Specification of
HIME(R) CryptoSystem

Hitachi, Ltd.

Abstract

This document specifies the public-key cryptosystem HIME(R). HIME(R) is based on a modular squaring (Rabin's public-key encryption scheme [34]) over $\mathbb{Z}_N$, where $N = p^d q$ (p and q are prime integers, and $d > 1$), and utilize the fast caluculation method for decryption. With HIME(R), security is additionally enhanced by the OAEP converting method [5].

HIME(R) has the following exceptional features:

- It is proven to be semantically secure against an adaptive chosen-ciphertext attack (IND-CCA2) in the random oracle model under the factoring assumption of N.
- It has a very fast encryption speed.
- The decryption speed (1536 bits) is about two-and-a-half times faster than that of RSA-OAEP (1024 bits) [5].
- The plaintext space is sufficiently large.
- The amount of computation for the encryption and decryption increases only slightly compared with previous schemes, even if the size of N increases in the future.

HIME(R) is the very practical public-key encryption scheme that is provably secure under the factoring assumption.

This document details the algorithm of HIME(R) and its implementation.

Contents

1 Background

Many public-key cryptosystems have been presented, and among them the RSA scheme is the most famous and is well used. Unfortunately, however, RSA scheme is not secure against an adaptive chosen-ciphertext attack and a concrete attack against an actual system was shown [6]. Thus, RSA must be utilized in secure environment that the active attack is not effective.

Many studies on provably security of public-key cryptosystems have been actively carried out since the early 1990's and many practical provably secure schemes have been presented.

Dolve, Dwork and Naor presented a cryptosystem that is IND-CCA2 using reasonable intractability assumption. However, their scheme is completely impractical inasmuch as it relies on general and expensive construction for a non-interactive zero-knowledge proof [14].

Bellare and Rogaway presented a method for converting public-key encryption schemes based on trapdoor permutation to be IND-CCA1 [5], called OAEP (Although at first it was believed that OAEP could convert such schemes to IND-CCA2 schemes, it has recently been pointed out that the converted schemes are not IND-CCA2 but IND-CCA1 [36]). Their method is very practical and its security can be demonstrated using two assumptions, i.e., the computational intractability of inverting the trapdoor permutation and the existence of ideal hash functions. That is, the proof of security is given in the *random oracle model*, and this is a heuristic proof.

Cramer and Shoup presented a practical public-key cryptosystem which is IND-CCA2 in the standard model [13]. The security of their scheme is based on the intractability of the Decisional Diffie-Hellman (DDH) problem.

Boneh presented the public-key encryption schemes Rabin-SAEP, Rabin-SAEP+ and RSA-SAEP+ which are obtained by applying SAEP or SAEP+ (simplified versions of OAEP or OAEP+[36]) to Rabin's scheme or RSA[8].

Next, we will classify the security of public-key cryptosystems.

Attacks on public-key cryptosystems are classified as follows:

- **Passive Attack**
  - **Chosen-Plaintext Attack (CPA):** An adversary can always gain the ciphertext for her chosen plaintext by sending the plaintext to an encryption oracle. Then the adversary attacks the given target ciphertext (An adversary can always wage this attack on public-key cryptosystems because the encipher keys are published.).
- **Active Attack**
  - **Non-Adaptive Chosen-Ciphertext Attack (CCA1):**An adversary can gain the plaintext for her chosen ciphertext by sending this ciphertext to a decryption oracle before the target ciphertext is given. Then the adversary attacks the given target ciphertext.
  - **Adaptive Chosen-Ciphertext Attack (CCA2):** An adversary can always gain the plaintext for all but her target ciphertext by sending ciphertext to a decryption oracle. Then the adversary attacks the given target ciphertext.

The above description shows that CCA1 is a stronger attack than CPA, and CCA2 is a stronger attack than CCA1.

Security levels of public-key cryptosystems are classified as follows.

- **One-Way (OW) :** It is hard for adversaries to invert the encryption function.
- **Semantically Secure / Indistinguishable (IND) :** It is hard for adversaries to compute partial information about the plaintext from its ciphertext.
- **Non-Malleability (NM) :** It is hard for adversaries to compute a relation for R and the ciphertexts $y_i = E(x_i)$ $(1 \leq i \leq k)$ which satisfy $R(x, x_1, x_2, \ldots, x_k)$ for the ciphertext $y = E(x)$, where E is an encryption function.

Now, we can form {security level}-{attack} pairs. For example, if we say that a public-key cryptosystem is NM-CCA2, it means that the cryptosystem is non-malleable against an adaptive chosen-ciphertext attack. Figure 1 shows the relation among these pairs [1]. Here, $A \rightarrow B$ denotes that if a public-key cryptosystem is A, then it is certainly B. The $A \not\rightarrow B$ denotes its denial. The important point is that IND-CCA2 and NM-CCA2 are equivalent. Therefore, public-key cryptosystems that are IND-CCA2 or NM-CCA2 will have the highest level of security.

$$\begin{tikzcd}
\text{NM-CPA} \arrow[d] \arrow[dr, shift left, "/" marking] & \text{NM-CCA1} \arrow[l] \arrow[d] \arrow[r, shift right, "/" marking] & \text{NM-CCA2} \arrow[l, shift right] \arrow[d, shift right] \\
\text{IND-CPA} \arrow[d] & \text{IND-CCA1} \arrow[l] \arrow[d] \arrow[ul, shift left, "/" marking] & \text{IND-CCA2} \arrow[l] \arrow[d] \arrow[u, shift right] \\
\text{OW-CPA} & \text{OW-CCA1} \arrow[l] & \text{OW-CCA2} \arrow[l]
\end{tikzcd}$$

Figure 1: Relation among definitions of security for public-key cryptosystems.

The main objective of this document is to present the specification of the public-key cryptosystem HIME(R). The design policy of HIME(R) and overview is described in Section 2.1. The algorithm of HIME(R) is given in Section 2.2, and its implementation in Chapter 3.

The security and performance of HIME(R) are described in Self-Evaluation Report.

[1] This relation is discussed in [3].

2 HIME(R)

2.1 Design Policy and Overview

The design policy of HIME(R) is as follows:

(1) Security : It can be proven to be secure in the sense of IND-CCA2 under the assumption of the intractability of primitive problems (whose computational intractability is expected under the enough studies, such as the factoring problem or the discrete logarithm problem).

(2) Efficiency :

(2-1) Both encryption and decryption speeds are fast.

(2-2) The ratio of a plaintext and a ciphertext "(Plaintext)/(Ciphertext)" is not small.

(2-3) The plaintext space is sufficiently large.

(2-4) It can be mounted with a small memory size (including public key and secret key sizes).

In terms of security, we believe that the factoring problem or the discrete logarithm problem are almost ideal as a number theoretic assumption of cryptosystems, because with sufficient study their computational intractability can be taken for granted [20, 25, 26]. Furthermore, there are two categories in number theoretic assumptions that are well utilized in the practical cryptosystems, i.e.:

Factoring Base: Factoring problem, RSA problem, Quadratic residue problem, etc,

Discrete Logarithm Base: Discrete logarithm problem, Computational Diffie-Hellman problem, Decisional Diffie-Hellman problem, etc,

and the factoring problem and the discrete logarithm problem are the most intractable problems in each category.

In constructing HIME(R), we focused on the modular square function (Rabin's encryption function), because it is well known that inverting the encryption function on $\mathbb{Z}_N$ is as intractable as the factoring of N, where $N = pq$ (p and q are prime numbers). Another reason is that it has fast encryption speed. However, the following problems were encountered:

(P-1) The modular square function is not one-way trapdoor permutation, i.e., the decryption is not done uniquely.

(P-2) Rabin's scheme is not secure against a chosen-ciphertext attack.

(P-3) The decryption speed is not fast (i.e., it is as same as that of RSA).

In HIME(R), we utilize OAEP [5] to solve the problems (P-1) and (P-2).

Owing to OAEP, we can get the probabilistically uniqueness of the decryption and prove that it is secure in the sense of IND-CCA2 in the random oracle model by using Coppersmith's algorithm. We used this idea, applying OAEP to Rabin's scheme to solve (P-1) and (P-2), in HIME-2 [22]. After that the same idea was used in Rabin-SAEP and Rabin-SAEP+ even though the padding method differs from OAEP. We can also nearly clear the above conditions (E-2) and (E-3) by using OAEP. We believe that the condition (2-3) is very important, because there are many protocols such as SET (Secure Electronic Transaction) in which the additional information, such as identity information of users or information on cryptosystems, are attached with the data encryption key, even though the main purpose of public-key encryption schemes is to distribute the data encryption key of secret-key encryption schemes.

In HIME(R), we make $N = p^d q$ (p, q: prime numbers, $d > 1$) instead of $N = pq$ and utilize our calculation method over $\mathbb{Z}_N$ to solve (P-3). Previously, a modified RSA scheme was proposed that utilizes such N and applies the original calculation method to make the decryption speed of RSA faster. The original calculation method was done over $\mathbb{Z}_{p^d}$ after $\mathbb{Z}_N$ is divided into $\mathbb{Z}_{p^d}$ and $\mathbb{Z}_q$ by using the Chinese Remainder Theorem (CRT), and the calculated values on $\mathbb{Z}_{p^d}$ and $\mathbb{Z}_q$ were combined on $\mathbb{Z}_N$ by using CRT again.

Our calculation method differs from this previous one in that ours require no calculation by using CRT. As a result, our method has the following advantages:

- It has less modular multiplications than the previous one.
- The actual decryption speed and mountaing size will be smaller than previous one because ours does not require Euclidean algorithm for CRT.

Although this difference is very small, it is expected that it will be non-negligible in smart card systems and in systems in which much decryption processing must be done at one time.

On the other hand, HIME(R) avoids the need for a hybrid scheme [2] with a secret-key encryption scheme, meaning that solving (E-4) would require no secret-key encryption scheme to enable public-key encryption. Another problem with hybrid schemes is that they may require the use of two different secret-key cryptosystems in a single system, which would add to development costs.

From the above discussion, HIME(R) has almost ideal features as follows:

(H-1) It is proven to secure in the sense of IND-CCA2 in the random oracle model under the factoring assumption of N.

(H-2) It has a very fast encryption speed.

(H-3) Its decryption speed (1536 bits) is about two-and-a-half times faster than that of RSA-OAEP (1024 bits).

(H-4) The plaintext space is sufficiently large.

[2] EPOC-2 [10] and EPOC-3 [32] are known as the factoring base hybrid scheme.

(H-5) The amount of computation for the encryption and decryption increases only slightly compared with previous schemes, even if the size of N increases in the future.

The condition (H-5) is important for future sonsiderations, although we did not adopt this condition in (E-1) $\sim$ (E-4) in Section 2.1. The processing ability of computers is increasing rapidly, then the key length must also increase to stay ahead. This increase in key length impairs the efficiency of encryption schemes. However, our scheme can be used well into the future, because it can achieve efficient encryption and decryption processing even if the key length increases.

The details of the above feature are described in the Self-Evaluation Report of HIEM(R).

2.2 Algorithm of HIME(R)

After this, $|x|$ denotes a binary length of x.

2.2.1 Key Generation

(K-1) Choose large prime numbers p, q, such that $|p| = |q|$, $p \equiv 3 \pmod 4$, and $q \equiv 3 \pmod 4$.

(K-2) choose an integer d with $d > 1$.

(K-3) Compute $N = p^d q$.

(K-4) Choose positive integers k_0, k_1 and n such that $n = k - k_0 - k_1 - 1$ and $2k_0 < k$, where $|N| = k$.

(K-5) Choose the hash functions G and H such that

$$G : \{0,1\}^{k_0} \to \{0,1\}^{k-k_0-1}, \qquad H : \{0,1\}^{k-k_0-1} \to \{0,1\}^{k_0}.$$

Then we make

Secret key: (p, q),

Public key: (N, k, k_0, k_1, G, H).

Note that $N/2 < 2^{k-1} < N < 2^k$. We give the details of the length of each parameter k_0, k_1 and k in section 2.3.

2.2.2 Encryption

(E-1) For a message $m \in \{0,1\}^n$, choose the random number $r \in \{0,1\}^{k_0}$, and compute

$$x = (m0^{k_1} \oplus G(r))||(r \oplus H(m0^{k_1} \oplus G(r))).$$

(E-2) Compute

$$y = x^2 \bmod N.$$

Then, y is given as a ciphertext of m.

2.2.3 Decryption

(D-1) For the given ciphertext y, check if y is a quadratic residue on $\mathbb{Z}_N$, namely check

$$y^{(p-1)/2} \equiv 1 \pmod p \qquad \text{and} \qquad y^{(q-1)/2} \equiv 1 \pmod q.$$

If y is not a quadratic residue, reject it.

(D-2) Compute $\gamma_0, \gamma_1, \gamma_2, \ldots, \gamma_d$ such that

$$\begin{array}{ll}
\gamma_0 = \pm\sqrt{y} \bmod p, & \Gamma_0 = \gamma_0, \\
\gamma_1 = (\pm\sqrt{y} - x_0)/p \bmod q, & \Gamma_1 = \gamma_0 + \gamma_1 p, \\
\quad \cdots & \quad \cdots \\
\gamma_i = \left(\dfrac{y - {\Gamma_{i-1}}^2 \bmod p^i q}{p^{i-1} q} \right) \times (2\gamma_0)^{-1} \bmod p, & \Gamma_i = \Gamma_{i-1} + \gamma_i p^{i-1} q \qquad (i \geq 2) \\
\quad \cdots & \quad \cdots \\
\gamma_{d-1} = \left(\dfrac{y - {\Gamma_{d-2}}^2 \bmod p^{d-1} q}{p^{d-2} q} \right) \times (2\gamma_0)^{-1} \bmod p, & \Gamma_{d-1} = \Gamma_{d-2} + \gamma_{d-1} p^{d-2} q \\
\gamma_d = \left(\dfrac{y - {\Gamma_{d-1}}^2 \bmod p^d q}{p^{d-1} q} \right) \times (2\gamma_0)^{-1} \bmod p, &
\end{array}$$

(D-3) For $\gamma_0, \gamma_1, \gamma_2, \ldots, \gamma_d$, compute

$$x = \gamma_0 + \gamma_1 p + \sum_{i=2}^{d} \gamma_i p^{i-1} q.$$

Four x are computed because each γ_0 and γ_1 takes two values. Let those x be x_1, x_2, x_3, x_4.

(D-4) For each x_i $(1 \leq i \leq 4)$, compute $s_i \in \{0,1\}^{n+k_1}$ and $t_i \in \{0,1\}^{k_0}$ such that

$$x_i = s_i || t_i,$$

if $x_i \in \{0,1\}^{k-1}$. Otherwise, reject y.

(D-5) For each s_i and t_i $(1 \leq i \leq 4)$, compute

$$r_i = H(s_i) \oplus t_i \qquad w_i = s_i \oplus G(r_i)$$

by using the hash functions G and H.

(D-6) For each w_i $(1 \leq i \leq 4)$, compute $m_i \in \{0,1\}^n$ and $z_i \in \{0,1\}^{k_1}$ such that

$$w_i = m_i || z_i,$$

and output

$$\begin{cases} m_i & \text{if } [z_i = 0^{k_1}], \\ \text{“reject”} & \text{otherwise,} \end{cases}$$

as the plaintext of the ciphertext y.

2.2.4 Soundness of Decryption

In the decryption of HIME(R), described in section 2.2.3, the soundness, namey the valid ciphertext is correctly decrypted, is shown probabilistically.

Theorem 2.1. Suppose that G and H are ideal hash functions. In the above algorithm, the plaintext is correctly decoded from the valid ciphertext except with a negligible probability.

Proof. We show that x_i $(1 \le i \le 4)$ are all square roots of y in $\mathbb{Z}_N$ (cf. section 2.2.3 (D-2)). If it is shown, there are at most four square roots of y in $\{0,1\}^{k-1}$. Therefore it is trivial that the probability that the decryption fails is less than $3/2^{k_1}$.

We show this by induction on d. Note that any element x in $\mathbb{Z}_N$ $(N = p^d q)$ can be written by

$$x = \gamma_0 + \gamma_1 p + \sum_{i=2}^{d} \gamma_i p^{i-1} q \qquad (0 \le \gamma_0, \gamma_2, \ldots, \gamma_{d-1} < p, \quad 0 \le \gamma_1 < q),$$

and such γ_i is uniquely determined.

Let $d = 2$. Then, the element x in $\mathbb{Z}_{p^2q}$ can be written by $x = \gamma_0 + \gamma_1 p + \gamma_2 pq$ for some $\gamma_0, \gamma_1, \gamma_2 \in \mathbb{Z}$ $(0 \le \gamma_0, \gamma_2 < p,\ 0 \le \gamma_1 < q)$. Suppose that

$$x^2 \equiv y \pmod{p^2}.$$

Then, we have

$$x^2 \equiv (\gamma_0 + \gamma_1 p + \gamma_2 pq)^2 \equiv {\gamma_0}^2 + {\gamma_1}^2 p^2 + 2\gamma_0\gamma_1 p + 2\gamma_0\gamma_2 pq \equiv y \pmod{p^2 q}. \tag{1}$$

And it follows that

$${\gamma_0}^2 \equiv y \pmod{p} \qquad \text{and} \qquad (\gamma_0 + \gamma_1 p)^2 \equiv y \pmod{q}.$$

Since p and q are Blum numbers, γ_0 and γ_1 can be computed as follows (after testing if $y \bmod p$ and $y \bmod q$ are quadratic residue on $\mathbb{Z}_p$ and $\mathbb{Z}_p$ respectively):

$$\begin{aligned}\gamma_0 &= \pm\sqrt{y} \bmod p = \pm y^{(p+1)/4} \bmod p,\\ \gamma_1 &= (\pm\sqrt{y} - \gamma_0)p^{-1} \bmod q = (\pm y^{(q+1)/4} - \gamma_0)p^{-1} \bmod q.\end{aligned}$$

Furthermore, γ_2 is induced from the equation (1) as follows:

$$\gamma_2 = \frac{y - (\gamma_0 + \gamma_1 p)^2 \bmod p^2 q}{pq} \times (2\gamma_0)^{-1} \bmod p.$$

Note that pq divides $y - (\gamma_0 + \gamma_1 p)^2 \bmod p^2 q$. We can also easily prove that y is a quadratic residue on $\mathbb{Z}_{p^2q}$ if and only if $y \bmod p$ and $y \bmod q$ are respectively quadratic residue on $\mathbb{Z}_p$ and $\mathbb{Z}_q$.

Hence, it was shown that x_0, x_1, x_2, x_4 are all square roots of y in $\mathbb{Z}_{p^2q}$.

Next, let $d > 2$. And assume that Γ_{d-1} $(= \gamma_0 + \gamma_1 p + \sum_{i=2}^{d-1} \gamma_i p^{i-1} q)$ are all square roots of y in $\mathbb{Z}_{p^{d-1}q}$. Note that Γ_{d-1} takes four values in total because γ_0 and γ_1 take two values respectively.

Suppose that

$$x^2 \equiv y \pmod{p^d q}, \tag{2}$$

for some $x \in \mathbb{Z}_{p^d q}$.

Then, from the assumption, x can be written by

$$x = \Gamma_{d-1} + \gamma_d p^{d-1} q,$$

for some $\gamma_d \in \mathbb{Z}$ $(0 \leq \gamma_d < p)$. And we have

$$x^2 \equiv (\Gamma_{d-1} + \gamma_d p^{d-1} q)^2 \equiv {\Gamma_{d-1}}^2 + 2\Gamma_{d-1}\gamma_d p^{d-1} q \equiv y \pmod{p^d q},$$

from the equation (2). Hence, γ_d can be obtained by

$$\gamma_d = \frac{y - {\Gamma_{d-1}}^2 \bmod p^d q}{p^{d-1} q} \times (2\gamma_0)^{-1} \bmod p.$$

Note that $p^{d-1}q$ divides $y - {\Gamma_{d-1}}^2 \bmod p^d q$.

We can also easily prove that y is a quadratic residue mod $p^d q$ if and only if $y \bmod p$ and $y \bmod q$ are respectively quadratic residue on $\mathbb{Z}_p$ and $\mathbb{Z}_q$, by induction.

From the above discussion, Theorem 2.1 was proved. □

2.2.5 Remark

We can send

$$\alpha = \begin{cases} 0 & \text{if } 0 < x < N/2, \\ 1 & \text{if } N/2 \leq x < N, \end{cases}$$

or the Jacobi symbol $\beta = \left(\frac{x}{N}\right)$ with a ciphertext to support the decryption processing, where β is useful when d is even. Note that the security proof is not broken even if α and β are sent with the ciphertext.

2.3 Key Length

We firstly describe the length of each parameters k_0, k_1 and k in HIME(R). We recommend to take $|k_0|, |k_1| \geq 128$ in the algorithms of HIME(R).

Table 1 gives the comparison of each modulus length, namely $|k|$, of $N = pq$, $N = p^2 q$ and $N = p^3 q$ where p and q are prime numbers. Each modulus length is determined to make the intractability of factoring almost same when NFS and ECM are used (cf. Self-Evaluation Report of HIME(R)).

RSA and RSA-OAEP are based on the compositive number $N = pq$. HIME(R) is based on the compositive number $N = p^2 q$ or $N = p^3 q$. Table 1 clarify the relation of the length of each modulus when the computational intractability of each factoring problem is the almost same. For example, 1024-bits RSA correponds to 1344bits HIME(R) $(N = p^2 q)$, and 2048-bits RSA corresponds to 2304-bits HIME(R) $(N = p^2 q)$.

Table 1: The length of modulus

	Modulus length (bits)		
$N = pq$	1024	2048	4096
$N = p^2q$	1344	2304	4032
$N = p^3q$	1536	3072	4032

2.4 Remark on Implementation (Manger's Attack)

Recently, Manger presented the chosen ciphertext attack against PKCS #1 v2.0 [27]. His attack is based on the "integrity check". The actual system must be implemented to be careful with integrity check as described in [27]. In this docunemt, we omit the details of countermeasure against this attack, because this problem is not peculiar to HIME(R) but is common to many other public-key cryptosystems.

3 Implementation

In this section, we will explain the method of implementation of HIME(R) public key encryption.

We assume that $d = 2$ in $N = p^d q$, and the key (modulus) length is 1344-bit (the length of the prime factor is 488-bit). Moreover, the length of the parameters k_0, k_1 are asuumed to be 128-bit.

Notations

$x \,||\, y$: the concatenation of bit sequences x and y (*e.g.* $(0110)||(101) = (0110101)$)
$x \oplus y$: the exclusive-or (XOR) of bit sequences x and y
$x \,\&\, y$: the logical multiplication (AND) of bit sequences x and y
$|x|$: the bit length of a bit sequence x
x^n : the most significant n-bit of a bit sequences x
x_n : the least significant n-bit of a bit sequences x
0^m : the 0-sequence of bit length m (*e.g.* $0^5 = (00000)$)
$\{0,1\}^*$: the set of all bit sequence of finite length
$\{0,1\}^i$: the set of all bit sequence of length i
$\mathbf{Z}_n$: the set of residues modulo a positive integer n $(= \{0, 1, 2, \ldots, n-1\})$
$a \bmod n$: the residue of an integer a modulo n $(\in \mathbf{Z}_n)$ (We assume that the representatives of residues are in $\{0, 1, 2, \ldots, n-1\}$.)

3.1 Auxiliary Functions

For implementation of HIME(R), some auxiliary functions such as operations on multiple-precision integers, pseudorandom bit generation, prime number generation are needed. In this section, we will describe these auxiliary functions which are necessary.

3.1.1 Multiple-precision Integers

Ordinarily, large integers are represented by arrays which consist small integers such as "int" on computers. That is, first, a large integer a is written in the form:

$$a = a_n b^n + a_{n-1} b^{n-1} + \cdots + a_1 b + a_0$$

$$(b : \text{the base}, 0 \leq a_i < b)$$

and each $a_i (0 \leq i \leq n)$ is stored in an array $A[i]$. These arrays $A[i] (0 \leq i \leq n)$ are treat as an integer.

In this form, "the most significant bit" of a means the most significant bit of a_n.

For practical methods of addition, subtraction, multiplication, division and exponentiation, see [28].

For the modular multiplication, we recommend the Montogomery method for efficiency. Moreover, some efficient methods for modular inverse and modular exponentiation are known([28]).

3.1.2 Pseudorandom bit generation and a Hash function

It is desired that the random numbers used in our scheme are truely "random", but in practice, we will use outputs of a pseudorandom bit generator which is used in general. We redommend to use the methods which are described in ANSI X9.17[1], X9.31[2], FIPS 186-1[16] and so on.

The functions G_i and H_i $(i = 1, 2)$ which are used in our scheme are able to realized by the Hash function SHA-1 [16]. For more detail, see Section 3.2.

3.1.3 Prime number generation

The prime numbers p,q should be choosen so that the factorization of N is infeasible and it is recommended that the following conditions are satified:

- $p - 1$ has a large prime factor r.
- $p + 1$ has a large prime factor s.
- $r - 1$ has a large prime factor t.

(q is similar)

We call these prime numbers "the strong prime numbers" in this document. For methods of generating strong prime numbers which satisfy the above conditions, see [28].

3.1.4 Representation

We use some constants in this document. The constants are represented in hexadecimal form and the left edge is the most significant bit.

3.2 The functions G, H

We define the functions G, H used in the encryption and the decryption as follows:

h : the hash function SHA-1 $\{0,1\}^* \to \{0,1\}^{160}$
At first, we define some constants (hexadecimal form):

$$\begin{aligned}
C_1 &= h(\mathrm{ABCDEFGHIJ})_{128} = \texttt{9F67EFC6AFA95F1AEF9B3351D6B01D7E} \\
C_2 &= h(\mathrm{BCDEFGHIJA})_{128} = \texttt{170888BEB90A04C3E376F38B82BD1CE3} \\
C_3 &= h(\mathrm{CDEFGHIJAB})_{128} = \texttt{6B7251B714CEA740141D297F8F668AE7} \\
C_4 &= h(\mathrm{DEFGHIJABC})_{128} = \texttt{C8194A67C58DF324670E3809AB2A2520} \\
C_5 &= h(\mathrm{EFGHIJABCD})_{128} = \texttt{AE8908B2099F10ED1D4636879758E7DA} \\
C_6 &= h(\mathrm{FGHIJABCDE})_{128} = \texttt{85A21740116888CEF94EF96E832DB5AB} \\
C_7 &= h(\mathrm{GHIJABCDEF})_{128} = \texttt{980B37185C562631188652C45129D6ED} \\
C_8 &= h(\mathrm{HIJABCDEFG})_{128} = \texttt{25E5813CF47EE7224910F4AA54588C92} \\
C_9 &= h(\mathrm{IJABCDEFGH})_{128} = \texttt{6EB6545C336D76DE9F03288032E31BB1} \\
C_{10} &= h(\mathrm{JABCDEFGHI})_{128} = \texttt{4F20B5C790DF24CF1BE34053D26740DB}
\end{aligned}$$

$C =$
$h(\mathrm{ABCDEFGHIJ})^{64} || h(\mathrm{BCDEFGHIJA})^{64} || \ldots || h(\mathrm{HIJABCDEFG})^{64} =$
`D6B01D7E0591B74882BD1CE3F322876C8F668AE72DDE0ED8AB2A25204C830C79`
`9758E7DAC38E99AE832DB5ABC3AC5B885129D6ED7148036954588C923C159271`

$h'(x) = h((x||x) \oplus C)_{128} : \{0,1\}^{256} \to \{0,1\}^{128}$

$G(x) = \{h'(x||C_1)\ \&\ \alpha\}||h'(x||C_2)||\cdots||h'(x||C_9)||h'(x||C_{10})_{64} : \{0,1\}^{128} \rightarrow \{0,1\}^{1216}$,

$(\alpha = 7\overbrace{\mathtt{F}\ldots\mathtt{F}}^{31}$ (=$2^{127}-1$ as an integer))

$H(x) = h'(x_1||C_1) \oplus h'(x_2||C_2) \oplus \cdots \oplus h'(x_9||C_9) \oplus h'(x_{10}||C_{10}) : \{0,1\}^{1216} \rightarrow \{0,1\}^{128}$
where, $x||0^{64} = x_1||x_2||\cdots||x_9||x_{10}$, $|x_i| = 128$.

Remarks.
1. In the above, we define G so that, as integers, the outputs of G are less than 2^{1215}.
2. The above construction for G, H follows [4].

3.3 Key Generation ($d = 2$, $|N| = 1344$)

We assume that some function **PGen** which outputs the "strong prime numbers" for some inputs "seeds" are available (3.1.3).

Input "seeds" for **PGen**
Output public key (N), secret key (p, q, α, β, z)

1. Generate a strong prime number (using **PGen**) p such that $p \equiv 3 \bmod 4$, $|p| = 448$.
2. Generate a strong prime number (using **PGen**) q such that $q \equiv 3 \bmod 4$, $|q| = 448$, and $q \neq p$.
3. Calculate $N = p^2 q$.
4. If $|N| < 1344$ then update the "seeds" and goto step 1.
5. Calculate $\alpha = (p+1)/4$.
6. Calculate $\beta = (q+1)/4$.
7. Calculate $z = p^{-1} \bmod q$.
8. Rerurn (N) and (p, q, α, β, z) and end.

3.4 Convert

Input m ($|m| = 1088$), a random number R ($|R| = 192$)
Output m' ($|m'| = 1344$)

1. Calculate r = most significant 128-bit of SHA-1(R).
2. Calculate $s = (m||0^{128}) \oplus G(r)$.
3. Calculate $t = r \oplus H(s)$.
4. Return $m' = s||t$ and end.

3.5 Convert^{-1}

Input m' ($|m'| = 1344$)
Output m ($|m| = 1088$), w ($|w| = 128$)

```
1. Let $m' = s'||t'$, $|s'| = 1216$, $|t'| = 128$.
2. Calculate $r' = t' \oplus H(s')$.
3. Calculate $M = s' \oplus G(r')$.
4. Let $M = m||w$, $|m| = 1088$, $|w| = 128$.
5. Return $m$, $w$ and end.
```

3.6 Encryption

Input a palintext m ($|m| = 1088$, where the most significant bit of $m = 0$, *i.e.* as an integer $m < 2^{1087}$), the public key (N)
Output the ciphertext C ($|C| = 1344$)

```
1. Choose a random number $R$ such that $|R| = 192$.
2. Calculate $x = \mathrm{Convert}(m, R)$.
3. Calculate $C = x^2 \bmod N$.
4. Return $C$ and end.
```

3.7 Decryption

Input a ciphertext C ($|C| = 1344$), the public key (N) and the secret key (p, q, α, β, z)
Output the plaintext m ($|m| = 1088$) or ”reject”

```
1. Calculate $C_p = C \bmod p$,  $C_q = C \bmod q$,
2. Calculate $a_1 = C_p^{\alpha} \bmod p$.
   If $a_1^2 \bmod p = C_p$ then calculate $a_2 = p - a_1$ else go to 6.
3. Calculate $b_1 = C_q^{\beta} \bmod q$.
   If $b_1^2 \bmod q = C_q$ then calculate $b_2 = q - b_1$ else go to 6.
4. Calculate
    1)  $y = (b_1 - a_1)z \bmod q$, and $X_1 = a_1 + yp$ (as an integer).
    2)  $y = (b_1 - a_2)z \bmod q$, and $X_2 = a_2 + yp$ (as an integer).
    3)  $y = (b_2 - a_1)z \bmod q$, and $X_3 = a_1 + yp$ (as an integer).
    4)  $y = (b_2 - a_2)z \bmod q$, and $X_4 = a_2 + yp$ (as an integer).
5. For $i$ from 1 to 4 do
    1)  Calculate $s = (X_i^2 - C)/pq$ (as an integer).
    2)  Calculate $t = p - (s \bmod p)$.
    3)  Calculate $Y = t/2X_i \bmod p$.
    4)  Calculate $x = (X_i + Ypq) \bmod N$.
    5)  Calculate $(m', w) = \mathrm{Convert}^{-1}(x)$.
    6)  If $w = 0^{128}$ then $m = m'$ and go to 7.
6. Let $m =$”reject”.
7. Return $m$ and end.
```

3.8 Bit Length

In this section, we describe the bit length of variables used in the encryption and the decryption.

Key

Public key	$\|N\| = 1344$
Secret Key	$\|p\| = \|q\| = \|\alpha\| = \|\beta\| = \|z\| = 488$

Encryption

Input	$\|m\| = 1088$
Variables	$\|R\| = 192, \ \|x\| = 1344$
Output	$\|C\| = 1344$

Decryption

Input	$\|C\| = 1344$
Variables	$\|C_p\| = \|C_q\| = 448, \ \|a_i\| = \|b_i\| = 488, \ \|y\| = 448, \ \|X_i\| = \|s\| = 896$
	$\|t\| = \|Y\| = 488, \ \|x\| = 1344, \ \|m'\| = 1088, \ \|w\| = 128$
Output	$\|m\| = 1088$

References

[1] ANSI X9.17, "American National Standard - Finacial institusion key management (wholesale)", ASC X9 Secretariat - American Bankers Association, 1985.

[2] ANSI X9.31 (Part 2), "American National Standard for Finacial Services - Public Key cryptography using RSA for the financial services industry - Part 2: Hash algorithms for RSA", 1995.

[3] M. Bellare, A.Desai, D.Pointcheval and P. Rogaway. : Relations among notions of security for public-key encryption schemes, *Advances in Cryptology – Crypto'98*, LNCS 1462, Springer-Verlag, pp.26–45 (1998)

[4] M. Bellare and P. Rogaway. : Random oracles are practical – a paradigm for designing efficient protocol, *First ACM Conference on Computer and Communications Security*, pp.62–73 (1993)

[5] M. Bellare and P. Rogaway. : Optimal asymmetric encryption – How to encrypt with RSA, *Advances in Cryptology – Eurocrypt'94*, LNCS 950, Springer-Verlag, pp.92–111 (1994)

[6] D. Bleichenbacher. : Chosen Ciphertext Attacks against Protocols Based on the RSA Encryption Standard PKCS#1, *Advances in Cryptology – Crypto'98*, LNCS 1462, Springer-Verlag, pp.1–12 (1998)

[7] M. Blum and S. Goldwasser. : An efficient probabilistic public-key encryption scheme which hides all partial information, *Advances in Cryptology – Crypto'84*, LNCS 196, Springer-Verlag, pp.289-299 (1985)

[8] D. Boneh. : Simplified OAEP for the RSA and Rabin functions, *Advances in Cryptology – Crypto2001*, LNCS 2139, Springer-Verlag, pp.275-291 (2001)

[9] D. Boneh, G.Durfee and N. Howgrave-Graham. : Factoring $N = p^r q$ for large r, *Advances in Cryptology – Crypto'99*, LNCS 1666, Springer-Verlag, pp.326-337 (1999)

[10] Call for Contributions on New Work Item Proposal on Encryption Algorithms, NTT, 2000-3-10.

[11] D. Coppersmith. : Modifications to the number field sieve, *Journal of in Cryptology*, 6, 3, pp.169-180 (1993)

[12] D. Coppersmith. : Finding a small root of a univariate modular equation, *Advances in Cryptology – Eurocrypt'96*, LNCS 1070, Springer-Verlag, pp.155-165 (1996)

[13] R. Cramer and V. Shoup. : A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack, *Advances in Cryptology – Crypto'98*, LNCS 1462, Springer-Verlag, pp.13-25 (1998)

[14] D. Dolve, C. Dwork and M. Naor. : Non-malleable cryptography, *Proceedings of the 23rd Annual Symposium on Theory of Computing*, ACM, pp.542–552 (1991)

[15] T. ElGamal. : A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Trans. Information Theory*, IT-31, 4, pp.469-472(1985)

[16] FIPS 186, "Digital signature standard", Federal Information Processing Standards Publication 186, U.S. Department of Commerce/N.I.S.T., National Technical Information Service, Springfield, Virginia, 1994.

[17] E. Fujisaki, T. Okamoto and D. Pointcheval : RSA-OAEP is secure under the RSA assumption, *Advances in Cryptology – Crypto2001*, LNCS 2139, Springer-Verlag, pp.269-274 (2001)

[18] S. Goldwasser and M. Bellare. : *Lecture Notes on Cryptography*, `http:/www-cse.ucsd.edu/users/mihir/` (1997)

[19] S. Goldwasser and S. Micali: Probabilistic encryption, *Journal of Computer and System Sciences*, 28, 2, pp.270–299 (1984)

[20] D.M. Gordon : Designing and detecting trapdoors for discrete log cryptosystems, *Advances in Cryptology – Crypto'92*, LNCS 740, Springer-Verlag, pp.66-75 (1992)

[21] Specification of HIME-1 CryptoSystem, Hitachi, Ltd. (2000)

[22] Specification of HIME-2 CryptoSystem, Hitachi, Ltd. (2000)

[23] D. E. Knuth. : *The Art of Computer Programming*, Addison-Wesley (1981)

[24] N. Koblitz. : Elliptic curve cryptosystems, *Math. Comp.*, 48, 177, pp.203-209 (1987)

[25] A.K. Lenstra and H.W. Lenstra,Jr. : *The Development of the Number Field Sieve*, Lect. Notes Math. 1554, Springer-Verlag (1993)

[26] H.W. Lenstra,Jr. : Factoring integers with elliptic curves, *Annals of Math.*, 126, pp.649-673 (1987)

[27] J. Manger : A chosen ciphertext attack on RSA optimal asymmetric encryption padding (OAEP) as standardized in PKCS#1 v2.0, *Advances in Cryptology – Crypto2001*, LNCS 2139, Springer-Verlag, pp.230-238 (2001)

[28] A. J. Menezes, P. van Oorschot and S. A. Vanstone, Handbook of Applied Cryptography, CRC Press (1996).

[29] V. S. Miller. : Use of elliptic curves in cryptography, *Advances in Cryptology – Crypto'85*, LNCS 218, Springer-Verlag, pp.417-426 (1985)

[30] National Institute of Standards, FIPS Publication 180, Secure Hash Standards (1993)

[31] M.Naor and M.Yung. : Public-key cryptosystems provably secure against chosen ciphertext attacks, *Proceedings of the 22nd Annual Symposium on Theory of Computing*, ACM, pp.427–437 (1990)

[32] T. Okamoto and D.Pointcheval: EPOC-3: Efficient Probabilistic Public-Key Encryption-V3 (Submission to P1363a), May 2000

[33] J. M. Pollard. : A Monte-Carlo method for factorization, BIT 15, pp.331-334 (1975)

[34] M. O. Rabin. : Digital signatures and public-key encryptions as intractable as factorization, MIT, Technical Report, MIT/LCS/TR-212 (1979)

[35] R. L. Rivest, A. Shamir and L.Adleman. : A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, Vol.21, No.2, pp.120-126 (1978)

[36] V. Shoup. : OAEP reconsidered, *Advances in Cryptology – Crypto2001*, LNCS 2139, Springer-Verlag, pp.239-259 (2001)

[37] T. Takagi. : Fast RSA-type Cryptosystem Modulo $p^k q$, *Advances in Cryptology – Crypto'98*, LNCS 1462, Springer-Verlag, pp.318-326 (1998)

[38] H.C.Williams. : A modification of the RSA public key encryption procedure, *IEEE Trans. on Information Theory*, IT-26, 6, pp.726-729 (1980)

[39] H. Woll. : Reductions among number theoretic problems, *Information and Computation*, 72, 3, pp.167-179 (1987)

[40] Y. Zheng and J. Seberry. : Practical approaches to attaining security against adaptive chosen Ciphertext Attacks, *Advances in Cryptology – Crypto'92*, LNCS 740, Springer-Verlag, pp.292-304 (1992)