暗号技術仕様書
HIME(R) 暗号

(株) 日立製作所

概 要

本ドキュメントは，公開鍵暗号 HIME(R) の仕様に関するものである．
HIME(R) では，

- 暗号化に合成数 $N = p^d q$ (但し，p,q は素数．$d > 1$) を法とする剰余環上のモジュラー平方関数を用いている．
- 復号化に合成数 $N = p^d q$ に応じた高速計算方法を用いている．
- セキュリティ強化のために，OAEP[5] を利用している．

その結果，次の優れた特徴を持つ．

1. 合成数 $N = p^d q$ の素因数分解問題の困難性を仮定として，ランダムオラクルモデルの下で適応的選択暗号文攻撃に対して強秘匿（IND-CCA2）であることが証明できる．
2. 非常に高速な暗号化処理が可能（１回のモジュラー積のみ）．
3. 復号化速度について，HIME(R) (1536 bits) は RSA-OAEP (1024 bits)[5] に対して，約 2.5 倍の高速処理が可能（モジュラー積の個数による比較）．
4. RSA-OAEP と同等以上の十分大きな平文空間を持つ．
5. 合成数 N のサイズを大きく選んでも，従来方式に比べ，暗号化・復号化の効率性を損なわない．

このように，HIME(R) は素因数分解問題の困難性を前提として安全性証明可能な実用的公開鍵暗号方式である．

本ドキュメントでは，HIME(R) のアルゴリズム，実装の詳細について述べる．

目 次

1 背景

現在まで数多くの公開鍵暗号方式が提案されている．とりわけ，RSA 暗号 [35] は最も有名な方式であり様々なシステムにおいて実用化されている．しかし，RSA 暗号は適応的選択暗号文攻撃を行えば解読可能であり，実システムにおける具体的な攻撃方法も知られている [6]．このように，RSA 暗号を利用する場合は，利用環境を考慮した上で使用する必要がある．

これに対して，公開鍵暗号の安全性証明理論が 90 年代前半から盛んになり，最も強力な攻撃方法である適応的選択暗号文攻撃に対して安全性を証明できる実用的な公開鍵暗号方式が色々と提案されるようになった．簡単に歴史を振り返ると，1991 年に，Dolve, Dwork と Naor は，初めて IND-CCA2 スキームの存在を示した [14]．しかし，彼らの方式は非対話型ゼロ知識証明を使っていることで効率性が悪く実用的ではなかった．1994 年には，Bellare と Rogaway が OAEP (Optimal Asymmetric Encryption Padding) を発表 [5]．OAEP は，落し戸付き一方向性置換から IND-CCA1 の意味において安全であることが証明可能な公開鍵暗号方式を構築する一般的な方法であり（当初，IND-CCA2 の意味において安全であると考えられていたが，Shoup によって一般的には IND-CCA1 の意味において安全であることが指摘される [36]），理想的ランダム関数という非現実的な仮定を必要とするが，優れた効率性を確保できる実用的なアイデアであった．1998 年には，Cramer と Shoup により，理想的ランダム関数を仮定することなく IND-CCA2 の意味で安全である実用的な公開鍵暗号方式が提案された．また，最近では OAEP の修正版である OAEP+ [36] や，簡易版である SAEP, SAEP+ （具体的なスキームとしては，Rabin-SAEP, Rabin-SAEP+, RSA-SAEP+）[8] が提案されている．

以下では，公開鍵暗号の安全性の分類について簡単に述べる．

公開鍵暗号への攻撃方法は次のように分類される：

- 受動的攻撃（Passive Attack）
  - 選択平文攻撃（Chosen-Plaintext Attack: CPA）: 暗号文への攻撃者は，任意の平文に対応する暗号文をいつでも見ることができる環境下において行う攻撃（公開鍵暗号の場合は暗号化鍵が公開されているため，常に選択暗号文攻撃を行うことができる.）.
- 能動的攻撃（Active Attack）
  - 選択暗号文攻撃（Non-Adaptive Chosen-Ciphertext Attack: CCA1）: 暗号文への攻撃者は，ターゲットとなる暗号文が与えられる前にのみ復号化オラクルから任意の暗号文に対応する平文を得ることのできる環境下において行う攻撃.
  - 適応的選択暗号文攻撃（Adaptive Chosen-Ciphertext Attack: CCA2）: 暗号文への攻撃者は，ターゲットとなる暗号文が与えられる前後に拘わらず復号化オラクルから（ターゲット以外の）任意の暗号文に対応する平文を得ることのできる環境下において行う攻撃.

上記から判るように，CPA，CCA1，CCA2 の順でより強力な攻撃である．
公開鍵暗号のセキュリティレベルは次のように分類される．

- 一方向（One Way: OW）：暗号化関数の逆関数を求めることは難しい．
- 強秘匿 / 識別不能性 (Semantic Security / Indistinguishability: IND)：暗号文から平文に関するいかなる部分情報を計算することは難しい．
- 頑強 (Non-Malleable: NM)：暗号文 $y = E(x)$ において，$R(x, x_1, x_2, \ldots, x_k)$ を満たす（別なる）暗号文 $y_i = E(x_i)$ $(1 \leq i \leq k)$ と関係 R を計算することは難しい．

このとき，{ セキュリティレベル }-{ 攻撃 } のペアを考えることができる．例えば，ある公開鍵暗号が NM-CCA2 であるとは，適応的選択暗号文に対して頑強であることを意味する．図 1 は，これらのペアの間の関係を示す[1]．

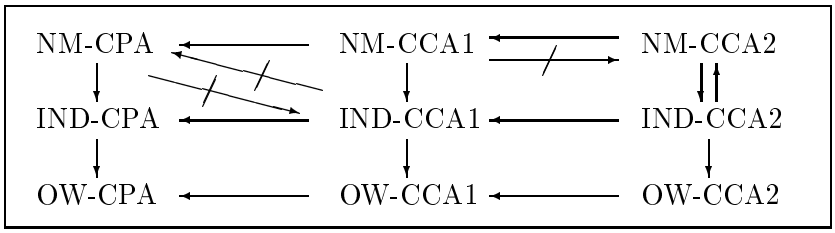

図 1: Relation among definitions of security for public-key cryptosystems.

ここで，$A \to B$ とは,「ある公開鍵暗号が A であれば，その暗号は必ず B である.」ことを意味する．$A \not\to B$ はその否定である．ここで重要なポイントは，IND-CCA2 と NM-CCA2 の等価性が示されている点である．このことから，IND-CCA2 または NM-CCA2 である公開鍵暗号方式が最も安全であると考えられている．

本ドキュメントの目的は，公開鍵暗号 HIME(R) の仕様について述べることにある．第 2 章において，HIME(R) の設計方針と概要（第 2.1 節），アルゴリズム（第 2.2 節）について述べる．また，第 3 章では，HIME(R) の実装について述べる．

HIME(R) の安全性, 性能評価などは自己評価書において行う.

[1] この関係は，文献 [3] で示されている．

2 HIME(R)

2.1 設計方針および概要

HIME(R) の設計方針は次の通りである.

(1) 安全面：プリミティブな問題（素因数分解問題や離散対数問題のように十分な研究の下で計算量的困難性が予想されている問題）の計算量的困難性を仮定として，IND-CCA2 の意味で安全性が証明できること.

(2) 効率面：

(2-1) 暗号化および復号化処理スピードが速いこと.

(2-2) 平文と暗号文の比“ 平文/暗号文 ”が小さくならないようにすること.

(2-3) 平文空間が十分大きいこと.

(2-4) 公開鍵暗号として共通鍵暗号とのハイブリッドにしないこと（公開鍵暗号を実現させるために，共通鍵暗号を利用しない.）.

暗号学的仮定として利用する数論的問題としては，素因数分解問題または離散対数問題が理想に近いものであると考えた．なぜなら，それらの問題は十分な研究の下での計算量的困難性が予想されており [20, 25, 26]，また，従来，実用的なスキームにおいて暗号学的仮定として用いられる数論問題の 2 つのカテゴリー（素因数分解問題系と離散対数問題系）において最も難しい問題であることが理由である（すなわち，できる限り弱い仮定の下で安全性を証明できることが理想）.

素因数分解問題系： 素因数分解問題，RSA 問題，平方剰余問題，等,

離散対数問題系： 離散対数問題，Diffie-Hellman 計算問題，Diffie-Hellman 決定問題，等.

HIME(R) では効率面を考慮して素因数分解問題の困難性と等価の安全性を持つように設計する方針とした．そこで，モジュラー平方関数（Rabin 暗号化関数）に着目した．$N = pq$（p, q は素数）を法とするモジュラー平方関数の逆関数を求めることは N の素因数分解問題の困難性と等価であることが知られている以外にも，暗号化処理スピードが速いという利点がある．しかし，

(P-1) モジュラー平方関数は一方向性置換ではない（すなわち，復号化が一意的に行われない）.

(P-2) Rabin 暗号は，選択暗号文攻撃に対して安全でない.

(P-3) 復号化処理スピードが速くない（RSA 暗号と同程度）.

の問題があった．

そこで，(P-1) と (P-2) の問題を解決するために，OAEP [5] を利用した．OAEP は落し戸付き一方向性置換から得られる公開鍵暗号を IND-CCA1 に変換する方式である（当初，IND-CCA2 に変換できると考えられていたが，一般には IND-CCA1 であることが指摘された [36].）．OAEP を利用することで,（ランダムオラクルモデル上で）復号化一意性を確率的に保証することができ，また，Coppersmith のアルゴリズムを利用することにより，ランダムオラクルモデル上で IND-CCA2 であることが証明できた.（Rabin 暗号に OAEP を適用することにより，(P-1) と (P-2) を解決するアイデアは HIME-2 [22] において既に用いている．その後，OAEP とはパディングの方法が異なるが，Rabin-SAEP，Rabin-SAEP+ でも同様のことが行われている.）さらに，OAEP を利用することで，上記設計方針の (2-2)，(2-3) の条件もある程度クリアすることができる．特に，(2-3) の平文空間を大きく取ることについては，次の理由から重要であると考えた：公開鍵暗号の主たる目的は共通鍵暗号のデータ暗号化鍵の配送である．しかし，実システムでは，SET （Secure Electronic Transaction ）のように，データ暗号化鍵だけではなく付加情報（暗号の種類，ユーザの ID 情報，等）を一緒に送ることがシステムが多数存在するためである．

(P-3) の問題を解決するために，法とする合成数を $N = p^d q$ (p, q：素数，$d > 1$) として，これに応じた新しい計算方法を用いた．従来，このような合成数 N を法として高速な復号化を行う変形版 RSA 暗号が提案されている [37]．従来方法では，Chinese Remainder Theorem (CRT) を用いて $\mathbb{Z}_N$ を $\mathbb{Z}_{p^d}$ と $\mathbb{Z}_q$ の直積に分解し，$\mathbb{Z}_{p^d}$ において高速計算方法を利用した後，$\mathbb{Z}_q$ 上の計算結果とあわせて再度 CRT により張り合わせを行っている．HIME(R) ではモジュラー積計算の個数を減らすことを目的に，CRT を用いることのない計算方法を開発した（cf. 2.2.3 節）．これにより，従来方式に比べて次のメリットがある．

- モジュラー積計算の個数が減少した．
- 新方式では，CRT を使わないため Euclid の互除法による計算部分が不要になる．これにより，実測処理時間及び実装サイズの短縮が図れる．

この差は１回の復号化を通常のパソコンで実行した場合は殆ど無視できる差であるが，IC カード等の計算能力の低い媒体を使って計算する場合や一度の多くの復号化処理を必要とするシステムに適用した場合は無視できない差となって表れるものと予想される．

また，上記 (2-4) の公開鍵暗号として共通鍵暗号とのハイブリッド方式[2]を避けた理由としては，

(a) IC カード等のメモリが限られた媒体に実装する場合を考慮して，プログラムのサイズを大きくしないため．

(b) 共通鍵暗号のデータ暗号化鍵の配送に利用する場合，使用される共通鍵暗号アルゴリズムに依存させないため．例えば，ハイブリッド方式では，最悪の場合，２種類の共通鍵暗号アルゴリズムを用意する必要があり開発コストが問題になるケースが考えられる．

[2]例えば，素因数分解問題ベースの hybrid 方式としては，EPOC-2 [10], EPOC-3 [32] が挙げられる．

等が挙げられる.

以上のような方針で設計された HIME(R) は次の特徴を持つ.

(H-1) 合成数 $N = p^d q$ の素因数分解問題の困難性を仮定として，ランダムオラクルモデル上で IND-CCA2 の意味において安全であることが証明できる.

(H-2) 非常に高速な暗号化処理が可能（１回のモジュラー積のみ）.

(H-3) 復号化速度について，HIME(R) (1536 bits) は RSA-OAEP (1024 bits)[5] に対して，約 2.5 倍の高速処理が可能（モジュラー積の個数による比較）.

(H-4) RSA-OAEP と同等以上の十分大きな平文空間を持つ.

(H-5) 合成数 N のサイズを大きく選んでも，従来方式に比べ，暗号化・復号化の効率性を損なわない.

2.2 HIME(R) のアルゴリズム

以下，$|x|$ は x のビット長を表わす.

2.2.1 鍵生成手順

(K-1) $|p| = |q|$, $p \equiv 3 \pmod 4$, $q \equiv 3 \pmod 4$ なる十分大きな素数 p, q を選ぶ.

(K-2) $d > 1$ なる整数 d を選ぶ.

(K-3) $N = p^d q$ を計算する.

(K-4) $n = k - k_0 - k_1 - 1$, $2k_0 < k$ なる正整数 k_0, k_1, n を選ぶ．但し，$|N| = k$.

(K-5) ハッシュ関数 G, H

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

を選ぶ.

このとき,

秘密鍵： (p, q),

公開鍵： (N, k, k_0, k_1, G, H).

とする.

$N/2 < 2^{k-1} < N < 2^k$ が成立することに注意する.

また，各パラメータ k_0, k_1, k の選び方の詳細については，第 2.3 節において説明を行う.

2.2.2 暗号化手順

(E-1) メッセージ文 $m \in \{0,1\}^n$ に対して，乱数 $r \in \{0,1\}^{k_0}$ を選び，

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

を計算する．

(E-2) x に対して，

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

を計算する．

このとき，y をメッセージ文 m の暗号文とする．

2.2.3 復号化手順

(D-1) 暗号文 y に対して，

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

を検査し，y が $\mathbb{Z}_N$ 上で平方剰余であることを検査する．もし，そうでなければ，y をリジェクトする．

(D-2) 暗号文 y に対して，

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

にて，$\gamma_0, \gamma_1, \gamma_2, \dots, \gamma_d$ を計算する．但し，$1 \leq i \leq d$.

(D-3) $\gamma_0, \gamma_1, \gamma_2, \dots, \gamma_d$ に対して，

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

を計算する．このとき，γ_0 と γ_1 が各々２通りの値を取るため，合計４通りの x が計算される．これらを４個の x を x_1, x_2, x_3, x_4 と置く．

(D-4) 各 x_i $(1 \leq i \leq 4)$ に対して，$x_i \in \{0,1\}^{k-1}$ ならば，

$$x_i = s_i || t_i \qquad (s_i \in \{0,1\}^{n+k_1}, \quad t_i \in \{0,1\}^{k_0})$$

なる s_i, t_i を計算する．$x_i \notin \{0,1\}^{k-1}$ ならば，x_i をリジェクトする．

(D-5) s_i および t_i に対して，ハッシュ関数 G, H を用いて，

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

を計算する．

(D-6) w_i に対して，

$$w_i = m_i || z_i \qquad (m_i \in \{0,1\}^n, \quad z_i \in \{0,1\}^{k_1})$$

なる m_i, z_i を計算し，

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

を暗号文 y のメッセージ文として出力する．

2.2.4 復号化の正当性について

第 2.2.3 節で述べた復号化アルゴリズムでは，正しい暗号文に対して正しくメッセージ文が復号化されることを確率的に保証することができる．次の定理をこのことについて述べている．

定理 2.1. G および H を理想的ランダム関数と仮定する．このとき，無視できる確率を除いて，HIME(R) は正しく復号化される．

Proof. 第 2.2.3 節の (D-2) における x_1, x_2, x_3, x_4 は暗号文 y の $\mathbb{Z}_N$ 上での全ての平方根を与えることを示す．もし，これが示されれば，G および H は理想的ランダム関数より HIME(R) の復号化が失敗する確率は高々 $3/2^{k-1}$ であることから，定理 2.1 は明らかである．

d についての帰納法を用いる．また，$\mathbb{Z}_N$ $(N = p^d q)$ の元 x は，

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

なる形に一意的に書ける点に注意する．

$d = 2$ とする．このとき，$\mathbb{Z}_{p^2 q}$ の元 x に対して，$x = \gamma_0 + \gamma_1 p + \gamma_2 pq$ $(0 \leq \gamma_0, \gamma_2 < p,\ 0 \leq \gamma_1 < q)$ と置き，

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

が成立しているものと仮定する．このとき，

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

となり，これより，

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

が導かれる．よって，p および q は Blum 数であることから，

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

によって，γ_0, γ_1 が計算される ($y \bmod p$ と $y \bmod q$ は，それぞれ $\mathbb{Z}_p$ および $\mathbb{Z}_q$ 上で平方剰余であることを確認した上で)．

さらに，式 (1) から，γ_2 は，

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

により，計算される．ここで，$y - (\gamma_0 + \gamma_1 p)^2 \bmod p^2 q$ は pq で割り切れることに注意する．また，以上のことから，y が $\mathbb{Z}_{p^2 q}$ 上で平方剰余であることと，$y \bmod p$ および $y \bmod q$ が各々$\mathbb{Z}_p$ および $\mathbb{Z}_q$ 上で平方剰余であることが等価であることがわかった．

これより，$d = 2$ の場合，x_0, x_1, x_2, x_4 は $\mathbb{Z}_{p^2 q}$ における全ての平方根を与えることを示した．

次に，$d > 2$ とする．Γ_{d-2} $(= \gamma_0 + \gamma_1 p + \sum_{i=2}^{d-2} \gamma_i p^{i-1} q)$ は $\mathbb{Z}_{p^{d-2} q}$ における y の平方根の全てを与えるものと仮定する（Γ_{d-2} は，γ_0 と γ_1 が各々２通りの値を取るため，合計４通りの値を取る.)．

このとき，$x \in \mathbb{Z}_{p^{d-1} q}$ について，

$$x^2 \equiv y \pmod{p^{d-1} q} \qquad (2)$$

が成立するとき，仮定より x は，

$$x = \Gamma_{d-2} + \gamma_{d-1} p^{d-2} q \qquad (0 \le \gamma_d < p)$$

と一意的に書ける．よって，式 (2) から，

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

となり，γ_{d-1} は，

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

により，計算される．ここで，$y - {\Gamma_{d-2}}^2 \bmod p^{d-1} q$ は $p^{d-2} q$ で割り切れることに注意する．また，以上のことから，y が $\mathbb{Z}_{p^d q}$ 上で平方剰余であることと，$y \bmod p$ および $y \bmod q$ が各々$\mathbb{Z}_p$ および $\mathbb{Z}_q$ 上で平方剰余であることが等価であることを帰納的に容易に示すことができる．

以上のことから，定理 2.1 は証明された． □

2.2.5 補足事項

正しく復号化される精度を向上させることを目的に,

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

または，Jacobi 記号 $\beta = \left(\frac{x}{N}\right)$ を暗号文 y と一緒に送ることも可能である．特に，d が奇数の場合は，上記 α と β を送ることにより，復号化確率を 1 にすることができる．仮に，α と β を送っても，安全性証明に影響しないことに注意する．

2.3 鍵長について

HIME(R) におけるパラメータ k_0, k_1 および k について, まず $|k_0|, |k_1| \geq 128$ を推奨する.

表 1 に $N = pq$，$N = p^2q$ および $N = p^3q$ における N のビット長の比較をまとめる．各モジュラス長は数体ふるい法と楕円曲線法を用いて素因数分解を行った場合に同等の困難さを持つように決定している（詳細は，自己評価書に記述）.

表 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

RSA や RSA-OAEP は $N = pq$ タイプのモジュラスを用いており，HIME(R) は $N = p^2q$ や $N = p^3q$ タイプのモジュラスを用いる．表 1 により，素因数分解問題困難性を規準とした各々の方式におけるパラメータ（モジュラスのビット長）の比較が与えられる．例えば，1024 ビット RSA と 1344 ビット HIME(R)（$N = p^2q$）が対応し，2048 ビット RSA と 2304 ビット HIME(R)（$N = p^2q$）が対応する.

2.4 実装上の注意（Manger's Attack）

最近，Manger によって，インテグリティチェックを利用した PKCS #1 v2.0 に対する攻撃が発表されている [27]．HIME(R) についても，実装を行う上では，この攻撃方法を考慮して文献 [27] に記載の対策を施す必要がある．具体的には，HIME(R) の復号化手順において，与えられた暗号文の平方根のビット長が正しい（暗号文の）ビット長であるか否かを第三者に判らないような設計を行う．本件については，HIME(R) 固有の問題ではなく，実装設計における根本的問題でもあるので，本ドキュメントにおいてはその詳細については省略する.

3　提案方式の実装

本章では, HIME(R) 公開鍵暗号アルゴリズムの実現方法について説明する. なお, 実際の使用を想定して, $N = p^d q$ の d を 2 に設定する. また鍵長 (法長) を 1344-bit(素因子長を 488-bit) としている. さらにパラメータ k_0, k_1 はともに 128-bit とした.

記号

$x \,||\, y$: ビット列 x とビット列 y の連結 (例 : $(0110)||(101) = (0110101)$)
$x \oplus y$: ビット列 x とビット列 y の排他的論理和
$x \,\&\, y$: ビット列 x とビット列 y の論理積
$|x|$: ビット列 x の長さ
x^n : ビット列 x の上位 n ビット
x_n : ビット列 x の下位 n ビット
0^m : m 個の 0 からなるビット列 (例 : $0^5 = (00000)$)
$\{0,1\}^*$: 有限長のビット列の集合.
$\{0,1\}^i$: 長さ i のビット列の集合.
$\mathbf{Z}_n$: 法 n の剰余全体 $(= \{0, 1, 2, \ldots, n-1\})$
$a \bmod n$: 法 n での整数 a の剰余 $(\in \mathbf{Z}_n)$　(剰余の代表は $\{0, 1, 2, \ldots, n-1\}$ の中にとるものとする)

3.1　補助関数

提案方式を実際に計算機上に実装する場合には, 多倍長整数演算をはじめとする計算や, 乱数生成関数, 素数生成関数などが必要になる. 本章では, これら実装上で必要となる補助関数について示す.

3.1.1　多倍長整数

計算機上で計算可能な小さな数 (C 言語での int など) の配列を用いることにより, 大きな数を表現する (多倍長表現). つまり, 大きな整数 a を

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

$$(b : \text{基数}, 0 \leq a_i < b)$$

と分解し, a_i を配列 $A[i](0 \leq i \leq n)$ にそれぞれ格納する. この配列全体を 1 つの数として扱う.

整数 a が上記のように表現されているとき, a の最上位 bit(the most significant bit) とは a_n の最上位 bit をあらわす.

上記多倍長整数表現による整数四則演算, 剰余四則演算, べき乗剰余演算などの実現方法については, 文献 [28] を参照されたい. 剰余乗算については効率性向上のため Montgomery 法を用いることを推奨する. また, 剰余算における逆元計算や, べき乗剰余算についてもいくつかの効率的計算法が知られており ([28]), それらを利用することを推奨する.

3.1.2 乱数生成およびハッシュ関数

提案方式で使用する乱数は, まったくランダムな数値である真の乱数を用いることが望ましいが, 実装上, 一般的に利用されている疑似乱数生成関数を使用する. 具体的には, ANSI X9.17[1], X9.31[2], FIPS 186-1[16] などに示されている手法を使用することにより実現可能である.

鍵生成および暗号化処理で使用するハッシュ関数 G, H は, SHA-1[16] ハッシュ関数を使用することにより実現可能である. 詳細は 3.2 節を参照のこと.

3.1.3 素数生成

提案方式の法として用いる合成数 N には素因数分解が困難であることが要求される. そのためには素因子 p, q について以下のような条件を満足することが推奨されている.

- $p-1$ が大きな素数 r を含む.
- $p+1$ が大きな素数 s を含む.
- $r-1$ が大きな素数 t を含む.

（q も同様）

このような素数を以下では”強い素数”(”strong prime number”) と呼ぶ. ”強い素数”の生成方法に関しては, 文献 [28] 等に示されており, そちらを参照されたい.

3.1.4 数値表記

本仕様書中でいくつかの定数を用いているが, その表記はビット列として, 右が最下位ビット, 左が最上位ビットの 16 進表示とする.

3.2 The functions G, H

暗号化, 復号化で用いる関数 G, H を次のように定める.

h : the hash function SHA-1 $\{0,1\}^* \to \{0,1\}^{160}$
まず定数を定める (16 進表示):

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

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

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

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

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

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

Remarks.
1. G の出力値が整数として 2^{1215} 未満となるように最上位 bit が 0 となるようにした.
2. G, H の構成は [4] による.

3.3 鍵生成 ($d = 2$, $|N| = 1344$)

ある入力 ("seed") に対して"強い素数"を生成する関数 **PGen** が用意されているとする (3.1.3 節参照).

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 暗号化

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 復号化

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 ビット長

ここでは暗復号化で用いた変数の bit 長をまとめておく.

鍵

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

暗号化

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

復号化

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	$\|m\| = 1088$

参考文献

[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)