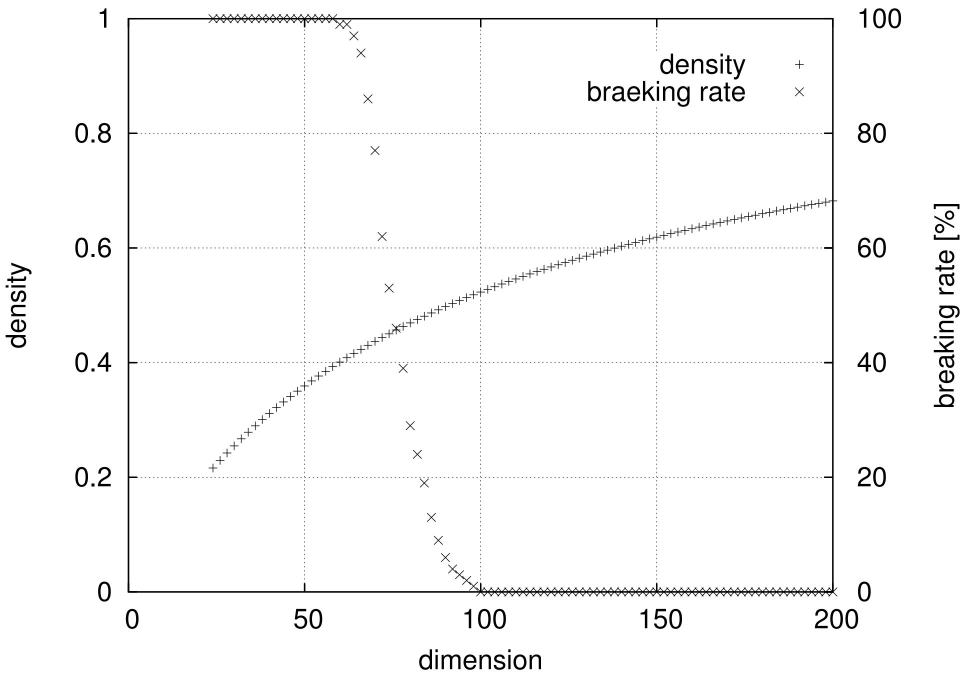

Figure 2: Density and breaking rate

References

[1] P.W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” SIAM J. Comput., vol.26, no.5, pp.1484–1509, 1997.

[2] R.C. Merkle and M.E. Hellman, “Hiding information and signatures in trapdoor knapsacks,” IEEE Transactions on Information Theory, vol.24, no.5, pp.525–530, 1978.

[3] A. Shamir, “A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem,” CRYPTO, pp.279–288, 1982.

[4] L.M. Adleman, “On breaking the iterated Merkle-Hellman public-key cryptosystem,” CRYPTO, pp.303–308, 1982.

[5] J.C. Lagarias and A.M. Odlyzko, “Solving low-density subset sum problems,” J. ACM, vol.32, no.1, pp.229–246, 1985.

[6] M.J. Coster, B.A. LaMacchia, A.M. Odlyzko, and C.P. Schnorr, “An improved low-density subset sum algorithm,” Lecture Notes in Computer Science, vol.547, pp.54–67, 1991.

[7] A. Lenstra, H.L. Jr., and L. Lovasz, “Factoring polynomials with rational coefficients,” Mathematische Annalen, vol.261, no.4, pp.515–534, 1982.

[8] J.P.M. Schalkwijk, “An algorithm for source coding,” IEEE Transactions on Information Theory, vol.IT-18, no.3, pp.395–399, May 1972.

[9] V. Shoup, NTL: A Library for doing Number Theory. Courant Institute, New York University, New York, 2005. Available at http://shoup.net/ntl/.

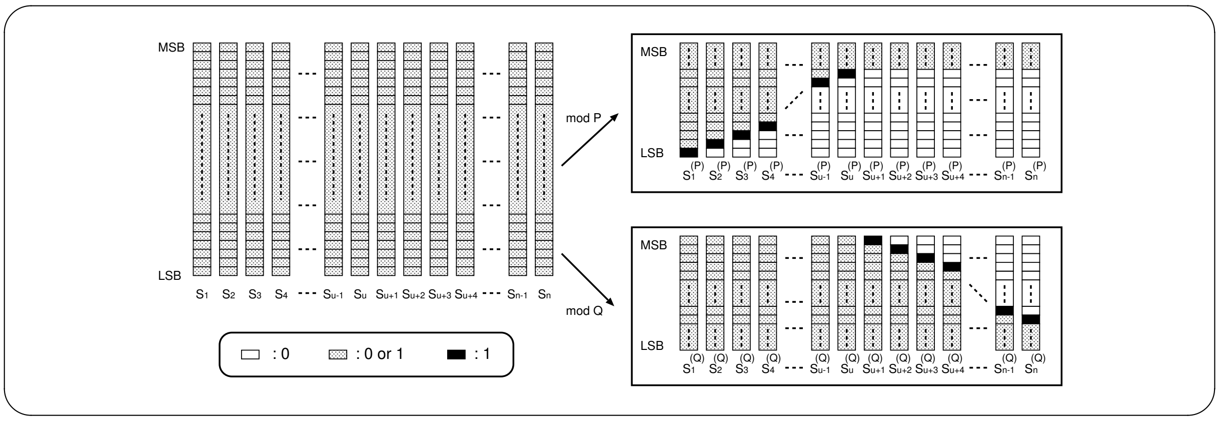

Figure 3: Trapdoor in simple scheme

A Simple Scheme

In this section, we shall present a simple version of the proposed scheme for easy understanding. This scheme corresponds to the case that Bob decides the set $\mathcal{L} = \{1, 2, \ldots, u\}$ and $\mathcal{H} = \{u+1, u+2, \ldots, n\}$ in the proposed scheme.

A.1 Key Generation

[Algorithm $\mathcal{K}$]

1. Decide the parameter t which is the bit-size of the minimum random number.
2. For $i = n$ downto $u+1$ do:
   Generate a t-bit integer $z_i^{(P)}$ at random and computes $s_i^{(P)}$ as follows:
   $$s_i^{(P)} = 2^u z_i^{(P)}$$
   Generate a $(t+n-i+1)$-bit random integer $s_i^{(Q)}$ such that
   $$s_i^{(Q)} > \sum_{k=i+1}^{n} s_k^{(Q)} \tag{3}$$
3. For $i = u$ downto 1 do:
   Generate a $(t+u-i+1)$-bit odd integer $z_i^{(P)}$ at random and computes $s_i^{(P)}$ as follows:
   $$s_i^{(P)} = 2^{i-1} z_i^{(P)}$$
   Generate a $(t+n-u)$-bit integer $s_i^{(Q)}$ at random.
4. Choose integers P and Q such that
   $$P > \sum_{k=1}^{n} s_k^{(P)},$$
   $$Q > \sum_{k=1}^{n} s_k^{(Q)},$$
   $$\gcd(P, Q) = 1,$$

and let $N = PQ$.

5. Compute $\boldsymbol{s} = (s_1, s_2, \ldots, s_n) \in \mathbb{Z}_N^n$ such that

$$s_i \equiv \begin{cases} s_i^{(P)} & (\text{mod } P), \\ s_i^{(Q)} & (\text{mod } Q) \end{cases}$$

   for $i = 1$ to n with the Chinese remainder theorem.

6. Select a random permutation $\sigma \in S_n$.
7. For $i = 1$ to n do:

$$a_i = s_{\sigma(i)}. \tag{4}$$

8. Publicize the public key $\boldsymbol{a}$.

Note that the values, $P - \sum_{k=1}^{n} s_k^{(P)}$, $Q - \sum_{k=1}^{n} s_k^{(Q)}$ and $s_i^{(Q)} - \sum_{k=i+1}^{n} s_k^{(Q)} (i = u+1, u+2 \ldots n)$ should be small so that Eq.(3) may hold in the simple scheme.

A.2 Encryption

Alice encrypts a message $\boldsymbol{m} = (m_1, m_2, \ldots, m_n) \in \{0,1\}^n$ into the ciphertext $C \in \mathbb{Z}$ by doing the following:

[Algorithm $\mathcal{E}$]

1. Obtain Bob's public key $\boldsymbol{a}$.
2. Compute the ciphertext $C \in \mathbb{Z}$ as follows:

$$C = \sum_{i=1}^{n} a_i m_i.$$

3. Send the ciphertext C to Bob.

A.3 Decryption

Bob decrypts the message $\boldsymbol{m} = (m_1, m_2, \ldots, m_n) \in \{0,1\}^n$ from the ciphertext $C \in \mathbb{Z}$ by doing the following:

[Algorithm $\mathcal{D}$]

1. Compute $C_P \in \mathbb{Z}_P$ and $C_Q \in \mathbb{Z}_Q$ as follows:

$$C_P = C \text{ mod } P,$$
$$C_Q = C \text{ mod } Q.$$

2. For $i = 1$ to u do:

$$\begin{aligned}
\widehat{m}_i &= \begin{cases} 1 & (f(C_P) = i - 1) \\ 0 & (f(C_P) \neq i - 1) \end{cases} \\
C_P &\leftarrow C_P - \widehat{m}_i s_i^{(P)} \\
C_Q &\leftarrow C_Q - \widehat{m}_i s_i^{(Q)}.
\end{aligned}$$

3. For $i = u + 1$ to n do:

$$\begin{aligned}
\widehat{m}_i &= \begin{cases} 1 & (C_Q \geq s_i^{(Q)}) \\ 0 & (C_Q < s_i^{(Q)}) \end{cases} \\
C_P &\leftarrow C_P - \widehat{m}_i s_i^{(P)} \\
C_Q &\leftarrow C_Q - \widehat{m}_i s_i^{(Q)}.
\end{aligned}$$

4. If $C_P \neq 0 \quad \vee \quad C_Q \neq 0$ then abort this Algorithm.

5. For $i = 1$ to n do:

$$m_i = \widehat{m}_{\sigma(i)}.$$