Sliding Window法の誤りつき演算情報を用いたCRT-RSA秘密鍵復元手法

大西 健斗[1,a)]　國廣 昇[2]

概要：本稿では，Sliding Window 法で実装された CRT-RSA 暗号方式に対し，サイドチャネル情報を用いた秘密鍵復元手法を提案する．Sliding Window 法は，二乗算と倍算を用いた，べき乗算の計算手法である．Bernstein らは，CHES 2017 において，Sliding Window 法の二乗算と倍算の実行履歴の取得手法を提案した．さらに，彼らは，実行履歴が正確に取得された状況下で，CRT-RSA 秘密鍵復元手法を提案し，window 幅 w が 4 以下ならば，秘密鍵復元が多項式時間であることを証明した．しかし，実際の実行履歴の取得には誤りが伴い，その場合，彼らの手法は，秘密鍵復元に失敗する．Oonishi と Kunihiro は，誤りつき実行履歴からの秘密鍵復元手法を提案したが，$w = 1, 2$ の場合でしか，多項式時間で復元可能な誤りの解析を行っていない．本稿では，まず，新たな秘密鍵復元手法を提案する．次に，$w = 1, 2$ では，より多くの誤りの下で，多項式時間で復元可能であることを証明し，$w = 3, 4$ においても，復元可能な誤りの解析を行う．さらに，数値実験による検証を行う．

キーワード：CRT-RSA 暗号方式，べき乗算，Sliding Window 法，秘密鍵復元，誤り訂正

Recovering CRT-RSA Secret Keys from Square-and-Multiply Sequences with errors in Sliding Window Method

Kento Oonishi[1,a)]　Noboru Kunihiro[2]

Abstract: We discuss side-channel attacks on the CRT-RSA encryption scheme implemented by Sliding Window method. Sliding Window method calculates exponentiations by repeating squaring and multiplication. In CHES 2017, Bernstein et al. proposed side-channel attacks of obtaining square-and-multiply sequences in Sliding Window method. Moreover, they proposed the method of recovering CRT-RSA secret keys from the correct square-and-multiply sequences, and they proved that their method recover secret keys in polynomial time when window size w is less than 4. However, there are errors in obtained square-and-multiply sequences, and their method fail to recover secret keys because of errors. Oonishi and Kunihiro proposed the method of recovering secret keys from square-and-multiply sequences with errors, but they analyze tolerable errors only when $w = 1, 2$. In this paper, we propose new method of recovering secret keys. We prove that we recover secret keys in polynomial time with more errors when $w = 1, 2$ than previous result, and we analyze tolerable errors when $w = 3, 4$. After that, we verify our analysis by numerical experiment.

Keywords: The CRT-RSA Encryption Scheme, Exponentiation, Sliding Window Method, Recovering Secret Keys, Error Correction

1. はじめに

1.1 研究背景

本稿では，CRT-RSA 暗号方式 [12] (中国人剰余定理 (CRT) を用いて復号の高速化を行った RSA 暗号方

[1] 東京大大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo
[2] 筑波大学大学院システム系
University of Tsukuba
[a)] kento_oonishi@mist.i.u-tokyo.ac.jp

式 [11]) に対する安全性解析，特に，サイドチャネル攻撃 [4] に対する安全性解析を行う．CRT-RSA 暗号方式 [12] は，現在広く利用されている公開鍵暗号方式である．CRT-RSA 暗号方式では，秘密鍵 d_p, d_q を用いたべき乗算 c^{d_p}, c^{d_q} を行うことで復号を行う．d_p, d_q は，正規の復号者のみが保持する値であり，公開鍵から導出するのは, 時間計算量の観点から困難である．

べき乗算は，上位ビットからべき指数を読み取り，二乗算と倍算の二種類の演算を，読み取ったべき指数の内容に応じて実行することで計算される．べき乗算の主なアルゴリズムとしては，Binary 法, Sliding Window 法, 2^m-ary 法がある．Binary 法では，べき指数を 1 ビットずつ読み取り，対応する演算を行う．Sliding Window 法および 2^m-ary 法では，window size なるパラメータ w に基づいて，複数のビットをまとめて読み取り，対応する演算を行う．なお，$w = 1$ の Sliding Window 法は，Binary 法に対応する．

Bernstein ら [1] は，サイドチャネル攻撃によって，べき乗算が Sliding Window 法で実装された CRT-RSA 暗号方式の秘密鍵を復元する手法を提案した．サイドチャネル攻撃とは，実装された暗号方式について，復号や署名生成等の処理にかかる実行時間 [4] や消費電力 [5]，キャッシュアクセス [13] 等の情報から，秘密鍵を復元する手法である．[1] では，まず，秘密鍵 d_p, d_q を用いたべき乗算 c^{d_p}, c^{d_q} に対し，Flush+Reload 攻撃 [13] によって，秘密鍵 d_p, d_q について生成される二乗算 (**S**) と倍算 (**M**) の実行履歴の取得を行う手法を提案した．以下，本稿では，この二乗算と倍算の実行履歴を SM 時系列と呼ぶ．さらに，彼らは，SM 時系列が正確に取得可能である仮定の下で，CRT-RSA 暗号方式における分枝限定法の枠組み (Heninger-Shacham の手法 [2]) を用いて，CRT-RSA 秘密鍵復元手法の提案を行い，$w \leq 4$ のとき，CRT-RSA 秘密鍵の復元が可能であることを証明した．彼らの手法では，下位ビット側から d_p, d_q の復元を行い，部分的に復元された d_p, d_q を SM 時系列に変換し，取得した SM 時系列の対応部分と比較を行って，一致しない場合に枝刈りを行っている．しかし，Sliding Window 法では，上位ビット側から d_p, d_q を読み取って SM 時系列を生成するため，d_p, d_q の正しい下位ビットが得られたとしても，取得された SM 時系列の対応部分とは一致しない可能性がある．そこで，彼らは，SM 時系列の正確性を用いて，上記の手法が必ず成功する時点で，枝刈りを行っている．

しかし，実際には，物理的な情報の観測における誤差や使用する機器の性能等の要因により，SM 時系列は誤りを含む [1]．Bernstein らの手法は，SM 時系列の正確性に基づいているため，SM 時系列が誤りを含む場合には，CRT-RSA 秘密鍵の復元に失敗する．

この問題を解決するために，Oonishi と Kunihiro は，SM 時系列が誤りを含む状況下で，正しい秘密鍵を復元する手法の提案を行った [8], [9]．まず， [9] では，Binary 法，つまり，Sliding Window 法の $w = 1$ の場合において，誤りつき SM 時系列を取得した状況について考察を行った．彼らは， [1] と同様に，Heninger-Shacham の手法 [2] に基づいた CRT-RSA 秘密鍵復元手法の提案を行うとともに，5.8% 以下の誤りの下で，CRT-RSA 秘密鍵を復元可能であることを証明した．彼らの手法では，下位ビット側から d_p, d_q の復元を行い，部分的に復元された d_p, d_q に基づいた枝刈りを行っており，$w = 1$ のみに成立する，d_p, d_q と SM 時系列の一対一対応性に基づいている．

[8] では，Sliding Window 法が，上位ビット側から d_p, d_q を読み取って SM 時系列を生成する点に着目し，新たに，上位ビット側から秘密鍵を復元する手法の提案を行った．[8] では，Sliding Window 法において，誤りつき SM 時系列を取得した場合に，CRT-RSA 秘密鍵復元手法の提案を行うとともに，$w = 2$ において，1.7%以下の誤りの下で，CRT-RSA 秘密鍵を復元可能であることを証明した．しかし，上位ビットから秘密鍵を復元する手法では，1 ビットずつ総当たり法を用いており，秘密鍵同士の数学的関係を十分に用いていないため，葉の生成の効率が落ちてしまう．さらに，彼らは，$w \geq 3$ に対する安全性解析を行っていない．したがって，$w \geq 3$ の場合について，安全性解析を行うことが重要である．

表 1 CRT-RSA 秘密鍵が復元可能な SM 時系列の誤り率の上限
Table 1 The Upper bound of Error Rate in Square-and-Multiply Sequences that We can Recover CRT-RSA Secret Keys.

w	1	2	3	4
誤り率 δ	0.108	0.067	0.034	0.008

1.2 研究成果

本稿では，まず， [8] の秘密鍵復元手法を改良し，下位ビット側から CRT-RSA 秘密鍵の復元を行う手法を構成することで，葉の生成に関して効率的な手法の提案を行う．さらに，提案アルゴリズムの解析を行うことにより，CRT-RSA 秘密鍵が復元可能であるような，SM 時系列の誤り率の上限の解析を行う．本稿では，SM 時系列の各要素が独立に，一定の確率 δ で反転すると仮定する．このとき，$w \leq 4$ なる各 w に対し，CRT-RSA 秘密鍵が復元可能であるような，SM 時系列の誤り率 δ の上限は，表 1 で与えられる．なお，$w \geq 5$ については，解析の結果，CRT-RSA 秘密鍵が復元可能であるような δ の範囲は存在しない．既存研究 [8], [9] では，$w = 1, 2$ において訂正可能な誤り率は，それぞれ 5.8 %，1.7 %であるため，既存研究よりも多くの誤りを訂正可能である．

最後に，本稿では，数値実験を行うことで，表 1 以下の誤りがあっても，CRT-RSA 秘密鍵を復元可能であることの

検証を行う．さらに，提案アルゴリズムを用いて，現実に生じる 1.1%の誤りの下での CRT-RSA 秘密鍵復元を行う．

2. 準備

本節では，まず，攻撃対象である CRT-RSA 暗号方式，およびその復号の実装に用いられるべき乗算 Sliding Window 法について述べる．その上で，本稿で用いる記号について述べる．

2.1 CRT-RSA 暗号方式 [12]

CRT-RSA 暗号方式は，RSA 暗号方式 [11] の復号を，中国人剰余定理 (CRT) を用いて高速化した公開鍵暗号方式である．通常の RSA 暗号方式は，公開鍵 (N, e) と秘密鍵 d で構成される．N は，相異なる $n/2$ ビットの奇素数 p と q の積とし，e は $(p-1)(q-1)$ と互いに素である自然数，d は $ed \equiv 1 \mod (p-1)(q-1)$ なる自然数とする．RSA 暗号方式では，平文 m の暗号化が $C = m^e \mod N$ によって行われ，復号は，$m = C^d \mod N$ で行われる．なお，公開鍵 e の値には，通常，素数 $2^{16}+1$ が用いられている．

CRT-RSA 暗号方式は，中国人剰余定理 (CRT) を用い，通常の RSA 暗号方式と比較して，復号の高速化を行った暗号方式である．ここでは，CRT-RSA 暗号方式の一種である PKCS#1 [12] について述べる．PKCS #1 では，通常の RSA 暗号方式の秘密鍵に加え，d_p, d_q, q_p を秘密鍵として用いる．d_p, d_q, q_p は，それぞれ，

$$d_p \equiv d \mod (p-1) \tag{1}$$
$$d_q \equiv d \mod (q-1) \tag{2}$$
$$q_p \equiv q^{-1} \mod p.$$

によって定義される．CRT-RSA 暗号方式では，公開鍵を (N, e)，秘密鍵を (p, q, d, d_p, d_q, q_p) とする．ここで，式 (1,2) より，

$$ed_p = 1 + k_p(p-1), \tag{3}$$
$$ed_q = 1 + k_q(q-1) \tag{4}$$

なる $k_p, k_q \in \mathbb{Z}$ が存在する．k_p, k_q は $0 < k_p, k_q < e$ を満たす [2].

次に，PKCS #1 の暗号化，復号について述べる．PKCS #1 の暗号化は，通常の RSA 暗号方式と同様である．PKCS #1 の復号はアルゴリズム 1 で行われる．アルゴリズム 1 では，通常の RSA 暗号方式と比較して，法およびべき指数のビット長が半分となっており，約 4 倍の高速化が行われる．

Algorithm 1 PKCS #1 の復号

Input：暗号文 $c \in \mathbb{Z}_N$，公開鍵 (N, e)
Output：平文 $m \in \mathbb{Z}_N$
$m_1 = c^{d_p} \mod p, m_2 = c^{d_q} \mod q$
$h = (m_1 - m_2)q_p \mod p$
$m = m_2 + qh$
return m

2.2 Sliding Window 法

Sliding Window 法は，べき乗算を計算する手法の一種である．Sliding Window 法のアルゴリズムは，以下のアルゴリズム 2 で与えられる．

Algorithm 2 Sliding Window 法 [7]

Input：$c, d = (d_t, d_{t-1}, \cdots, d_0)_2, w \geq 1$
Output：c^d
Precomputation
$c_1 = c, c_2 = c^2$
for $i = 1$ **to** $2^{w-1} - 1$
  $c_{2i+1} = c_{2i-1} \cdot c_2$
end for
Exponentiation
$A = 1, i = t$
While $i \geq 0$
  If $e_i = 0$
    $A = A^2$ (二乗算)
    $i = i - 1$
  Else
    $i - l + 1 \leq w$ かつ $e_l = 1$ であるような最長のビット列 $e_i e_{i-1} \cdots e_l$ を見つける．
    $A = A^{2(i-l+1)} \cdot c_{(e_i e_{i-1} \cdots e_l)_2}$ (倍算)
    $i = l - 1$
  End If
End While
Return A

Sliding Window 法では，まず，**Precomputation** ステップにおいて，小さなべき指数に対するべき乗算の結果を保存する．**Exponentiation** ステップでは，べき指数を上位ビットから読み取っていき，二乗算 (**S**) と倍算 (**M**) の二種類の演算を繰り返すことによって計算が行われる．なお，window size w を大きくすると，倍算として呼び出す値の候補が増え，メモリを多く利用する代わりに，倍算の回数が減少するため，計算が高速となる．

2.3 Sliding Window 法に対するサイドチャネル攻撃

アルゴリズム 1 では，べき乗算 c^{d_p}, c^{d_q} が行われており，Flush+Reload 攻撃 [13] を行うと，SM 時系列の取得が可能である [1]． [1] では，SM 時系列に，約 1.1%の誤りが生じている．以上を踏まえ，本稿では，べき乗算が Sliding Window 法で実装された場合に，べき乗算 c^{d_p}, c^{d_q} を構成する二乗算 (**S**) と倍算 (**M**) が，それぞれの演算について独立に，一定の確率 δ で反転すると仮定する．$\delta = 0$ の場合は，SM 時系列を誤りなく取得した状況に対応し，その下では，Bernstein ら [1] が提案した手法によって，$w \leq 4$ の場合に，CRT-RSA 秘密鍵を多項式時間で復元することが可能である．しかし，$\delta > 0$ のときは，SM 時系列が誤り

を含むため，Bernstein ら [1] の手法による CRT-RSA 秘密鍵復元は失敗する．

本稿では，誤りつきの SM 時系列を取得した状況下での，CRT-RSA 秘密鍵の復元手法について議論する．さらに，本稿では，提案アルゴリズムが，どの程度の誤り δ に適用可能か議論を行う．

2.4 使用する記号

本稿では，Heninger-Shacham [2] と同様の記号を利用する．まず，非負の整数 x が，二進数表示で $x = x_{n-1}x_{n-2}\cdots x_0$ と表せるとき，$x[i]$ を，$x[i] = x_i$ $(0 \leq i \leq n-1)$ と定義する．次に，$\tau(x)$ を，$\tau(x) = \max_{m\in\mathbb{Z}} 2^m | x$ と定義する．また，本稿の log の底は全て 2 であるとする．

3. 既存研究：Heninger-Shacham の手法 [2]

本節では，SM 時系列からの秘密鍵復元 [1], [9] に用いられている CRT-RSA 秘密鍵復元手法の枠組みである Heninger-Shacham の手法 [2] について紹介する．Heninger-Shacham の手法 [2] は，まず，公開鍵 (N, e) を用いて，(k_p, k_q) の組を求める．ここで，(k_p, k_q) の組は，$2(e-1)$ 通り存在する．次に，それぞれの (k_p, k_q) について，木構造を用いた分枝限定法を以下のように行う．

(1) **Initialize**：木の根を求める．
(2) **Expansion**：$n/2$ ビット分 p, q が求まるまで，以下を繰り返す．
- ビットを t ビット分求める．
- 枝刈りを行う．

(3) $N = pq$ なる秘密鍵を探索する．

まず，**Initialize** では，p, q, d_p, d_q の決定可能な下位ビットを求める．p, q は奇数なので，p, q の下位 1 ビットは 1 である．さらに，

$$ed_p \equiv 1 \mod 2^{\tau(k_p)+1}$$

$$ed_q \equiv 1 \mod 2^{\tau(k_q)+1}$$

より，d_p の下位 k_p ビット，d_q の下位 k_q ビットが求まる．以上を木の根とする．

次に，**Expansion** においては，p, q の下位 i ビット，d_p の下位 $\tau(k_p)+i$ ビット，d_q の下位 $\tau(k_q)+i$ ビットが求まっている状態で，

$$p' = \sum_{j=0}^{i-1} p[j]2^j, q' = \sum_{j=0}^{i-1} q[j]2^j,$$

$$d'_p = \sum_{j=0}^{i+\tau(k_p)-1} d_p[j]2^j, d'_q = \sum_{j=0}^{i+\tau(k_q)-1} d_q[j]2^j$$

としたうえで，p, q の下位 $i+1$ ビット目，d_p の下位 $\tau(k_p)+i+1$ ビット目，d_q の下位 $\tau(k_q)+i+1$ ビット目を，

$$p[i] + q[i] \equiv (N - p'q')[i], \tag{5}$$

$$d_p[i+\tau(k_p)] + p[i] \equiv \left(k_p(p'-1) + 1 - ed'_p\right)[i+\tau(k_p)], \tag{6}$$

$$d_q[i+\tau(k_q)] + q[i] \equiv \left(k_q(q'-1) + 1 - ed'_q\right)[i+\tau(k_q)] \tag{7}$$

によって求める．以上を繰り返すことによって，下位ビット側からビットの復元を行う．なお，上記の計算をより効率的に行うため，実際の実装では，**Expansion** を $i-1$ 回行った後の深さ $i-1$ の葉には，

- p, q の下位 i ビット
- d_p の下位 $\tau(k_p)+i$ ビット
- d_q の下位 $\tau(k_q)+i$ ビット

に加え，方程式 (5)–(7) の右辺に該当する

- $(N - p'q')/2^i$
- $\left(k_p(p'-1)+1-ed'_p\right)/2^{i+\tau(k_p)}$
- $\left(k_q(q'-1)+1-ed'_q\right)/2^{i+\tau(k_q)}$

が格納されている．

ここで，式 (5)–(7) は自由度 1 の方程式なので，ビットを 1 ビット計算するごとに，葉の数は 2 倍となる．したがって，もし，全く枝刈りを行わずに Heninger-Shacham の手法 [2] を適用すると，最終的な葉の数は $2^{n/2}$ 個，つまり指数個となる．ゆえに，適切な枝刈りを行うことが重要である．

4. 提案手法：誤りつき SM 時系列からの CRT-RSA 秘密鍵復元手法

本節では，誤りつき SM 時系列から CRT-RSA 秘密鍵の復元を行う手法の提案を行う．本稿で提案する手法は，Heninger-Shacham の手法 [2] に基づき，下位ビット側から d_p, d_q の計算を行い，秘密鍵の復元を行う．本稿では，特に，Heninger-Shacham の手法 [2] の **Expansion** を，誤りつき SM 時系列に適用できるよう変更した．以下，提案アルゴリズムの **Expansion** について述べる．

提案アルゴリズムでは，まず，式 (5)–(7) を用いて，p, q, d_p, d_q を 1 ビット分計算して，新たに葉を生成する．次に，生成されたそれぞれの葉について，取得された誤りつき SM 時系列との距離 D を計算する．距離 D は，d_p, d_q の下位ビットから生成される SM 時系列と，誤りつき SM 時系列の対応部分との不一致率によって定義する．ここで，d_p の下位 t ビットから生成される SM 時系列と，誤りつき SM 時系列の対応部分との不一致率を $D_{p,t}$，d_q の下位 t ビットから生成される SM 時系列と，誤りつき SM 時系列の対応部分との不一致率を $D_{q,t}$ とすると，d_p の下位 $\tau(k_p)+i$ ビット，d_q の下位 $\tau(k_q)+i$ ビットが格納されている葉と誤りつき SM 時系列との距離 D は，

$$D = \max\left(\min_{0\leq j\leq w-1} D_{p,\tau(k_p)+i-j}, \min_{0\leq j\leq w-1} D_{q,\tau(k_q)+i-j}\right)$$

表 2　$w=3$ での具体例
Table 2　Example in $w=3$.

	j	0	1	2
d_p	ビット	10110	0110	110
	計算系列	**SSSMSMS**	**SSSMS**	**SSMS**
	取得系列	**SMSSMMS**	**SSMMS**	**SMMS**
	不一致率	$3/7=0.429$	$1/5=0.2$	$1/4=0.25$
	$\min_t D_{p,t}$		0.2	
d_q	ビット	01101	1101	101
	取得系列	**SSSMSSM**	**SSMSSM**	**SSSM**
	観測系列	**SSMSMSM**	**SMSMSM**	**SMSM**
	不一致率	$3/7=0.429$	$3/6=0.5$	$1/4=0.25$
	$\min_t D_{q,t}$		0.25	
D	0.25			

によって定義される．距離 D の計算を行った後，距離 D が小さいほうから L 個の葉を残す．以上が，本稿で提案する CRT-RSA 秘密鍵復元アルゴリズムである．

ここで，$w=3$ の場合について，距離の計算法の具体例を述べる．まず，d_p の SM 時系列が **SSMSSMMS**，d_q の SM 時系列が **SSSMSMSM** と取得されているとする．さらに，ある葉において，d_p の下位 5 ビットが 10110，d_q の下位 5 ビットが 01101 と計算されたとする．ここで，この葉と，取得された SM 時系列の距離 D を計算すると，表 2 より，0.25 となる．なお，表 2 の計算系列は，ビットから変換された SM 時系列を表す．取得系列は，取得された SM 時系列のうち，計算系列と同じ数だけの，下位ビット側の SM 時系列に対応する．

5. 提案アルゴリズムの解析

本節では，以下の定理 1 の証明を行うことで，CRT-RSA 秘密鍵が復元可能であるような，SM 時系列の誤り率の上限が表 1 で与えられることを証明する．

定理 1　各 $w \le 4$ に対して，Y_w の値を表 1 のように与える．また，誤り率 δ が，$\delta < Y_w$ を満たすとする．このとき，提案アルゴリズムにおいて，一度に保存する葉の数を L 個とすると，時間計算量は $\max\left(O\left(n^2L\right), O\left(nL\log L\right)\right)$ であり，失敗確率の上限は，ある正の実数 ε に対して，

$$\frac{nw^2}{2L} + \frac{2\exp\left(2(w-1)\varepsilon^2\right)}{1-\exp\left(-2\varepsilon^2\right)} L^{-2\varepsilon^2\log e}$$

となる．

なお，既存研究 [8] では，同様に Sliding Window 法における CRT-RSA 秘密鍵復元アルゴリズムの提案を行っているが，時間計算量は $O\left(n^2L^2\right)$，失敗確率は $O\left(n^2/L\right)$ であるため，本稿の提案アルゴリズムは， [8] と比較して，時間計算量，成功確率ともに改良を行っている．

定理 1 において，L の値を大きくすると，時間計算量が大きくなる代わりに，成功確率が 1 に収束する．これは，保存する葉の数が多くなると，正しい情報を持つ葉が残りやすくなり，成功しやすくなることに対応する．特に，時間計算量が多項式時間となるような L は，以下の系 1 で与えられる．

系 1　提案アルゴリズムにおいて，一度に保存する葉の数を $L=n^{1+\gamma}$ $(\gamma>0)$ とする．このとき，$n\to\infty$ の下で，提案アルゴリズムの成功確率は 1 に収束し，時間計算量は多項式時間 $O\left(n^{3+\gamma}\right)$ となる．

なお，γ を大きくすると，L を大きくすることに対応するため，時間計算量が大きくなる代わりに，成功確率が 1 に収束する．本節では，まず，時間計算量の解析を行った後，成功確率の評価を行う．

5.1 時間計算量の解析

まず，木の深さ i の葉が求まっているときに，深さ $i+1$ の葉を求めるのにかかる時間計算量を見積もる．まず，方程式 (5)–(7) を解くのにかかる時間計算量は $O(n)$ である．次に，求まったビットを SM 時系列に変換するのにかかる時間計算量は $O(w)$ である．さらに，葉と取得系列の距離を計算するのにかかる時間計算量は $O(w)$ である．ゆえに，一つの葉の情報を計算するのにかかる時間計算量は $O(n)$ である．葉は $2L$ 個生成されるため，全ての葉の情報を計算するのにかかる時間計算量は $O(nL)$ である．したがって，p,q が i ビット求まっている状態で，$i+1$ ビット目を求めるのにかかる時間計算量は $O(nL)$ である．

次に，深さ $i+1$ の葉を求めた段階で，枝刈りを行うのにかかる時間計算量を見積もる．まず，$2L$ 個の葉をソートするのにかかる時間計算量は $O\left(L\log L\right)$ である．その後，距離 D について下位 L 個の葉を取り除くのにかかる時間計算量は $O(L)$ である．したがって，枝刈りを行うのにかかる時間計算量は $O\left(L\log L\right)$ である．

以上より，**Expansion** 一回分，つまり，深さ i の葉が求まっているときに，深さ $i+1$ の葉を求め，枝刈りを行うのにかかる時間計算量は $\max\left(O(nL), O\left(L\log L\right)\right)$ である．ここで，求めるビット数は $O(n)$ なので，**Expansion** 全体を通しての時間計算量は，$\max\left(O\left(n^2L\right), O\left(nL\log L\right)\right)$ となる．また，木の根を求める時間計算量は $O\left(\left(\log e\right)n\right)$ である．さらに，**Expansion** 終了後に，一つの葉を $N=pq$ がどうか判定するのにかかる時間計算量は $O(1)$ のため，L 個の葉を $N=pq$ がどうか判定するのにかかる時間計算量は $O(L)$ である．したがって，提案アルゴリズムの時間計算量は，$\max\left(O\left(n^2L\right), O\left(nL\log L\right)\right)$ となる．

5.2 失敗確率の評価

本稿では，提案アルゴリズムの失敗確率の評価を行う．既存の解析手法 [6], [10] では，共通する下位ビットを考慮

した仮定を置いたうえで解析を行なっていた．しかし，各葉において生成される SM 時系列は上位ビット側から生成されるため，共通の下位ビットを持つ二枚の葉であっても，SM 時系列は同じになるとは限らない．そこで，本稿では，提案手法の失敗確率の評価を行うにあたり，以下の仮定 1 を用いる．

仮定 1　各葉の SM 時系列は，葉ごとに独立に，ランダムビットに基づいて生成される．

ここで，提案アルゴリズムの失敗確率の評価を行う前に，まず，表 1 で与えられる誤りの下で，提案アルゴリズムの成功確率が高いことを，補題 1 を用いて，直感的に説明する．

補題 1　各 $w \leq 4$ に対して，Y_w の値を表 1 のように与え，$\delta \leq Y_w$ とする。ここで，ランダムに計算された t ビットのビット列を x_t とする．さらに，ランダムビットから生成され，かつ，誤り率 δ の誤りが含まれる SM 時系列 O_t が与えられたとする．ここで，x_t から変換された SM 時系列と O_t の不一致率を計算した際，不一致率が $\delta + \varepsilon_\delta$ 以下となる確率が $2^{-t/2}$ 以下であるような正の実数 ε_δ が存在する．

補題 1 より，木の深さ t において，誤り率 $\delta\,(\leq Y_w)$ の誤りが含まれ，SM 時系列との距離が δ 以下となる葉の数の期待値の上限は，$2^t\left(w2^{-t/2}\right)^2 = w^2$ となり，高々定数個となる．ここで，枝刈りの回数は高々 $n/2$ 回なので，L の値を $\Theta\left(n^{1+\gamma}\right)$ $(\gamma > 0)$ に設定すると，提案アルゴリズムの成功確率は，

$$\left(1 - \frac{1}{\Theta\left(n^{1+\gamma}\right)}\right)^{n/2} = 1 - O(n^{-\gamma})$$

となる．以下，補題 1 の証明を行う．

補題 1 の証明

はじめに，誤り率 δ の誤りが含まれる SM 時系列に対して，不一致率が Y 以下となる確率を評価する．t ビットのビット列からランダムに選んだ元を x_t とする．このとき，x_t を SM 時系列に変換した際の長さを l_{x_t}，x_t から変換された SM 時系列と O_t の誤り数を m_{x_t} とする．さらに，確率変数 Z_t を $Z_t = l_{x_t}Y - m_{x_t}$ と定義する．このとき，不一致率が Y 以下となる確率 P は

$$P = \Pr\left[Z_t \geq 0\right] \leq \inf_{s>0} \Pr\left[\exp\left(sZ_t\right) \geq 1\right] \leq \inf_{s>0} E\left[\exp\left(sZ_t\right)\right]$$

と評価できる．次に，各 w について，$E\left[\exp\left(sZ_t\right)\right]$ を

$$E\left[\exp\left(sZ_t\right)\right] = \begin{bmatrix}1 & 0 & \dots & 0\end{bmatrix} A_s^t \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}$$

表 3　行列 A_s の固有値
Table 3　The Eigenvalue of A_s.

w	1	2	3	4
s	1.7	1.9	2.1	3.1
λ_s	$2^{-0.5001}$	$2^{-0.5008}$	$2^{-0.5002}$	$2^{-0.5026}$

と書き表す．行列 A_s は，X_t, O_t のマルコフ性を用いて計算する．例えば，$w = 1$ の場合，$T = (1-\delta) + \delta\exp(-s)$，$F = (1-\delta)\exp(-s) + \delta$ の下で，A_s は 6 次元の行列

$$\begin{bmatrix}
\frac{T}{4}e^{sY} & \frac{T}{4}e^{sY} & 0 & \frac{TF}{8}e^{2sY} & \frac{TF}{8}e^{2sY} & \frac{T^2}{4}e^{2sY}\\
0 & 0 & \frac{F}{2}e^{sY} & \frac{F^2}{4}e^{2sY} & \frac{F^2}{4}e^{2sY} & 0\\
\frac{T}{4}e^{sY} & \frac{T}{4}e^{sY} & 0 & \frac{TF}{8}e^{2sY} & \frac{TF}{8}e^{2sY} & \frac{T^2}{4}e^{2sY}\\
\frac{T}{4}e^{sY} & \frac{T}{4}e^{sY} & 0 & \frac{TF}{8}e^{2sY} & \frac{TF}{8}e^{2sY} & \frac{T^2}{4}e^{2sY}\\
0 & 0 & \frac{F}{2}e^{sY} & \frac{F^2}{4}e^{2sY} & \frac{F^2}{4}e^{2sY} & 0\\
\frac{T}{4}e^{sY} & \frac{T}{4}e^{sY} & 0 & \frac{TF}{8}e^{2sY} & \frac{TF}{8}e^{2sY} & \frac{T^2}{4}e^{2sY}
\end{bmatrix}$$

となる．同様に，各 w について行列 A_s を生成すると，$w = 2$ の場合は 20 次元，$w = 3$ の場合は 64 次元，$w = 4$ の場合は 176 次元の行列となる．行列 A_s のジョルダン標準形を J_s とすると，$A_s = QJQ^{-1}$ なる正則な正方行列 Q が存在し，$A_s^t = QJ^tQ^{-1}$ となるので，

$$E\left[\exp\left(sZ_t\right)\right] = \left(\begin{bmatrix}1 & 0 & \dots & 0\end{bmatrix} Q\right) J^t \left(Q^{-1}\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}\right)$$

となるため，行列 A_s の絶対値最大の固有値を λ_s とすると，

$$E\left[\exp\left(sZ_t\right)\right] \sim \left(\inf_{s>0}\lambda_s\right)^t$$

となる．ゆえに，$P \leq \left(\inf_{s>0}\lambda_s\right)^t$ となるため，補題 1 で与えた δ および $Y = \delta$ について，$\inf_{s>0}\lambda_s < 2^{-1/2}$ となることを証明すればよい．

しかし，$\inf_{s>0}\lambda_s$ を解析的に求めることは困難であるため，本稿では，Matlab の固有値ソルバーを用いて，$\inf_{s>0}\lambda_s$ を数値的に計算する．はじめに，表 1 について補題 1 が正しいことを述べる．まず，δ を，表 1 の値として，$Y = \delta$ とする．次に，s の値を表 3 の通りに定める．さらに，上記で定めた δ, Y, s に基づいて，行列 A_s を生成する．そのうえで，Matlab の固有値ソルバーを用いて，λ_s を求める．数値計算の結果は，表 3 の通りである．ここで，表 3 は，全て $\lambda_s < 2^{-1/2}$ である．なお，δ の値を表 1 よりも小さくして，$Y = \delta$ とした場合，同様に，$\inf_{s>0}\lambda_s < 2^{-1/2}$ であることが検証できる．また，δ の値を表 1 よりも大きくして，$Y = \delta$ とした場合，$\inf_{s>0}\lambda_s > 2^{-1/2}$ となる．したがって，補題 1 で与えた δ および $Y = \delta$ について，確率 P は $2^{-t/2}$ 未満となる．以上より，補題 1 が示せた．■

ここからは，提案アルゴリズムの失敗確率を計算する．

まず，深さ t において CRT-RSA 秘密鍵の正確な部分情報を含む葉が枝刈りされる確率 P_t を評価する．深さ t において生成される葉の数は 2^t 個である．ここで，CRT-RSA 秘密鍵の部分情報を正しく含む葉を X_1 とし，それ以外の葉を，X_i $(2 \le i \le 2^t)$ とする．さらに，各葉 X_i と取得された SM 時系列との距離を C_i とする．このとき，[6] と同様の不等式評価を行うと，$\varepsilon = \min(\varepsilon_\delta, Y_w - \delta)$ の下で，

$$\begin{aligned} P_t &\le \Pr\left[\sum_{i=2}^{2^t} \mathbf{1}\left[C_i \le C_1\right] \ge L\right] \\ &\le \Pr\left[\left(\sum_{i=2}^{2^t} \mathbf{1}\left[C_i \le \delta+\varepsilon\right] \ge L\right) \cup \left(C_1 \ge \delta+\varepsilon\right)\right] \\ &\le \Pr\left[\sum_{i=2}^{2^t} \mathbf{1}\left[C_i \le \delta+\varepsilon\right] \ge L\right] + \Pr\left[C_1 \ge \delta+\varepsilon\right] \\ &\le \frac{1}{L} E\left[\sum_{i=2}^{2^t} \mathbf{1}\left[C_i \le \delta+\varepsilon\right]\right] + \Pr\left[C_1 \ge \delta+\varepsilon\right] \\ &\le \frac{1}{L}\sum_{i=2}^{2^t} \Pr\left[C_i \le \delta+\varepsilon_\delta\right] + \Pr\left[C_1 \ge \delta+\varepsilon\right] \qquad (8) \end{aligned}$$

となる．ここで，式 (8) の第一項について，補題 1 より，

$$\frac{1}{L}\sum_{i=2}^{2^t} \Pr\left[C_i \le \delta+\varepsilon_\delta\right] \le \frac{2^t-1}{L}\left(w2^{-t/2}\right)^2 \le \frac{w^2}{L} \quad (9)$$

となる．次に，式 (8) の第二項の評価を行うために，定理 2 を導入する．

定理 2　[3] $X_1, \ldots, X_n$ が独立にパラメータ p のベルヌーイ分布に従う確率変数であるとする．このとき，$X = \sum_{i=1}^{n} X_i$ と定めると，

$$\Pr[X \ge n(p+\gamma)] \le \exp\left(-2n\gamma^2\right)$$

である．

ここで，深さ t の葉に対応する SM 時系列の長さは，少なくとも $t-w+1$ なので，片方の秘密鍵に注目したとき，誤り率 δ の下で，不一致率が $\delta+\varepsilon$ 以上となる確率は，高々 $\exp\left(-2(t-w+1)\varepsilon^2\right)$ である．ゆえに，

$$\Pr\left[C_1 \ge \delta+\varepsilon\right] \le 2\exp\left(-2(t-w+1)\varepsilon^2\right) \qquad (10)$$

となる．したがって，式 (8)，(9)，(10) より，

$$P_t \le \frac{w^2}{L} + 2\exp\left(-2(t-w+1)\varepsilon^2\right)$$

となる．枝刈りは $(\lfloor \log L \rfloor + 1) \le t \le n/2$ の際に行われるため，失敗確率 P の上限は，

$$\begin{aligned} P &= \sum_{t=\lfloor \log L \rfloor+1}^{n/2} P_t \\ &\le \sum_{t=\lfloor \log L \rfloor+1}^{n/2} \left(\frac{w^2}{L} + 2\exp\left(-2(t-w+1)\varepsilon^2\right)\right) \\ &\le \frac{nw^2}{2L} + \frac{2\exp\left(2(w-1)\varepsilon^2\right)\exp\left(-2\left(\lfloor \log L \rfloor+1\right)\varepsilon^2\right)}{1-\exp\left(-2\varepsilon^2\right)} \\ &\le \frac{nw^2}{2L} + \frac{2\exp\left(2(w-1)\varepsilon^2\right)\exp\left(-2\varepsilon^2 \log L\right)}{1-\exp\left(-2\varepsilon^2\right)} \\ &\le \frac{nw^2}{2L} + \frac{2\exp\left(2(w-1)\varepsilon^2\right)}{1-\exp\left(-2\varepsilon^2\right)} L^{-2\varepsilon^2 \log e} \end{aligned}$$

となる．

表 4　$w=1, L=1024$ における実験結果
Table 4　Experimental Result in $w=1, L=1024$.

δ	0	0.04	0.08	0.10	0.12
成功率 (%)	100	88	31	7	1
実行時間 (s)	8.76	8.11	9.24	9.13	10.8

表 5　$w=2, L=1024$ における実験結果
Table 5　Experimental Result in $w=2, L=1024$.

δ	0	0.02	0.04	0.06	0.08	0.1
成功率 (%)	100	89	45	16	1	0
実行時間 (s)	10.9	11.2	10.1	9.85	8.06	–

表 6　$w=3, L=1024$ における実験結果
Table 6　Experimental Result in $w=3, L=1024$.

δ	0	0.01	0.02	0.03	0.04	0.05
成功率 (%)	100	84	42	23	7	0
実行時間 (s)	9.23	9.10	10.0	9.54	13.0	–

表 7　$w=4, L=1024$ における実験結果
Table 7　Experimental Result in $w=4, L=1024$.

δ	0	0.005	0.01	0.015	0.02
成功率 (%)	100	58	29	7	5
実行時間 (s)	11.5	11.3	11.8	9.92	15.6

6. 数値実験

本節では，提案した CRT-RSA 秘密鍵復元手法の数値実験を行う．本稿では，

- 表 1 の誤りが訂正可能であることの検証
- 実際の誤り率 $\delta = 0.011$ が訂正可能であることの検証

を行った．

本稿では，NTL ライブラリのバージョン 11.3.2 を用い，C++を用いて数値実験を行った．なお，本稿では，16GB のメモリを持つ 2.70GHz の Intel Core i7 を用いて数値実験を行った．本稿では，まず，w, δ, L の値を設定した後，それぞれの (w, δ, L) の組に対して，$n=1024$，つまり，p, q が 512 ビットの CRT-RSA 秘密鍵をランダムに 100 組生成

表 8　$w = 1, \delta = 0.011$ における実験結果
Table 8　Experimental Result in $w = 1, \delta = 0.011$.

L	8	16	32
成功率 (%)	87	98	100
実行時間 (s)	0.0961	0.156	0.359

表 9　$w = 2, \delta = 0.011$ における実験結果
Table 9　Experimental Result in $w = 2, \delta = 0.011$.

L	256	512	1024
成功率 (%)	94	99	100
実行時間 (s)	2.65	5.57	10.7

表 10　$w = 3, \delta = 0.011$ における実験結果
Table 10　Experimental Result in $w = 3, \delta = 0.011$.

L	1024	2048	4096
成功率 (%)	72	82	84
実行時間 (s)	11.5	22.3	44.2

表 11　$w = 4, \delta = 0.011$ における実験結果
Table 11　Experimental Result in $w = 4, \delta = 0.011$.

L	1024	2048	4096
成功率 (%)	18	35	37
実行時間 (s)	14.9	25.0	41.5

する．そのうえで，それぞれの (w, δ, L) の組に対して，提案手法の成功時の平均実行時間および成功率を測定する．なお，本節では，(k_p, k_q) の値は既知であるという仮定の下で，数値実験を行った．

6.1　表 1 の誤りが訂正可能であることの検証

まず，表 1 の誤りが訂正可能であることの検証を行う．本小節では，$L = n^{1+\gamma}$ に基づき，L の値を 1024 に固定して，数値実験を行った．実験結果は，表 4–7 の通りである．

ここで，表 4–7 より，表 1 よりも小さい誤りについて，CRT-RSA 秘密鍵の復元に成功している．したがって，表 1 の誤りが確かに訂正可能であることが検証された．

6.2　実際の誤り率 $\delta = 0.011$ での数値実験

次に，SM 時系列の実際の誤りである，1.1%の誤り [1] について数値実験を行った．本小節では，δ の値を 0.011 に固定し，L の値を変化させ，数値実験を行った．実験結果は，表 8–11 の通りである．

まず，表 8 および 9 より，$w = 1$ では 0.4 秒程度で，$w = 2$ では 10 秒程度で，全ての CRT-RSA 秘密鍵の復元に成功している．さらに，表 10 より，$w = 3$ では，約 40 秒程度で 8 割程度の CRT-RSA 秘密鍵の復元に成功している．定理 1 より，理論上，$w \leq 3$ では，SM 時系列に 1.1%の誤りがあっても CRT-RSA 秘密鍵の復元が可能なため，上記の結果は理論と合致している．

次に，表 11 より，$w = 4$ では，約 40 秒程度で 4 割程度の CRT-RSA 秘密鍵の復元に成功している．定理 1 では，$w = 4$ において，SM 時系列に 1.1%の誤りがある場合については，CRT-RSA 秘密鍵の復元が可能かどうか不明であるが，実験では CRT-RSA 秘密鍵の復元が可能である．

以上より，本稿で提案したアルゴリズムは，SM 時系列の実際の 1.1%の誤りについても，CRT-RSA 秘密鍵の復元が可能である．

謝辞　本研究の一部は，JST CREST JPMJCR14D6 の支援および JSPS 科研費 JP16H02780 の助成を受けて行われた．

参考文献

[1] Bernstein, D.J., Breitner, J., Genkin, D., Groot-Bruinderink, L., Heninger, N., Lange, T., van Vredendaal, C., Yarom, Y.: “Sliding Right into Disaster: Left-to-Right Sliding Windows Leak,” CHES 2017, LNCS, vol. 10529, pp. 555–576 (2017).

[2] Heninger, N., Shacham, H.: “Reconstructing RSA Private Keys from Random Key Bits,” CRYPTO 2009, LNCS, vol. 5677, pp. 1–17 (2009).

[3] Hoeffding, W.: “Probability Inequalities for Sums of Bounded Random Variables,” Journal of the American Statistical Association, vol. 58, No. 301, pp. 13–30 (1963).

[4] Kocher, P.C.: “Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems,” CRYPTO 1996, LNCS, vol. 1109, pp. 104–113 (1996).

[5] Kocher, P., Jaffe, J., Jun, B.: “Differential Power Analysis,” CRYPTO 1999, LNCS, vol. 1666, pp. 388–397 (1999).

[6] Kunihiro, N., Honda, J.: “RSA Meets DPA: Recovering RSA Secret Keys from Noisy Analog Data,” CHES 2014, LNCS, vol. 8731, pp. 261–278 (2014).

[7] Menezes, A.J.,van Oorschot, P.C., Vanstone, S.A.: *Handbook of Applied Cryptography*, CRC Press, Boca Raton, Florida (1996).

[8] 大西 健斗，國廣　昇：“Sliding Window 法の誤りつき演算情報を用いた CRT-RSA 秘密鍵復元アルゴリズム,” SCIS 2018 (2018).

[9] Oonishi, K., Kunihiro, N.: “Attacking Noisy Secret CRT-RSA Exponents in Binary Method,” ICISC 2018, LNCS, vol. 11396, pp. 37–54 (2019).

[10] Paterson, K.G., Polychroniadou, A., Sibborn, D.L.: “A Coding-Theoretic Approach to Recovering Noisy RSA Keys,” ASIACRYPT 2012, LNCS, vol. 7658, pp. 386–403 (2012).

[11] Rivest, R.L., Shamir, A., Adleman, L.: “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” Commun. ACM., vol. 21, pp. 120–126 (1978).

[12] RSA Laboratories: “PKCS#1 v2.2: RSA Cryptography Standard,” `https://www.emc.com/collateral/white-papers/h11300-pkcs-1v2-2-rsa-cryptography-standard-wp.pdf` (2012).

[13] Yarom, Y., Falkner, K.: “FLUSH+RELOAD: A High Resolution, Low Noise, L3 Cache Side-Channel Attack,” USENIX 2014, pp. 719–732 (2014).