剰余演算系基底のランダム選択手法の電力解析に対する脆弱性

小池正修[†,††] 松本勉[†]

剰余演算系でのモンゴメリ乗算手法へのサイドチャネル解析対策として Leak Resistant Arithmetic（LRA）が知られている．LRA は剰余演算系の基底をランダムに選択することで，データをランダム化する手法である．そのため LRA は，従来から知られているサイドチャネル解析対策手法である，メッセージブラインディングの代替になるとされている．本論文ではサイドチャネル解析対策として LRA のみを採用している RSA 実装に対する電力解析手法を提案する．提案手法は，特別な入力を与えることで，数回のベキ乗剰余算の電力波形から秘密指数を導出する．その一方で，提案手法はメッセージブラインディングで防げることも示す．したがって LRA はメッセージブラインディングの代替にはならず，他の対策と併用する必要があるといえる．

Vulnerability over Power Analysis of Randomized Base Selection Method of Residue Number Systems

Masanobu Koike[†,††] and Tsutomu Matsumoto[†]

In this paper we show a side-channel attack on RSA based on Residue Number Systems (RNS) with the Leak Resistant Arithmetic (LRA) countermeasure. The LRA, a random selection method of RNS bases, is proposed to resist some side-channel attacks. It has been believed that the LRA randomizes input data as well as classical message blinding method and that it provides a protection against power analysis. We propose a simple power analysis on RSA with the LRA countermeasure under some conditions. It reveals the secret exponent by a few power traces. On the other hand, it is prevented by classical message blinding method. Hence we show a counterexample of the expectation above. We conclude that the LRA cannot replace classical message blinding method and that we should use the LRA in combination with classical countermeasures.

1. はじめに

多倍長整数の高速演算手法として，剰余演算系（Residue Number Systems，以下 RNS と略記）の利用が知られている．RNS は整数を小さな整数の組で表現する手法で，ハードウェアによる並列処理に向いている．Posch らは RNS とモンゴメリ乗算[15] を組み合わせた，RNS モンゴメリ乗算アルゴリズムを提案した[18]．それ以後，アルゴリズムの改良[2),10)] や，RNS モンゴメリ乗算を利用したベキ乗剰余算の実装についての報告がされている[5),6),16)]．

その一方で，装置等に実装された暗号アルゴリズムへの攻撃手法としてサイドチャネル解析が脅威となっている．サイドチャネル解析は，装置が暗号処理中に副次的に漏洩する情報（処理時間，消費電力，電磁波等）を解析し，装置内の秘密情報を推測する解析手法である．サイドチャネル解析の研究は近年さかんに行われ，多くの解析手法，対策が提案されている[1),7),8),11),14),17),21)]．実装の安全性を評価する ISO/IEC 15408[9] にもサイドチャネル解析に関する項目がある．現時点では対象となる装置は IC カードが主流であるが，将来的にはその他の暗号処理装置も同様の評価の対象となるものと予想される．そこで本論文では，RNS モンゴメリ乗算を実装した文献 5), 6) や 16) の装置を対象として，サイドチャネル解析について考察する．

筆者らが知る限り，RNS モンゴメリ乗算にサイドチャネル解析への耐性を持たせる手法は，2 つ知られている[4)~6)]．どちらも RNS 基底をランダムに選ぶというアイディアに基づいており，違いは用意する RNS 基底の個数と基底拡張アルゴリズムである．Ciet らの手法[5),6)] は，RNS 基底を generalized Mersenne num-

† 横浜国立大学大学院環境情報学府/研究院
Graduate School of Environment and Information Sciences, Yokohama National University
†† 東芝ソリューション株式会社 SI 技術開発センター
Systems Integration Technology Center, Toshiba Solutions Corporation

ber（$2^{\kappa_1}-2^{\kappa_2}-1$ の形）からランダムに選ぶ方法で，これらの数をあらかじめ RNS モンゴメリ乗算で必要とする以上の個数を準備しておく．基底拡張アルゴリズムは文献 2) の手法を採用している．一方 Bajard らの手法[4] では，RNS 基底の集合は必要最小限だけ用意し，そこからランダムに基底を選択する手法である．基底拡張アルゴリズムは Garner のアルゴリズムを利用している．Bajard らは提案手法を Leak Resistant Arithmetic（以下 LRA と略記）と呼び，空間レベルおよびデータレベルでのランダム化の効果があると主張している．空間レベルのランダム化は電磁波解析への耐性を，データレベルのランダム化はタイミング解析および電力解析への耐性を高める効果があるとしている．

文献 4)〜6) は，通常の基数表現に対するサイドチャネル解析を対象として，RNS の特長を活かした対策手法を提案している．しかしながら解析手法については何も言及されておらず，逆に RNS 特有の性質を利用したサイドチャネル解析についてはまったく考慮されていない．

本論文では RNS 特有の性質に注目し，ある条件下で LRA に対する電力解析手法を提案する．すなわち LRA によるランダム化はサイドチャネル解析対策としては不十分であることを示す．したがって LRA は他の対策と併用しなければならないといえる．

2. サイドチャネル解析

ここでは**図 1** に示す，バイナリ法によるベキ乗剰余算 $m=c^d \bmod N$ に対する電力解析を復習する．本論文では d は RSA[19] の秘密ベキ指数とする．図 1 のステップ 3，4 の剰余乗算をそれぞれ 2 乗算，乗算と呼ぶことにする．多くの電力解析手法は，乗算が $d_i=1$ のときのみに実行されることに注目し，2 乗算と乗算を区別することで d_i の値を順次推測して秘密指数 d を暴く[14),21)]．

その 1 つの手法として，消費電力はそのタイミングで扱っているデータのハミング重みに比例するという原理に注目した解析手法が提案されている[14]．図 1 の $i=j$ でのループにおけるステップ 4 の乗算が行われるか否かに応じて，$i=j-1$ でのループにおけるステップ 3 の 2 乗算への入力のハミング重みが異なるような入力 c を与えることで，秘密指数 d を推測する手法である．

特にデータが 0 の場合はハミング重みが 0 のため，消費電力が小さくなる．この考えに基づいた解析手法として，文献 1)，8)，17) が知られている．文献 1)，8) は楕円曲線暗号において，文献 17) は RSA 等における中国剰余定理（Chinese Remainder Theorem，以下 CRT と略記）を利用した実装手法に対し，値が 0 となるデータを演算中に出現させることによって秘密情報を推測する解析手法である．

Input: $c, N, d=\sum_{i=0}^{\ell-1} d_i 2^i$ $(d_i \in \{0,1\})$
Output: $m=c^d \bmod N$

1: $m=1$
2: for $(i=\ell-1$ downto $0)$ {
3: $\quad m=m^2 \bmod N$
4: $\quad$ if $(d_i=1)$ then $m=mc \bmod N$
5: }

図 1 バイナリ法によるベキ乗剰余算
Fig. 1 Modular exponentiation using binary method.

以上述べた解析手法は選択入力攻撃，すなわち攻撃者が都合の良い値を入力できる攻撃である．一般に RSA は，入力データに対してハッシュ関数を作用させたり，ランダムなパディングをしたりするため，ベキ乗剰余算のオペランドを外部から自由に操作できない場合もある[20]．しかし OAEP[20] による RSA 復号操作では，入力データをそのままベキ乗剰余算に用いるため，入力データを外部から自由に操作することが可能である．以下では RSA 復号を考察の対象とし，入力値は攻撃者が自由に選択できるものとする（以下，このような攻撃を選択暗号文攻撃と呼ぶ）．

上述のような選択暗号文攻撃は，一般的にメッセージブラインディングにより防御可能である．RSA に対するメッセージブラインディング手法は次のとおりである[11]．まず $\sigma=(\tau^{-1})^e \bmod N$ を満たす乱数の組 (σ,τ) を用意する．ここで e は RSA の公開ベキ指数である．暗号文 c に σ を乗じた $c\sigma \bmod N$ に対してベキ乗剰余算を行い，得られた結果に τ を乗じて正しい出力 m を得る．図 1 のループ内では値がランダム化されているため，選択暗号文攻撃を防御できる．

3. RNS モンゴメリ乗算

RNS は整数 x を表現する一手法として知られている．$\mathcal{B}_1=\{b_1,\ldots,b_n\}$ をどの 2 つも互いに素な整数の集合とする．整数 x $(0 \le x < B_1=\prod_{i=1}^{n} b_i)$ の RNS 表現は，剰余の組

$$\langle x \rangle_{\mathcal{B}_1}=(x[b_1],\ldots,x[b_n])$$

である．ここで $x[b_i]=x \bmod b_i$ である．集合 $\mathcal{B}_1$ を RNS 基底，n を基底のサイズと呼ぶ．さらに b_i のビット長はすべて r と仮定する．一般に r は任意の値が可能であるが，現在の演算器のビット幅の主流が

Function: $w = MM(x, y, N, \mathcal{B}_1, \mathcal{B}_2)$		
Input: $\langle x\rangle_{\mathcal{B}_1\cup\mathcal{B}_2}, \langle y\rangle_{\mathcal{B}_1\cup\mathcal{B}_2}$		
Output: $\langle w\rangle_{\mathcal{B}_1\cup\mathcal{B}_2} (w \equiv xyB_1^{-1} \pmod N)$		
	Base $\mathcal{B}_1$ operation	Base $\mathcal{B}_2$ operation
1:	$\langle s\rangle_{\mathcal{B}_1} \leftarrow \langle x\rangle_{\mathcal{B}_1}\langle y\rangle_{\mathcal{B}_1}$	$\langle s\rangle_{\mathcal{B}_2} \leftarrow \langle x\rangle_{\mathcal{B}_2}\langle y\rangle_{\mathcal{B}_2}$
2:	$\langle t\rangle_{\mathcal{B}_1} \leftarrow \langle s\rangle_{\mathcal{B}_1}\langle(-N)^{-1}\rangle_{\mathcal{B}_1}$	
3:	$\langle t\rangle_{\mathcal{B}_1} \Longrightarrow \langle t\rangle_{\mathcal{B}_1\cup\mathcal{B}_2}$	
4:		$\langle u\rangle_{\mathcal{B}_2} \leftarrow \langle t\rangle_{\mathcal{B}_2}\langle N\rangle_{\mathcal{B}_2}$
5:		$\langle v\rangle_{\mathcal{B}_2} \leftarrow \langle s\rangle_{\mathcal{B}_2} + \langle u\rangle_{\mathcal{B}_2}$
6:		$\langle w\rangle_{\mathcal{B}_2} \leftarrow \langle v\rangle_{\mathcal{B}_2}\langle B_1^{-1}\rangle_{\mathcal{B}_2}$
7:	$\langle w\rangle_{\mathcal{B}_1\cup\mathcal{B}_2} \Longleftarrow \langle w\rangle_{\mathcal{B}_2}$	

図 2 RNS モンゴメリ乗算アルゴリズム
Fig. 2 RNS Montgomery multiplication algorithm.

32 であるため，本論文では $r = 32$ を念頭におく．

CRT により，RNS 表現された $\langle x\rangle_{\mathcal{B}_1}$ から x を求める式は

$$x = \rho_1 + b_1(\rho_2 + b_2(\rho_3 + \cdots + b_{n-1}\rho_n)\cdots) \quad (1)$$

で与えられる．ただし ρ_i は $\theta_{i,j} = b_i^{-1} \bmod b_j$ として

$$\begin{aligned}
\rho_1 &= x[b_1] \\
\rho_2 &= (x[b_2] - \rho_1)\theta_{1,2}[b_2] \\
&\vdots \\
\rho_n &= (\cdots(x[b_n] - \rho_1)\theta_{1,n} - \cdots - \rho_{n-1})\theta_{n-1,n}[b_n]
\end{aligned}$$

という値である．式 (1) は Garner のアルゴリズムと呼ばれる[13]．また

$$\begin{aligned}
x &= \sum_{i=1}^{n}(x[b_i]B_{1,i}^{-1}[b_i])B_{1,i} \bmod B_1 \\
&= \sum_{i=1}^{n}(x[b_i]B_{1,i}^{-1}[b_i])B_{1,i} - kB_1 \qquad (2)
\end{aligned}$$

で与えられる Gauss のアルゴリズムもよく用いられる．ここで $B_{1,i} = B_1/b_i$，k は

$$k = \left\lfloor \sum_{i=1}^{n} \frac{\xi_i}{b_i} \right\rfloor \qquad (3)$$

($\xi_i = x[b_i]B_{1,i}^{-1}[b_i] \bmod b_i$) で定まる n 未満の整数である．

RNS の利点は，加減乗算が各 RNS 基底の下での剰余加減乗算で実現できることである．各 RNS 基底での演算は独立であるため，RNS は n 個の剰余演算器での並列実装に適した表現である．ただし 2 数の大小比較と除算の効率的アルゴリズムは知られていない．そのため実効的な大小比較および除算を必要としないモンゴメリ乗算と組み合わせて剰余乗算を実現する方法が提案されている[18]．

図 2 で示す RNS モンゴメリ乗算アルゴリズムは，サイズ n の 2 つの RNS 基底 $\mathcal{B}_1$，$\mathcal{B}_2 = \{b_{n+1}, \ldots, b_{2n}\}$ を用いる．基底 $\mathcal{B}_1$，$\mathcal{B}_2$ は $B_2 = \prod_{i=n+1}^{2n} b_i$ としたとき $\gcd(B_1, B_2) = 1$，$\gcd(B_1, N) = 1$ を満たすものとする．入力 x，y に対し，RNS モンゴメリ乗算は

$$\begin{aligned}
w &= \frac{xy + (xy(-N^{-1}) \bmod B_1)N}{B_1} \qquad (4) \\
&\equiv xyB_1^{-1} \pmod N
\end{aligned}$$

を計算する．この式を

$$w = MM(x, y, N, \mathcal{B}_1, \mathcal{B}_2)$$

と書くことにする．上の式から分かるように，モンゴメリ定数は B_1 であり，入力 x のモンゴメリ系での表現は $xB_1 \bmod N$ となる．

図 2 のステップ 3 と 7 の操作を基底拡張と呼ぶ．基底拡張 $\langle x\rangle_{\mathcal{B}_1} \Rightarrow \langle x\rangle_{\mathcal{B}_1\cup\mathcal{B}_2}$ は CRT に基づいて行われる．すなわち，まず CRT により x を求め，各 $b_i (i = n+1, \ldots, 2n)$ に対して $x \bmod b_i$ を求めるという方法である．CRT に用いるアルゴリズムにより，基底拡張の方法がいくつか知られている．たとえば文献 4) は Garner の，文献 5)，6)，10) は Gauss のアルゴリズムに基づく手法である．

RNS モンゴメリ乗算を用いてベキ乗剰余算を行うには，最初に RNS 表現への変換およびモンゴメリ系への変換が，最後にそれらの逆変換が必要である．アルゴリズムの詳細は文献 10) を参照されたい．

基数表現と RNS 表現への変換のコストを削減するために，Bajard らは入出力，したがって通信時にも RNS 表現のままで行う手法を提案している[3]．以下，この手法を full RNS implementation と呼ぶことにする．この場合，通信二者間で RNS 基底を共有しておく必要がある．そのため RNS 基底は共通パラメータとして公開されることもありうる．

4. Leak Resistant Arithmetic

LRA は RNS モンゴメリ乗算にサイドチャネル解析への耐性を持たせる目的で提案された手法である[4]. LRA の基本原理は RNS 基底をあらかじめ固定した $2n$ 個の基底要素からなる集合 $\{b_1, b_2, \ldots, b_{2n}\}$ からランダムに選択することであり，選択時期に応じて 2 種類の方法が提案されている．1 つ目はベキ乗剰余算の最初にランダム選択を行う手法で，Initial Random Bases（以下 IRB と略記）と呼ばれ，2 つ目はベキ乗剰余算の途中でランダムに RNS 基底を変更する手法で，Random Bases During Exponentiation（以下 RBDE と略記）と呼ばれている．

4.1 Initial Random Bases

RNS 基底のランダム選択は，置換 γ をランダムに選択することで行われる．すなわち 1 つ目の RNS 基底は $\mathcal{B}_{1,\gamma} = \{b_{\gamma(1)}, \cdots, b_{\gamma(n)}\}$ であり，2 つ目は $\mathcal{B}_{2,\gamma} = \{b_{\gamma(n+1)}, \cdots, b_{\gamma(2n)}\}$ となる．このとき入力データ c のモンゴメリ系での表現は，$B_{1,\gamma} = \prod_{i=1}^{n} b_{\gamma(i)}$ としたときに $cB_{1,\gamma} \bmod N$ となる．この計算は固定値 $B = \prod_{i=1}^{2n} b_i$ を事前計算しておくことで

$$MM(c, B \bmod N, N, \mathcal{B}_2, \mathcal{B}_1)$$

で実現できる．

IRB により RNS 基底およびモンゴメリ定数 $B_{1,\gamma}$ は ${}_{2n}C_n$ 種類の値をとりうる．このため，Bajard らはモンゴメリ定数 $B_{1,\gamma}$ が乱数 σ と同じ働きをし，メッセージブラインディングの効果があるとしている．たとえば N が 1,024 ビットの場合，Bajard らのパラメータ設定である $r = 32$，$n = 34$ とすると ${}_{2n}C_n \approx 2.8 \times 10^{19} \approx 2^{64}$ であるので，64 ビット分の自由度があるメッセージブラインディングのランダム化と同じレベルとなると主張している．

4.2 Random Bases During Exponentiation

RBDE はベキ乗剰余算の途中で RNS 基底を変更する操作を指す．具体的なアルゴリズムを**図 3** に示す．図 3 は入力データ c，ベキ乗剰余算の途中結果 M が置換 γ で与えられる RNS 基底で表現されているときに，置換 γ' で与えられる RNS 基底での表現に変換するアルゴリズムである．

Function: $(M', C', \gamma') = RBDE(M, C, c, \gamma)$

Input: $M = c^{\alpha} B_{1,\gamma} \bmod N$,
$C = cB_{1,\gamma} \bmod N, c, \gamma$

Output: $M' = c^{\alpha} B_{1,\gamma'} \bmod N$,
$C' = cB_{1,\gamma'} \bmod N, \gamma'$

Precomputation: $B \bmod N$

1: If new γ' is randomly selected {
2: $M'' = MM(M, B \bmod N, N, \mathcal{B}_{2,\gamma'}, \mathcal{B}_{1,\gamma'})$
3: $M' = MM(M'', 1, N, \mathcal{B}_{1,\gamma}, \mathcal{B}_{2,\gamma})$
4: $C' = MM(c, B \bmod N, N, \mathcal{B}_{2,\gamma'}, \mathcal{B}_{1,\gamma'})$
5: } else {
6: $M' = M, C' = C, \gamma' = \gamma$
7: }

図 3 RBDE アルゴリズム
Fig. 3 RBDE algorithm.

Input: $c, N, d = \sum_{i=0}^{\ell-1} d_i 2^i$ $(d_i \in \{0, 1\})$
Output: $m = c^d \bmod N$
Precomputation: $B \bmod N$

1: Radix-RNS conversion: N, c
2: Select random permutation γ_ℓ
3: $c_\ell = MM(c, B \bmod N, N, \mathcal{B}_{2,\gamma_\ell}, \mathcal{B}_{1,\gamma_\ell})$
4: $m_\ell = MM(1, B \bmod N, N, \mathcal{B}_{2,\gamma_\ell}, \mathcal{B}_{1,\gamma_\ell})$
5: for $(i = \ell - 1$ down to $0)$ {
6: $(m_i, c_i, \gamma_i) = RBDE(m_{i+1}, c_{i+1}, c, \gamma_{i+1})$
7: $m_i = MM(m_i, m_i, N, \mathcal{B}_{1,\gamma_i}, \mathcal{B}_{2,\gamma_i})$
8: if $(d_i = 1)$ {
9: $(m_i, c_i, \gamma_i) = RBDE(m_i, c_i, c, \gamma_i)$
10: $m_i = MM(m_i, c_i, N, \mathcal{B}_{1,\gamma_i}, \mathcal{B}_{2,\gamma_i})$
11: }
12: }
13: $m = MM(m_0, 1, N, \mathcal{B}_{1,\gamma_0}, \mathcal{B}_{2,\gamma_0})$
14: RNS-Radix conversion: $\langle m \rangle_{\mathcal{B}_{1,\gamma_0} \cup \mathcal{B}_{2,\gamma_0}}$

図 4 LRA を用いたベキ乗剰余算アルゴリズム
Fig. 4 Modular exponentiation algorithm using LRA.

5. LRA に対する電力解析

LRA は演算レベルでの対策のため，上位レベルでその他の対策を組み合わせることが可能である．Bajard らはその例として，Montgomery ladder によるベキ乗剰余算に LRA を適用した例を示しているが，それが必須であるとの言及はない[4].

本論文では，このような上位レベルの対策は必須であることを示す．具体的には，LRA のみをサイドチャネル解析対策として用いたバイナリ法でのベキ乗剰余算は，RNS 特有の性質を利用した電力解析に対して脆弱であることを示す．

図 4 に，本論文で解析対象とするベキ乗剰余算アルゴリズムを示す．ステップ 2〜4 が IRB に対応する．またこのアルゴリズムには RBDE が 2 回（ステップ 6 と 9）使われているが，これは一方を省略した場合，

RBDE の次の RNS モンゴメリ乗算が，2 乗算なのか乗算なのかが区別できて d が推測可能となるためである．

5.1 提案する電力解析手法

提案する解析手法は，攻撃者がすべての基底要素 $\mathcal{B}_1 \cup \mathcal{B}_2$ を知っているということを仮定する．これは full RNS implementation でありうる状況である．

まず攻撃者は任意に j 個（$0 < j < n$）の基底要素 $\beta_1, \ldots, \beta_j$ を $2n$ 個の基底要素の集合 $\{b_1, b_2, \ldots, b_{2n}\}$ から選ぶ．次に $c = (\beta_1 \cdots \beta_j)^{-1} \bmod N$ を計算し，図 4 のアルゴリズムへの入力データとする．ここで IRB で選択された基底 $\mathcal{B}_{1,\gamma_\ell} = \{b_{\gamma_\ell(1)}, \ldots, b_{\gamma_\ell(n)}\}$ がすべての $\beta_1, \ldots, \beta_j$ を含むと仮定する．一般性を失うことなしに $\beta_i = b_{\gamma_\ell(i)}$ としてよい．このとき最初の RNS モンゴメリ乗算（図 4 のステップ 3）により

$$\begin{aligned} c_\ell &= MM(c, B \bmod N, N, \mathcal{B}_{2,\gamma_\ell}, \mathcal{B}_{1,\gamma_\ell}) \\ &= cB_{1,\gamma_\ell} \bmod N \\ &= \frac{B_{1,\gamma_\ell}}{\beta_1 \cdots \beta_j} \bmod N \\ &= b_{\gamma_\ell(j+1)} \cdots b_{\gamma_\ell(n)} \bmod N \end{aligned} \tag{5}$$

が計算される．もし $b_{\gamma_\ell(j+1)} \cdots b_{\gamma_\ell(n)} < N$ でかつ上の RNS モンゴメリ乗算が N 未満の値を出力すると仮定すると，図 4 のステップ 3 の RNS モンゴメリ乗算の出力 c_ℓ は $c_\ell = b_{\gamma_\ell(j+1)} \cdots b_{\gamma_\ell(n)}$ となる．よって c_ℓ の RNS 表現は $(c_\ell[b_1], \ldots, c_\ell[b_j], 0, \ldots, 0)$ となるため，ベキ乗剰余算におけるループ内での乗算（図 4 のステップ 10）時には，消費電力が小さくなる．一方 2 乗算（図 4 のステップ 7）での入力値 m_i は，ランダムな数と見なせ，一般に c_ℓ のように 0 の成分が多くなるような性質を持っていないと考えられる．よってベキ乗剰余算のループ内における乗算時のほうが 2 乗算時より消費電力が小さくなるため，2 章で述べたように，ベキ指数のビット値 d_i を推定できる．

ベキ乗剰余算の途中で RBDE により RNS 基底が変更された場合も入力データ c に対する変換式（図 3 のステップ 4）は式 (5) と同一のため，上と同じ議論が RBDE についても成り立つ．

5.2 成功確率の評価

上の議論をまとめると，ビット値 d_i を推定するには，次の 3 つの条件が成り立つ必要がある．

（C1） $b_{\gamma_i(j+1)} \cdots b_{\gamma_i(n)} < N$.

（C2） 図 3 のステップ 4 または図 4 のステップ 3 での RNS モンゴメリ乗算の出力 c_i は N 未満．

（C3） 基底 $\mathcal{B}_{1,\gamma_i}$ が $\beta_1, \cdots, \beta_j$ をすべて含む．

まず条件（C2）について考察する．式 (4) のモンゴメリ乗算において，入力値 x，y のうち y が固定されて x が動くとき，その出力が N 以上となる確率は文献 21) より

$$\frac{y}{2B_1} \tag{6}$$

である．式 (6) は基数表現でのモンゴメリ乗算に対して与えられた式であるが，LRA では基底拡張に Garner のアルゴリズムを採用しているため，RNS モンゴメリ乗算は正確に式 (4) を計算する．よって式 (6) は同様に成立する．したがって条件（C2）が成り立つ，すなわち $MM(c, B \bmod N, N, \mathcal{B}_{2,\gamma_i}, \mathcal{B}_{1,\gamma_i})$ の出力が N 未満になる確率は，次の式で与えられる：

$$1 - \frac{B \bmod N}{2B_{1,\gamma_i}}.$$

LRA では計算の中間値の上限を RNS で表現するため，$4N < B_{1,\gamma_i}$ という条件をおいている[4]．したがって式 (6) は下から

$$1 - \frac{B \bmod N}{2B_{1,\gamma_i}} > 1 - \frac{1}{2 \cdot 4} = \frac{7}{8}$$

と評価できる．

しかし実用上は，$r = 32$ であり N のビット数も 32 の倍数であることが多いため，$(B \bmod N)/B_{1,\gamma_i}$ はずっと小さくなる．実際 Bajard らが言及しているパラメータである $\log_2 N = 1024$，$n = 34$，$r = 32$（以下，このパラメータを典型値と呼ぶ）を式 (6) に代入すると

$$\begin{aligned} 1 - \frac{B \bmod N}{2B_{1,\gamma_i}} &> 1 - \frac{2^{1024}}{2(2^{31})^{34}} \\ &= 1 - 2^{-31} \\ &\approx 1 \end{aligned}$$

となる．したがって実用上は高確率で条件（C2）が成り立つとしてよい．

次に条件（C3）を考える．RNS 基底 $\mathcal{B}_{1,\gamma}$ が与えられた j 個の要素を含む確率は

$$\begin{aligned} \frac{{}_{2n-j}C_{n-j}}{{}_{2n}C_n} &= \frac{(2n-j)!n!}{(2n)!(n-j)!} \\ &= \frac{n(n-1)\cdots(n-j+1)}{(2n)(2n-1)\cdots(2n-j+1)} \end{aligned} \tag{7}$$

で与えられる．一般に j が大きくなると式 (7) の値は小さくなる．そのため，条件（C1）を満たす範囲で可能な限り小さな j を選ぶと，条件（C3）が成り立つ確率が大きくなる．たとえば $j = 2$ の場合は

$$\frac{{}_{2n-j}C_{n-j}}{{}_{2n}C_n} = \frac{n-1}{2(2n-1)} \approx \frac{1}{4}$$

であり，$j = 3$ の場合は

$$\frac{{}_{2n-j}C_{n-j}}{{}_{2n}C_n} = \frac{n-2}{4(2n-1)} \approx \frac{1}{8}$$

となる．

実際の場合を考察するため，再び Bajard らの典型値を考える．各 b_i が 32 ビットであることと N が 1,024 ビットであることより，N が 1,024 ビットの中でも大きい部類の数のとき $j=2$，小さい部類の数のとき $j=3$ ととれることが分かる．したがって条件（C3）が成り立つ確率は $j=2$ のとき

$$\frac{n-1}{2(2n-1)} = \frac{33}{134} \approx \frac{1}{4},$$

$j=3$ のとき

$$\frac{n-2}{4(2n-1)} = \frac{32}{268} \approx \frac{1}{8}$$

となる．よって秘密指数 d の各ビット d_i において，図 4 のステップ 10 の RNS モンゴメリ乗算時に条件（C3）が成立する確率，すなわち d_i を推測できる確率は，$j=2$ のときは約 1/4，$j=3$ のときは約 1/8 である．逆にいうと，約 4 回，または 8 回のベキ乗剰余算の消費電力を観測することで，d_i を推測することができる．また置換 γ はランダムに選ばれているため，各ビット d_i において条件（C3）が成立するか否かは他のビットとは独立の事象としてよい．したがって約 4 回，または 8 回のベキ乗剰余算の消費電力を観測することで，秘密指数 d 全体を推測することができる．

以上より，提案した解析手法は数回のベキ乗剰余算の消費電力を観測することで，高確率で秘密指数 d を推測することができるといえる．

6. 考　　察

本章では前章で述べた解析手法に対していくつかの考察を行う．

6.1 仮定に関する考察

提案解析手法は，攻撃者が RNS 基底をすべて知っていることを仮定しており，そのことを利用した解析手法となっている．ただし実用上は，5.2 節で述べたように，2，3 個の RNS 基底を知っていればよい．逆にいえば，提案解析手法を回避するためには，RNS 基底を秘密にしておく必要があるということである．したがって full RNS implementation をはじめ，RNS 基底の公開を前提としている使用方法は，サイドチャネル解析の観点からは利用してはならないといえる．

6.2 Ciet らの RNS 基底選択手法に関する考察

次に，あらかじめ準備する RNS 基底を Ciet らの手法[5),6)] のように選んだ場合を考察する．Ciet らは，512 ビットの法に対し，$58 \le r \le 64$ の範囲の generalized Mersenne numbers $2^{\kappa_1} - 2^{\kappa_2} - 1$ から 63 個の値を用意し，RNS モンゴメリ乗算の際はそこから $n=9$ 個をランダムに選ぶ手法をとっている．この場合，条件（C3）が成り立つ確率（式 (7)）は

$$\frac{{}_{63-j}C_{9-j}}{{}_{63}C_9} = \frac{9\cdot 8\cdots(10-j)}{63\cdot 62\cdots(64-j)}$$

となる．この式の値は $j=1$ のとき 1/7，$j=2$ のとき約 1/54，$j=3$ のとき約 1/473 となる．したがって，LRA の場合より提案解析手法が成功する確率を低くすることができる．

6.3 他の基底拡張アルゴリズムに関する考察

LRA は Garner のアルゴリズムを利用した基底拡張アルゴリズムを用いている．このアルゴリズムは基底拡張時に誤差が発生しないため，5.2 節で述べたように，RNS モンゴメリ乗算で N 未満の値を出力する確率は式 (6) で与えられる．一方，Kawamura らの方法[10)] や Bajard らの方法[2)] では基底拡張時に誤差を許すため，その確率は式 (6) では評価できず，提案解析手法の成功確率が変わるものと考えられる．そこで本節では，Kawamura ら，Bajard らの基底拡張アルゴリズムを考察する．どちらも Gauss のアルゴリズム（式 (2)）に基づいている．

まず Kawamura らの方法を考察する．この方法では式 (2) における k を近似計算で求めているため，誤差が生じて $k-1$ が計算される場合がある．誤差が生じた場合は RNS モンゴメリ乗算の出力は $+N$ されるため N 以上となる．したがって誤差の発生確率の分だけ 5.2 節であげた 2 番目の条件が成立する確率が小さくなることが見込まれる．文献 12) での PC 上でのシミュレーションによると，N が 1,024 ビットのときに基底拡張で誤差が生じる確率は約 0.125 である．したがって RNS モンゴメリ乗算が N 未満の値を出力する確率は約 0.875 倍となるため，LRA の場合と比較して必要な消費電力波形の個数は約 8/7 倍になると考えられる．すなわち Kawamura らの基底拡張アルゴリズムを採用することで提案解析手法に対する安全性を改善できる．ただしその改善度が十分であるとはいえない．

次に Bajard らの方法について考察する．この方法では式 (2) における減算 $-kB_1$ を行わず，$\bmod B_1$ で合同な数のままで計算を行う．そのため RNS モンゴメリ乗算の出力の上限が $(n+2)N$ と大きくなるので，出力が N 未満となる確率はさらに小さくなるものと期待される．

この確率を中心極限定理を利用して見積もる．Bajard らの方法では，RNS モンゴメリ乗算は式 (4) の代わりに

$$w = \frac{xy + (xy(-N)^{-1} \bmod B_{1,\gamma} + kB_{1,\gamma})N}{B_{1,\gamma}}$$

で計算される．この式から分かるように，$w < N$ となるのは $k = 0$ のときに限る．ただし逆は真ではないことに注意しておく．式 (3) より k の値は ξ_i/b_i の和であるが，ここで ξ_i/b_i を区間 $[0, 1)$ で一様分布する確率変数と見なすと，n が十分大きいときは中心極限定理より $\sum_{i=1}^{n} \xi_i/b_i$ の分布は平均 $n\mu$，分散 $n\sigma^2$ の正規分布 $N(n\mu, n\sigma^2)$ に従う．ここで $\mu = 1/2$, $\sigma^2 = 1/12$ である．よって $k = 0$ の確率 $P(k < 1)$ は

$$\begin{aligned} P(k<1) &= P\left(\frac{k-n\mu}{\sqrt{n}\sigma} < \frac{1-n\mu}{\sqrt{n}\sigma}\right) \\ &= \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx \end{aligned}$$

となる．ここで $z = (1 - n\mu)/\sqrt{n}\sigma = \sqrt{12}(1 - n/2)/\sqrt{n}$ である．したがって N が 1,024 ビットの場合の，Bajard らのパラメータの典型値 $n = 34$ の場合では

$$\int_{-\infty}^{-9.5} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx \approx 0$$

となる．文献 12) でのシミュレーション結果によると，n が 34 程度の大きさであっても k の分布は中心極限定理に従っていると見なせるので，上の $P(k < 1) \approx 0$ は妥当な見積もり値と考えられる．以上より，Bajard らの基底拡張アルゴリズムを用いることで，提案解析手法は防御可能であるといえる．

6.4 メッセージブラインディングとの比較

Bajard らは，N が 1,024 ビットの場合，LRA は 64 ビット分の自由度があるメッセージブラインディングのランダム化と同じ効果があると主張している．そこで LRA による入力データのランダム化の効果について，メッセージブラインディングと比較して考察する．

提案解析手法は選択暗号文攻撃であるため，2 章で述べたようにメッセージブラインディングで防御可能である．また，たとえ (σ, τ) として毎回固定した値を用いたとしても，c をモンゴメリ系の表現にしたときにいくつかの RNS 基底の積にならないようにその値を適切に選びさえすれば，提案解析手法を防ぐことができる．つまり LRA による入力データのランダム化の効果は 64 ビット分の自由度があるメッセージブラインディングと同じレベルの効果を与えるとはいえない．これは LRA によってランダム化された入力データの値は，入力データの空間内で偏った分布，すなわち偏った範囲から σ を選んでいるメッセージブラインディングと解釈できる．したがって LRA はサイドチャネル解析対策としてのメッセージブラインディングの代替としては，有効に機能していないといえる．

6.5 対　　策

提案解析手法は，バイナリ法のループ内での 2 乗算と乗算を区別することを目標にしているため，ベキ乗剰余算のアルゴリズムとして square-and-multiply-always や Montgomery ladder のように，2 乗算と乗算の列が秘密指数 d の値によらないアルゴリズムを採用することが対策となる．

また前述したように，適切に選んだ (σ, τ) によるメッセージブラインディングは有効な対策である．

これらは既存の対策手法であり，LRA はこのような対策を併用することが必須であるといえる．ただし既存の対策手法のうち，ベキ指数に群の位数の乱数倍を加えるというランダム化については，提案解析方法は単純電力解析であり，1 回のベキ乗剰余算で約 4 分の 1 のベキ指数のビットを暴くため，十分な対策となっていないと考えられる．

また前節までに述べたように，あらかじめ用意しておく RNS 基底の数を多くしておくこと，RNS モンゴメリ乗算が N 未満の値をほとんど出力しない基底拡張アルゴリズムを採用することがあげられる．さらに RNS 基底を秘密にすることも重要な対策であり，したがって full RNS implementation のような RNS 基底の公開を前提とする利用方法は避けたほうがよい．

7. むすび

本論文ではサイドチャネル解析対策として LRA が採用された RNS モンゴメリ乗算に対し，電力解析手法を提案した．LRA は RNS 基底をランダムに選択することで入力メッセージをランダム化できるため，電力解析に対して耐性があると考えられてきたが，提案した解析手法はその反例といえる．提案した解析手法の対策としてはメッセージブラインディング等の既存の対策手法があげられるため，LRA はこのような対策との併用が必須であるといえる．

参考文献

1) Akishita, T. and Takagi, T.: Zero-Value Point Attacks on Elliptic Curve Cryptosystem, *ISC 2003*, LNCS, Vol.2851, pp.218–223 (2003).
2) Bajard, J.C., Didier, L.S. and Kornerup, P.: Modular Multiplication and Base Extension in Residue Number Systems, *Proc. 15th IEEE symposium on Computer Arithmetic*, pp.59–65 (2001).

3) Bajard, J.C. and Imbert, L.: A Full RNS Implementation of RSA, *IEEE Trans. Comput.*, Vol.53, No.6, pp.769–774 (2004).
4) Bajard, J.C., Imbert, L., Liardet, P.Y. and Teglia, Y.: Leak Resistant Arithmetic, *CHES 2004*, LNCS, Vol.3156, pp.62–75 (2004).
5) Ciet, M., Neve, M., Peerers, E. and Quisquater, J.J.: Parallel FPGA Implementation of RSA with Residue Number Systems—Can Side-Channel Threats be Avoided?, *46th IEEE International Midwest Symposium on Circuits and Systems* (2003).
6) Ciet, M., Neve, M., Peerers, E. and Quisquater, J.J.: Parallel FPGA Implementation of RSA with Residue Number Systems—Can Side-Channel Threats be Avoided?—EXTENDED VERSION, *Cryptology ePrint Archive* (2004). http://eprint.iacr.org/2004/187.pdf
7) Dhem, J.F., Koeune, F., Leroux, P.A., Mestre, P., Quisquater, J.J. and Willems, J.L.: A Practical Implementation of the Timing Attack, *CARDIS '98*, pp.167–182 (1998).
8) Goubin, L.: Refined Power-Analysis Attack on Elliptic Curve Cryptosystems, *PKC 2003*, LNCS, Vol.2567, pp.199–210 (2003).
9) ISO/IEC15408: 1999 Information Technology —Security Techniques—Evaluation Criteria for IT Security (1999).
10) Kawamura, S., Koike, M., Sano, F. and Shimbo, A.: Cox-Rower Architecture for Fast Parallel Montgomery Multiplication, *EUROCRYPT 2000*, LNCS, Vol.1807, pp.523–538 (2000).
11) Kocher, P.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems, *CRYPTO '96*, LNCS, Vol.1109, pp.104–113 (1996).
12) 小池正修，松本 勉：RNS 表現に基づくべき乗剰余算に対するサイドチャネル解析，*ISEC2004–7*, pp.43–50, 電子情報通信学会 (2004).
13) Menezes, A.J., Oorschot, P.C.V. and Vanstone, S.A.: *Handbook of Applied Cryptography*, CRC Press (1997).
14) Messerges, T.S., Dabbish, E.A. and Sloan, R.H.: Power Analysis Attacks of Modular Exponentiation in Smartcards, *CHES '99*, LNCS, Vol.1717, pp.144–157 (1999).
15) Montgomery, P.L.: Modular Multiplication without Trial Division, *Math. Computation*, Vol.44, pp.519–521 (1985).
16) Nozaki, H., Motoyama, M., Shimbo, A. and Kawamura, S.: Implementation of RSA Algorithm Based on RNS Montgomery Multiplication, *CHES 2001*, LNCS, Vol.2162, pp.364–376 (2001).
17) Okeya, K. and Takagi, T.: Security Analysis of CRT-Based Cryptosystems, *ACNS 2004*, LNCS, Vol.3089, pp.383–397 (2004).
18) Posch, K.C. and Posch, R.: Modulo Reduction in Residue Number Systems, *IEEE Trans. Parallel and Distributed Systems*, Vol.6, No.5, pp.449–454 (1995).
19) Rivest, R.L., Shamir, A. and Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Comm. ACM*, Vol.21, pp.120–126 (1978).
20) RSA Laboratories: PKCS #1 v2.1: RSA Cryptography Standard (2002).
21) Walter, C.D. and Thompson, S.: Distinguishing Exponent Digits by Observing Modular Subtractions, *CT-RSA 2001*, LNCS, Vol.2020, pp.192–207 (2001).

(平成 16 年 11 月 29 日受付)
(平成 17 年 6 月 9 日採録)

小池 正修（正会員）

1974 年生．1996 年東京大学理学部数学科卒業．1998 年東京大学大学院数理科学研究科修士課程修了．同年株式会社東芝入社．2003 年東芝ソリューション株式会社に異動．入社以来，暗号と情報セキュリティの研究開発に従事．2005 年横浜国立大学大学院環境情報学府博士課程後期修了．博士（工学）．

松本 勉（正会員）

1986 年東京大学大学院博士課程修了．工学博士．同年横浜国立大学工学部専任講師．同助教授，教授を経て，2001 年より同大学大学院環境情報研究院教授．1981 年より暗号や情報セキュリティの研究に従事．「明るい暗号研究会」創設メンバ．現在，情報セキュリティ，暗号アルゴリズム，認証プロトコル，ディジタル証拠性，情報ハイディング，バイオメトリクス，人工物メトリクス，耐タンパー技術等に広く関心を持つ．国際暗号学会 IACR 理事．暗号技術検討会構成員．CRYPTREC 暗号モジュール委員会委員長．INSTAC 耐タンパー性標準化調査研究委員会委員長．電子情報通信学会より「情報セキュリティの基礎理論」への貢献に関して業績賞を受賞．