

サイドチャネル攻撃へのウィンドウ法を用いた防御法に対する 2階電力差分攻撃

桶屋 勝 幸[†] 櫻井 幸一^{††}

Möller によりウィンドウ法を利用したサイドチャネル攻撃への防御法が提案された。本稿では、Möller の防御法は、2階 DPA 攻撃に対して脆弱であることを示す。n 階 DPA 攻撃は、n 個の中間値に対する漏洩データを用いた DPA 攻撃法である。Möller の防御法に対する提案攻撃法は、同じ点の使用を見つけ出し、それをもとに秘密スカラー値の候補を限定する。その状況下で、直接計算法として Baby-Step-Giant-Step 法を用いることにより、完全にスカラー値を特定する。160 ビットのスカラー値に対して、提案 DPA 攻撃法により、その候補数を 2^{45} 程度に制限される。これらの候補からスカラー値を特定することは現実的に可能である。また、Möller の防御法を提案 DPA 攻撃に対しても耐性を有するように拡張した。そして、提案 DPA 攻撃を用いたあとの計算量的困難性、スカラー倍の計算量に関して、オリジナルの手法と拡張された防御法について比較した。

A Second-order Differential Power Analysis Breaks a Window-method Based Countermeasure against Side Channel Attacks

KATSUYUKI OKEYA[†] and KOUICHI SAKURAI^{††}

Möller proposed a countermeasure using window method against side channel attacks. In this paper, we show Möller's countermeasure is vulnerable to a second-order differential power analysis attack. A side channel attack is an attack that takes advantage of information leaked during execution of a cryptographic procedure. An n th-order differential power analysis attack is the side channel attack which uses n different leaked data that correspond to n different intermediate values during the execution. Our proposed attack against Möller's countermeasure finds out the use of same elliptic points, and restricts candidates of the secret scalar value. In these circumstances, the attack completely detects the scalar value using Baby-Step-Giant-Step method as a direct-computational attack. For a 160-bit scalar value, the proposed attack restricts the number of candidates of the scalar to a 45-bit integer, and the direct-computational attack can actually detect the scalar value. Besides, we improve Möller's countermeasure to prevent the proposed attack. We compare the original method and improved countermeasure in terms of the computational intractability and the computational cost of the scalar multiplication.

1. はじめに

サイドチャネル攻撃 (*side channel attacks*) に対する、ウィンドウ法を利用した防御法が Möller により提案された。本稿では、Möller の防御法 (*Möller's countermeasure*) は、サイドチャネル攻撃の 1 つである DPA 攻撃 (*Differential Power Analysis*) に対して脆弱であることを示す。

1.1 サイドチャネル攻撃

Kocher らは、暗号処理を行う際に漏洩するデータから、秘密情報を推定するサイドチャネル攻撃^{(13), (15)}を提案した。Messerges は n 階 DPA 攻撃⁽¹⁸⁾を与えた。n 階 DPA 攻撃は、n 個の中間値に対する漏洩データを用いた DPA 攻撃である。また Messerges は、1 階 DPA 攻撃に対して耐性を有するが 2 階 DPA 攻撃に対して脆弱⁽¹⁸⁾となる例を示した。そして Coron は、楕円曲線暗号にサイドチャネル攻撃⁽³⁾を拡張した。

サイドチャネル攻撃に対する防御法はいくつか提案されている。Coron はスカラー値のランダム化、点の不可視化、ランダム化射影座標といった防御法⁽³⁾を提案している。Hasan は Kobitz 曲線上の防御法⁽⁸⁾を、Okeya らはモンゴメリ型楕円曲線を用いた防御法⁽²³⁾

[†] 株式会社日立製作所システム開発研究所
Systems Development Laboratory, Hitachi, Ltd.

^{††} 九州大学システム情報科学研究院
Graduate School of Information Science and Electrical
Engineering, Kyushu University

を, Liardet らはヤコビ型楕円曲線を用いた防御法¹⁷⁾を, Joye らはヘジアン型楕円曲線を用いた防御法¹⁰⁾を, Joye らはランダム同型曲線を用いた防御法¹¹⁾を, Oswald らはランダム化加算減算鎖法²¹⁾を, Brier ら¹⁾, Fischer ら⁷⁾, Izu ら⁹⁾はワイエルシュトラス型楕円曲線における防御法を, 提案している.

1.2 Möller の防御法

Möller は, 楕円曲線上のスカラー倍計算におけるサイドチャネル攻撃の防御法として, ウィンドウ法を利用した防御法²⁰⁾を提案した. これは通常のウィンドウ法において, ピットの値によらず必ず(ダミー演算でない)加算を行うように加算鎖を工夫した方法である.

1.3 本研究の成果

Möller の防御法は 2 階 DPA 攻撃 (2nd-order DPA attack) に対して脆弱であることを示す. 提案攻撃法は次の 3 フェーズに分かれる.

- I: スカラー倍計算における 2 つの瞬間での計算に対する電力消費量の差の分散の値としてとりうる 2 つの値を決定する.
- II: 2 つの特定の瞬間での計算に対する電力消費量の差の分散を, スカラー倍計算の実行中に電力消費量を計測することにより特定し, 秘密スカラー値の候補を制限する.
- III: 計算量的攻撃を行い, スカラー値 d を特定する.

フェーズ I では, 楕円曲線上のある 2 点がレジスタに読み込まれる際の電力消費量の差に対する分散を特定する. フェーズ II では, その分散を用いて, 楕円演算として同じ点が使われる瞬間を見つけ出し, スカラー値の候補を限定する. フェーズ III では, 計算量的攻撃として Baby-Step-Giant-Step を用いてスカラー値を完全につきとめる. しかしながら, Baby-Step-Giant-Step 法を直接適用することはできない. これは, Giant-Step の演算が Baby-Step の演算に依存するからである. 提案 Baby-Step-Giant-Step 法では, Giant-Step の演算が Baby-Step の演算と独立となるように改良されている.

この攻撃のフェーズ I, II により制限されたスカラー値の候補数について見積もる. たとえば, 160 ビットのスカラー値に対して, ウィンドウ幅 w を $w = 4$ とすることにより, スカラー値の候補数は 2^{45} 程度になる. これは計算量的攻撃により現実的に求解可能である. したがって, Möller の防御法は提案攻撃法に対して脆弱である.

提案攻撃法に対する防御法として, 大ウィンドウ幅防御法を提案する. これはウィンドウ幅を大きくとる防御法であり, そのことによりスカラー値の候補数が増加するのでフェーズ III が難しくなる. しかしながら, この防御法は非常に遅いため, 提案攻撃法に対する高速防御法は, 議論すべき課題として残っている.

本稿の構成は以下のようになっている. 2 章はサイドチャネル攻撃のサーベイである. 3 章で Möller の防御法について述べる. 4 章で Möller の防御法に対する提案 DPA 攻撃について説明する. 5 章で提案 DPA 攻撃に対する防御法を提案する.

2. サイドチャネル攻撃

実際の環境下での暗号装置においては, 暗号処理を実行する際に, 入力データおよび出力データ以外にも漏洩するデータが存在する. たとえば, 暗号処理にかかる計算時間であったり, スマートカードであれば電力は外部より供給されるので, 電力消費量もその類のデータである. Kocher らは, それらの漏洩データより, 暗号装置内部に格納されている秘密情報を推定する攻撃法, いわゆるサイドチャネル攻撃^{13)~15)}を開発した. サイドチャネル攻撃には, タイミング攻撃¹³⁾や SPA (Simple Power Analysis) 攻撃¹³⁾, DPA (Differential Power Analysis) 攻撃^{14),15)}がある. スマートカードは, サイドチャネル攻撃の影響を特に受けやすい暗号装置である.

Kocher らが提案した攻撃は, その攻撃対象は主として DES⁴⁾や RSA²⁶⁾であり, 楕円曲線暗号^{12),19)}に対する攻撃は知られていなかった. Coron は, DPA 攻撃を楕円曲線暗号³⁾へと拡張した.

秘密情報に依らず固定された計算手順をとることにより SPA 攻撃を防ぐ防御法としては, ダミー演算を用いた Coron の耐 SPA スカラー倍計算方法³⁾, モンゴメリ型楕円曲線を用いた防御法^{22),23)}, ワイエルシュトラス型楕円曲線にモンゴメリ法を拡張した方法^{1),7),9)}などがある. SPA 攻撃に対する防御法をランダム化を用いて DPA 攻撃に対する防御法へと拡張する方法としては, ランダム化射影座標^{3),22),23)}やランダム同型曲線^{9),11)}などがある.

Kocher らは高階 DPA 攻撃 (a higher-order DPA attack)⁴⁾を提案した. 高階 DPA 攻撃は, 複数の中間値に対応する電力消費量を用いる DPA 攻撃である. より正確には, Messerges は n 階 DPA 攻撃の定義¹⁸⁾を与えた.

定義 1 n 階 DPA 攻撃は, アルゴリズム実行中に,

最近の論文²⁴⁾は, その防御法の基本バージョンは SPA 攻撃に対して脆弱であることを示した.

n 個の異なる中間値に対応する n 個の異なる電力消費信号データを利用する。

さらに, Messerges は共通鍵暗号における, 1 階 DPA 攻撃に対するランダムマスクを用いた防御法は, 2 階 DPA 攻撃に対して脆弱¹⁸⁾であることを示した。

3. Möller の防御法

Möller は, 楕円曲線上のスカラー倍計算におけるサイドチャンネル攻撃の防御法の 1 つとしてウィンドウ法を利用した防御法²⁰⁾を提案した。

d を整数とし, $w \geq 2$ をウィンドウ幅とする。 d を次のように表す。 $d = \sum_{i=0}^{k'} b'_i \cdot 2^{wi}$ 。ただし, $b'_i \in \{0, 1, \dots, 2^w - 1\}$ とし, k' は最小にとる, すなわち $b'_{k'} \neq 0$ である。次に, $d = \sum_{i=0}^k b_i \cdot 2^{wi}$ を満たす $b_i \in J : J = \{-2^w, 1, 2, \dots, 2^w - 1\}$ および k を次のようにして定める。 $c_0 = 0$ とし, $i = 0, \dots, k' + 1$ に対して, $t_i = b'_i + c_i$ および

$$(c_{i+1}, b_i) = \begin{cases} (1, -2^w) & \text{if } t_i = 0 \\ (0, t_i) & \text{if } 0 < t_i < 2^w \\ (2, -2^w) & \text{if } t_i = 2^w \\ (1, 1) & \text{if } t_i = 2^w + 1 \end{cases}$$

により, b_i を定める。 $b_{k'+1} \neq -2^w$ であれば $k = k' + 1$ とし, そうでなければ $k = k'$ とする。

前計算テーブルにおいては, 楕円曲線上の点 jP に対し, その値を格納するメモリを P_j で表す。また, $P_j + P_{j'}, 2P_j$ により, メモリに格納されている値を用いて計算される楕円曲線の加算, 2 倍算の値を表す。アルゴリズム (前計算ステージ)

入力 スカラー値 d , 点 $P = (x, y)$, ウィンドウ幅 w

出力 前計算テーブル $\{P_j\}_{j \in J}$

- (1) 乱数 λ を生成する。
- (2) $P_1 \leftarrow (\lambda x, \lambda y, \lambda)$
- (3) For $j = 2$ to $2^w - 2$ step 2 do
 - (a) $P_j \leftarrow 2P_{j/2}$
 - (b) $P_{j+1} \leftarrow P_j + P_1$
- (4) $P_{-2^w} \leftarrow -2P_{2^w-1}$

アルゴリズム (実計算ステージ)

入力 スカラー値 $d = \sum_{i=0}^k b_i 2^{wi}$, 点 P , ウィンドウ幅 w , 前計算テーブル $\{P_j\}_{j \in J}$

出力 スカラー倍 dP

- (1) $Q \leftarrow \mathcal{O}$, where \mathcal{O} is the point at infinity.

- (2) For $i = k$ down to 0 do

- (a) $Q \leftarrow 2^w Q$

- (b) $Q \leftarrow Q + P_{b_i}$

- (3) Q をスカラー倍 dP として出力する。

この防御法は, スカラー値 d によらず固定された計算手順をとるため, SPA 攻撃に対して耐性を有する。また, 前計算ステージで, 前計算点をランダム化座標によりランダム化し, 実計算ステージでランダム化された点を用いて計算するため, DPA 攻撃に対しても耐性を有すると考えられている。

4. Möller の防御法に対する DPA 攻撃

4.1 電力漏洩モデル

Messerges は, 漏洩情報は実行中のデータのハミングウェイトに依存する, という次の電力消費モデル¹⁸⁾を提案し, スマートカード上でそのモデルが有効であることを確認した。時間 t における電力消費量 $C(t)$ は, $C(t) = \epsilon \cdot H(t) + L + N$ と表される。ここで, $H(t)$, ϵ , L , N はそれぞれ, 時間 t における中間データのハミングウェイト, ハミングウェイトが '1' 増加したときの増加電力量, 全電力における定数部分, およびノイズである。 $\mathcal{E}[N]$, $\mathcal{V}[N]$ はそれぞれ, ノイズ N の平均と分散を表す。また, ノイズ N の平均は 0, すなわち $\mathcal{E}[N] = 0$, と仮定されている。

ランダムに選ばれたビット列のハミングウェイトの平均と分散に関して, 次の補題が成立する。

補題 1 ランダムに選ばれたビット長 l のデータのハミングウェイトは, 平均 $\mathcal{E}_l = \frac{l}{2}$, 分散 $\mathcal{V}_l = \frac{l}{4}$ となる。

証明 $\mathcal{E}_l = \frac{l}{2}$ は明らかである。 $\mathcal{V}_l = \frac{l}{4}$ を示す。分散の定義より, $\mathcal{V}_l = \frac{1}{2^l} \sum_{k=0}^l \binom{l}{k} \left(\frac{l}{2} - k\right)^2$ である。

他方, 二項係数の定義より, $(1+x)^l = \sum_{k=0}^l \binom{l}{k} x^k$ である。 $\frac{d}{dx} \Big|_{x=1}, \frac{d^2}{dx^2} \Big|_{x=1}$ をとることにより, $l \cdot 2^{l-1} = \sum_{k=1}^l k \binom{l}{k}$, $l(l-1) \cdot 2^{l-2} = \sum_{k=2}^l k(k-1) \binom{l}{k}$

ハミングウェイトは CPU のレジスタおよび RAM に格納されているデータに対するハミングウェイトを指す。また増加電力量は, CPU のレジスタに対するもの, および RAM に対するものが存在するが, ここではそれらの値は一致すると考えている。厳密に言えば, それらの値は異なる可能性があるが, CPU のレジスタに格納される値と RAM に格納される値は一般に性質の異なるものであるため, ここでの議論に対して大きな影響を与えないと考えられる。前者はアドレスやカウンタ, 後者は実際のデータが主に格納される。

これは射影座標を用いる場合である。Chudnovsky Jacobian 座標を用いる場合は, $P_1 \leftarrow (\lambda^2 x, \lambda^3 y, \lambda, \lambda^2, \lambda^3)$ とする。

となる．これらの式を v_i の式に代入することにより， $v_i = \frac{1}{4}$ を得る． □

次の命題は，Möller の防御法に対する DPA 攻撃を構成するうえで有用である． $C^A[Q, P](t)$, $t \in [0, T^A]$ を，点 Q と点 P の加算における電力消費量とし， $C^D[Q](t)$, $t \in [0, T^D]$ を，点 Q の 2 倍算における電力消費量とする．ただし， T^A は加算の計算時間， T^D は 2 倍算の計算時間である．

命題 1 加算の 2 番目のパラメータ (点 P もしくは点 P') が，時間 \tilde{t} においてレジスタに読み込まれるとする．そのとき，電力消費量の分散に関して次の関係が成立する．

$$\begin{aligned} v_1 &:= \mathcal{V}[C^A[Q, P](\tilde{t}) - C^A[Q', P](\tilde{t})] \\ &= \epsilon \cdot \frac{r - |P|}{2} + 2\mathcal{V}[N], \\ v_2 &:= \mathcal{V}[C^A[Q, P](\tilde{t}) - C^A[Q', P'](\tilde{t})] \\ &= \epsilon \cdot \frac{r}{2} + 2\mathcal{V}[N]. \end{aligned}$$

特に， $v_1 < v_2$ が成り立つ．ここで， $Q, Q', P, P' \in E$ はランダムに選ばれている．また， r はレジスタのビット長を， $|\cdot|$ はデータのビット長を表す．

証明 Messerges の電力漏洩モデル¹⁸⁾により，

$$\begin{aligned} v_1 &= \mathcal{V}[(\epsilon \cdot H[Q, P](\tilde{t}) + L + N(\tilde{t})) \\ &\quad - (\epsilon \cdot H[Q', P](\tilde{t}) + L + N(\tilde{t}))] \end{aligned}$$

となる．ここで， $H[Q, P](t)$, $H[Q', P](t)$ はそれぞれ，点 Q, P の加算における時間 t でのレジスタのハミングウェイト，点 Q', P の加算における時間 t でのレジスタのハミングウェイトを表す．すなわち，時間 t におけるレジスタ内のデータを 2 進表記した場合の 1 の個数である．定数部分はキャンセルし，ノイズ部分は独立である．また，点 P は時間 \tilde{t} にレジスタに読み込まれるので，加算 $Q + P$ における点 P のハミングウェイトと，加算 $Q' + P$ におけるものとは一致し，キャンセルする．そのため，各ハミングウェイトは，ビット長 $r - |P|$ のデータのハミングウェイトと考えてよい．前者のハミングウェイトを出

力する確率変数を X_1 ，後者に対する確率変数を X_2 とする．すなわち， $X_1(\tilde{t}) = (\text{加算 } Q + P \text{ におけるレジスタのハミングウェイト} - \text{点 } P \text{ のハミングウェイト})$ ， $X_2(\tilde{t}) = (\text{加算 } Q' + P \text{ におけるレジスタのハミングウェイト} - \text{点 } P \text{ のハミングウェイト})$ とする．点 P 以外におけるハミングウェイトは独立に変化すると考えられるので， X_1, X_2 が独立であると仮定する．独立な確率変数 X_1, X_2 に対して， $\mathcal{V}[X_1 + X_2] = \mathcal{V}[X_1] + \mathcal{V}[X_2]$ が成り立つ⁶⁾．ただし， $\mathcal{V}[X]$ は確率変数 X の分散を表す．補題 1 より， $\mathcal{V}[X_1] = \mathcal{V}[X_2] = \frac{r - |P|}{4}$ となる． $\mathcal{V}[-X_2] = \frac{r - |P|}{4}$ に注意すると，ハミングウェイトの差の分散 $\mathcal{V}[X_1 - X_2]$ は $\frac{r - |P|}{2}$ となる．同様にノイズに対する分散を考慮すると，ノイズは X_1, X_2 およびノイズどうしは独立であるから， $v_1 = \epsilon \cdot \frac{r - |P|}{2} + 2\mathcal{V}[N]$ を得る．

次に v_2 の値を導出する．点 P, P' のハミングウェイトの不一致を考慮すると，この場合の確率変数は $X_1(\tilde{t}) = (\text{加算 } Q + P \text{ におけるレジスタのハミングウェイト})$ ， $X_2(\tilde{t}) = (\text{加算 } Q' + P' \text{ におけるレジスタのハミングウェイト})$ となる．ビット長 r のデータのハミングウェイトとなる．したがって $v_2 = \epsilon \cdot \frac{r}{2} + 2\mathcal{V}[N]$ を得る． □

注意 1 v_1, v_2 はハードウェア特性や実装に依存すると考えられる．すなわち，メモリからレジスタへの読み込み方法などに依存する．たとえば，データをレジスタに読み込むときにデータ全体を一度に読み込むのか，それともデータをいくつかに分割したうえで読み込むのかにより，変わると考えられる．いずれにせよ，点 P が読み込まれる際に同じ部分が読み込まれるため， $v_1 < v_2$ が成り立つ．

4.2 提案攻撃法

Möller の防御法への提案 DPA 攻撃法は次の 3 段階に分かれる．

- I: 命題 1 の v_1, v_2 を特定する．
- II: $s = 0, 1, \dots, k$ に対して， C'_s を計算する．ただし， $C'_s := \{s' \mid b_s = b_{s'}\}$ である．
- III: 計算量的な攻撃により， C_j と C'_s との関係を見つけ出し，スカラー値 d を特定する．ただし， $C_j := \{s \mid b_s = j\}$ である．

4.3 フェーズ I

v_1, v_2 を特定する方法について，攻撃者が加算装置をシミュレートできる場合と，そうでない場合とにわけて説明する．まず，攻撃者が加算装置をシミュレートできるとする．この場合，シミュレーターの電力消費量を計測することにより， v_1, v_2 を特定できる．

ここでのレジスタは，CPU のレジスタおよび加算演算が使用する (値を変更する) RAM 領域を指す．

すなわち， r は CPU のレジスタおよび加算演算が使用する RAM 領域のビット長の総和を表す．この値は CPU の仕様や実装の仕方にも依存するが，実装者が最適な実装を目指せば，160 ビットの楕円曲線暗号を実装した場合，おおよそ 1,500 ~ 2,000 ビット (200 ~ 300 バイト) 程度の値になる．また $|P|$ は， $P = (X, Y, Z)$ と表されていれば，各 X, Y, Z のビット長の総和となる．

次に、攻撃者が加算装置をシミュレートできないとする。この場合は、スカラー倍計算を用いて特定できる。 t_s を、加算の 2 番目のパラメータ P_{b_s} が $i = s$ のときのステップ 2.b でレジスタに読み込まれる時間とする。

$t_s = (k - s)(T^A + wT^D) + wT^D + \tilde{t}$
 $s \neq s'$ のときの m_0 回の測定における相関関数 $g(t_s, t_{s'})$ を

$$g(t_s, t_{s'}) := \frac{1}{m_0} \sum_{m=1}^{m_0} [C_m(t_s) - C_m(t_{s'})]^2$$

とする。ただし、 $C_m(t)$ は、Möller の防御法の m 回の実行に付随する電力消費量である。命題 1 により、

$$\begin{aligned} g(t_s, t_{s'}) &= \frac{1}{m_0} \sum_{m=1}^{m_0} [C^A[Q, P_j](\tilde{t}) - C^A[Q', P_j](\tilde{t})]^2 \\ &\rightarrow v_1 \quad (\text{as } m_0 \rightarrow \infty) \end{aligned}$$

if $b_s = b_{s'}$ ($b_s P = b_{s'} P = P_j$) , および

$$\begin{aligned} g(t_s, t_{s'}) &= \frac{1}{m_0} \sum_{m=1}^{m_0} [C^A[Q, P_j](\tilde{t}) - C^A[Q', P_{j'}](\tilde{t})]^2 \\ &\rightarrow v_2 \quad (\text{as } m_0 \rightarrow \infty) \end{aligned}$$

if $b_s \neq b_{s'}$ ($b_s P = P_j, b_{s'} = P_{j'}$) を得る。すなわち、次の命題を得る。

命題 2 $b_s = b_{s'}$ if $g(t_s, t_{s'}) \rightarrow v_1$; $b_s \neq b_{s'}$ if $g(t_s, t_{s'}) \rightarrow v_2$.

攻撃者は、任意の s, s' に対して、 $g(t_s, t_{s'})$ を計算する。その値の数が 2 つ ($v_1, v_2, (v_1 > v_2)$) しか存在しないとすると、 $v_1 = v_1', v_2 = v_2'$ が成り立つ。値の数が 1 つ (v') のみの場合は、 $b_s = j$ ($s = 0, 1, \dots, k$) を代入することにより、ある $j \in J$ に対して $Q = dP$ が成立するか否かを調べる。もし、ある j に対して $Q = dP$ であれば、 $v_1 = v'$ であり、さらに $d = \sum_{i=0}^k j 2^{wi}$ が成り立つ。もし、任意の j に対して $Q \neq dP$ であれば、 $v_2 = v'$ である。この場合、任意の $s \neq s'$ に対して $b_s \neq b_{s'}$ が成り立つ。一方で、スカラー倍計算高速化のためのウィンドウ幅に対する通常の選択、すなわち $k > 2^w$, により、 $v_2 = v'$ となる場合は起こらないと考えてよい。

注意 2 時間 t_s と $t_{s'}$ に対する 2 つの異なる電力消費量を用いているため、提案攻撃法は 2 階 DPA 攻撃である。

4.4 フェーズ II

攻撃者は、固定された s に対し、任意の s' に対す

る $g(t_s, t_{s'})$ を計算し、命題 2 を用いて C'_s を見つけ出す。すなわち、 $b_s = b_{s'}$ となるすべての s' を特定する。同様に、すべての s に対する C'_s を見つけ出す。

注意 3 フェーズ I で v_2 だけの場合は、各 s に対する C'_s は、一点集合 $\{s\}$ となる。

4.4.1 フェーズ I, II の効果

フェーズ I, II はスカラー値 d の候補を制限する。この候補数について見積もる。

C'_s と C_j との各対応はスカラー値 d の候補を与えるので、その対応の数について見積もることにより、スカラー値 d の候補数を得ることができる。

命題 3 $S := \{s | s \leq s' \text{ for any } s' \text{ with } b_s = b_{s'}\}$, $q := \#S$ とする。そのとき、スカラー値 d の候補数は $\prod_{m=0}^{q-1} (2^w - m)$ である。

証明 スカラー値 d の候補数は、 2^w 個から q 個を選ぶ順列の数に等しい。□

命題 4 b_i がランダムに選ばれると仮定する。そのとき、 q の期待値は、 $2^w \left(1 - \left(\frac{2^w - 1}{2^w}\right)^{k+1}\right)$ となる。ただし、 k は (ウィンドウの数 - 1) を表す。

証明 選ばれない $j (\in J)$ の数の期待値が $2^w \cdot \left(\frac{2^w - 1}{2^w}\right)^{k+1}$ となることを、ウィンドウの数に関する帰納法により示す。ウィンドウの数が 1 の場合は、ただ 1 つの $j (\in J)$ が選ばれる。すなわち、期待値は $2^w - 1$ である。

次に、 k に対して、選ばれない j の数の期待値が $2^w \cdot \left(\frac{2^w - 1}{2^w}\right)^k$ ($=: x$) であると仮定する。そのとき、新しく選ばれる b_i が既知の j と等しくなる確率は、 $(2^w - x)/(2^w)$ である。そのため、 $k + 1$ に対する選ばれない j の数の期待値は

$$\begin{aligned} x \cdot \frac{x}{2^w} + (x - 1) \frac{2^w - x}{2^w} &= \frac{2^w - 1}{2^w} \cdot x \\ &= 2^w \cdot \left(\frac{2^w - 1}{2^w}\right)^{k+1} \end{aligned}$$

となる。□

表 1 は候補数に関する数値例である。たとえば、160 ビットスカラー値 d に対するスカラー倍を、ウィンドウ幅 $w = 4$ で、Möller の防御法により計算した場合、フェーズ I, II はその候補数をおよそ 2^{45} へと制限する。

4.5 フェーズ III

フェーズ I, II により、スカラー値 d の候補が制限される。フェーズ III は Baby-Step-Giant-Step 法²⁷⁾ を用いてスカラー値 d を完全に特定する。

表1 スカラー値に対する候補数
Table 1 Numbers of candidates for scalar values.

ウィンドウ幅	w	2	3	4	5	6
d =	160	2^5	2^{16}	2^{45}	2^{93}	2^{132}
	192	2^5	2^{16}	2^{45}	2^{100}	2^{148}
	256	2^5	2^{16}	2^{45}	2^{109}	2^{179}
	384	2^5	2^{16}	2^{45}	2^{117}	2^{222}
	512	2^5	2^{16}	2^{45}	2^{118}	2^{252}

各要素は、スカラー値 d に対してフェーズ I, II により制限されたときの候補数を表している。

まず最初に、 S を、次の条件を満たすように2つのクラス S_1, S_2 に分割する。

- (1) $S_1 \cap S_2 = \phi, S_1 \cup S_2 = S$.
- (2) $\#S_1 = \lfloor \#S/2 \rfloor$.

スカラー値 d の候補 \tilde{d} は

$$\tilde{d} = \sum_{s_1 \in S_1} \sum_{s' \in C'_{s_1}} j_{s_1} 2^{ws'} + \sum_{s_2 \in S_2} \sum_{s' \in C'_{s_2}} j_{s_2} 2^{ws'}$$

と表すことができる。このとき、任意の $s_1 \in S_1$ と、任意の $s_2 \in S_2$ に対して、 j_{s_1} と j_{s_2} とは等しくない。そのため、 $\{j_{s_2}\}_{s_2 \in S_2}$ と $\{j_{s_1}\}_{s_1 \in S_1}$ とは、結合可能であるが、 $\{j_{s_2}\}_{s_2 \in S_2}$ と、ある $s_1 \in S_1$ に対して $j'_{s_1} \in \{j_{s_2}\}_{s_2 \in S_2}$ となるような s_1 が存在する $\{j'_{s_1}\}_{s_1 \in S_1}$ とは、結合不可能である。したがって、 S_1 に対する演算を Baby-Step として行い、 S_2 に対する演算を Giant-Step として行うとすると、Giant-Step の演算は、Baby-Step の演算と独立には計算できない。その結果、Baby-Step-Giant-Step 法²⁷⁾を直接使うことはできない。

提案 Baby-Step-Giant-Step 法の Giant-Step では、すべての候補 \tilde{d} に対して $\{j_{s_2}\}_{s_2 \in S_2}$ に対する演算を計算することにより、Baby-Step の演算との独立性を保っている。

攻撃アルゴリズム (Baby-Step-Giant-Step)

入力 楕円曲線上の点 $P, Q = dP$, ウィンドウ幅 w , フェーズ I, II で得た情報 (S, C'_s, S_1, S_2)

出力 スカラー値 d

- (1) (Baby-Step) 任意の $(\dots, j_s, \dots) \in \bigoplus_{s \in S_1} J$ で $j_s \in J - \{j_{\tilde{s}} \mid \tilde{s} \in S_1, \tilde{s} < s\}$ を満たすものに対して、 $Q - \sum_{s \in S_1} \sum_{s' \in C'_s} j_s 2^{ws'} P$ を計算する。
- (2) (Giant-Step) 任意の $(\dots, j_s, \dots) \in \bigoplus_{s \in S_2} J$ で $j_s \in J - \{j_{\tilde{s}} \mid \tilde{s} \in S_2, \tilde{s} < s\}$ を満たすものに対して、 $\sum_{s \in S_2} \sum_{s' \in C'_s} j_s 2^{ws'} P$ を計算する。
- (3) $\left\{ Q - \sum_{s \in S_1} \sum_{s' \in C'_s} j_s 2^{ws'} P \right\}$

$\cap \left\{ \sum_{s \in S_2} \sum_{s' \in C'_s} j_s 2^{ws'} P \right\}$ を計算する。具

体的にいえば、Baby-Step で計算した値を x 座標の大きさに関してソートし、同様に、Giant-Step で計算した値もソートする。そして、一致する値を検索する。また、一致する値はただ1つ存在する。

$$(4) \quad Q - \sum_{s \in S_1} \sum_{s' \in C'_s} j_s^{(1)} 2^{ws'} P = \sum_{s \in S_2} \sum_{s' \in C'_s} j_s^{(2)} 2^{ws'} P \text{ が一致したとする。そのとき, } d = \left(\sum_{s \in S_1} \sum_{s' \in C'_s} j_s^{(1)} 2^{ws'} \right) + \left(\sum_{s \in S_2} \sum_{s' \in C'_s} j_s^{(2)} 2^{ws'} \right) \text{ となる。}$$

ただし、 $\bigoplus_{s \in S_1} J, \bigoplus_{s \in S_2} J$ は集合としての直和を表す。すなわち、 J の元を $\#S_1$ 個 (もしくは $\#S_2$ 個) 並べたもの全体からなる集合である。

注意4 提案 Baby-Step-Giant-Step 法は、計算時間 $O(\sqrt{n})$, 必要メモリ $O(\sqrt{n})$ の確定的アルゴリズムである。ただし、 $n = \left[\prod_{m=0}^{\lfloor (q+1)/2 \rfloor} (2^w - m) \right]^2$ である。離散対数問題を解く確率的アルゴリズムで、メモリ効率が良いカンガルー法^{25), 28)}がこの状況に適用できるか否かについては未解決である。

注意5 d が160ビット整数、ウィンドウ幅が4のとき、上記 n は 2^{58} 程度の大きさになる。これは、現実的に解法可能な楕円離散対数問題⁵⁾の範囲に入る。

5. 実験結果

フェーズ I および II では、電力消費量の分散の値が異なるかどうかを用いている。実際に IC カード上で実装を行った場合、そのような違いが生ずるかどうかについて実験を行った。

次の楕円曲線パラメータを用いて、Möller の防御法によるスカラー倍計算アルゴリズムを IC カード向け16ビットマイコン上で実装した。

$p =$	f54fd8ed 7ad7bac4
$6a94a8e5$	a6dd3cc5 797795f7
$a =$	723b4d97 59c4e2df
$5b49c00d$	54ad4573 a3fa0cc8
$b =$	21871cba 7e2f62ba
$0825529f$	1125d6b3 5a07cbe6
$G_x =$	3358488e 1c5daf87
$58b777e2$	c89f9b8a 1eb0de14
$G_y =$	2d5ccbe4 9ecef652
$5d242da5$	b492eb7c c1cd6c21

ただし, p, a, b, G_x, G_y はそれぞれ, 有限体の位数, 楕円曲線の定義方程式 $y^2 = x^3 + ax + b$ の a, b , ベースポイント G の x 座標, y 座標を表している.

実装したプログラムでは, 楕円加算を計算する際に次のような処理を行っている. 被加算点は計算ワーク領域上に存在している. 加算点を RAM から計算ワーク領域 (RAM 上に存在) に転送する. 楕円加算を実際に計算する. 計算結果は計算ワーク領域上にそのまま残す.

IC チップにプログラムを格納し, その IC チップをホワイトカードに貼り付けて IC カードとした. その IC カードをリーダー/ライタボード上で読み取らせ, その際の電力消費量 (電圧) を計測した. 計測に際して, 電源電圧を 5 V, CPU 動作周波数を 3.57 MHz に設定している. RAM に格納されている前計算された点を計算ワーク領域に転送した直後の電圧に注目し, 提案攻撃法の解析を行った. 基準となる電圧を 2 つ定め, その間を 4,096 等分し, 計測した電圧を 12 ビットの整数に変換した.

スカラー倍計算における各加算に対して, 1,000 回の計測を行った. その結果が表 2 である. スカラー倍計算における 1 回目の加算と, 2 回目, 3 回目, 4 回目の加算に対する電力消費量の差分値の分散を計算したものが表 3 である. この結果から, 1 回目と 3 回目では同じ前計算点を用いている, 1 回目と 2 回目, 1 回目と 4 回目では異なる前計算点を用いていると考えられる. また実際にそうであった. したがって, 実環境において, 攻撃者がフェーズ I および II の処理を行うことが可能といえる.

実装したプログラムでは, 使用する RAM 領域は約 500 バイトであった. このうち約 160 バイトが計算ワーク領域として割り当てられている. プログラムが使用しない RAM に格納されている値は, プログラム実行中には値が変わらないため, 全電力の定数部分として働いていると考えられる. また, 使用する RAM 領域においても, 前計算テーブルなどのスカラー倍実行中は値が変化しない領域については, 全電力の定数部分と見なすことができる. したがって, 命題 1 のレジスタに相当するものは, 加算演算により値が変更される領域と CPU のレジスタとなる. 加算演算により値が変更される領域は, 計算ワーク領域である約 160 バイトである. 実験に用いた IC チップでは, CPU のレジスタとして約 30 バイトの領域がある. 加算点を計算ワーク領域への転送する際には約 60 バイトのデータが転送される. 命題 1 の v_1, v_2 でノイズ部分の影響を無視すると, $v_2/v_1 = (160+30)/(160+30-60) = 1.46$

表 2 スカラー倍の各加算実行時の電力消費量

Table 2 The power consumption while the elliptic addition was performed on the scalar multiplication.

1 回目の加算	2008	2047	2018	...
2 回目の加算	2066	1999	2084	...
3 回目の加算	2031	2019	2058	...
4 回目の加算	1998	2044	2019	...
...

表 3 電力消費量の差分値の分散

Table 3 The variants of the differences of the power consumption.

1 回目-2 回目	2165.863525
1 回目-3 回目	1245.660645
1 回目-4 回目	2481.250488
...	...

となる. 実験結果では, $2166/1246 = 1.74$ となり, 近い値といえる.

6. 防 御 法

上述した DPA 攻撃法に対する防御法として, 次の防御法が考えられる.

大ウィンドウ幅: ウィンドウ幅を大きくする防御法である. 表 1 より, ある程度ウィンドウ幅を大きくすると, スカラー値の候補数が増加することが分かる. そのため, フェーズ III の実行を困難にすることができる.

大ウィンドウ幅防御法は, ウィンドウ幅が 7, スカラー値 d が 170 ビット整数の場合, スカラー値の候補数がおおよそ 2^{158} となる. これは, ほぼ 160 ビットのセキュリティを提供するといつてよい.

しかしながら, ウィンドウ幅を大きくしたことにより, 計算量が増加する. 加算では $\mathcal{J} + \mathcal{J}^c \rightarrow \mathcal{J}^m$, 加算直前の 2 倍算では $\mathcal{J}^m \rightarrow \mathcal{J}$, 2 倍算直前の 2 倍算では $\mathcal{J}^m \rightarrow \mathcal{J}^m$ と座標系を選び, 前計算点を \mathcal{J}^c で格納することにより, 実計算ステージにおいて高速に計算できる. ただし, $\mathcal{J}, \mathcal{J}^c, \mathcal{J}^m$ はそれぞれ Jacobian 座標, Chudnovsky Jacobian 座標, modi-

CPU のレジスタのうち値が変わらないものも存在するので, 実際にはもう少し値が大きくなると考えられる.

本稿での計算量の見積りは素体上の楕円曲線の場合である. 他の場合も同様に見積もることが可能である.

文献 20) で与えられている計算量は, 座標系の選び方を考慮していないため, 実際にはその計算量では計算できない. たとえば, 加算の計算量は乗算 15 回とされているが, これを満たす座標系のとり方は $\mathcal{J}^c + \mathcal{J}^c \rightarrow \mathcal{J}^m$ であり, この加算を行うには, その直前の 2 倍算で $\mathcal{J}^m \rightarrow \mathcal{J}^c$ と座標系を選ぶ必要がある. しかしながら, この計算量は乗算 9 回であり, 文献 20) であげている乗算 8 回とは異なる.

表 4 Möller の防御法に対する計算量的困難性とスカラー倍計算量

Table 4 Computational intractability and computational cost for Möller's countermeasure.

計算方法	ウィンドウ幅	ビット数	スカラー値の候補数	スカラー倍計算量
オリジナル	4	160	2^{45}	$1884.2M$
大ウィンドウ幅防御法	7	160	2^{152}	$2975.6M$
大ウィンドウ幅防御法	7	170	2^{158}	$3106.4M_{170}$

オリジナルは, Möller のオリジナルの防御法²⁰⁾を指す. M は 160 ビット有限体における乗算, M_{170} は 170 ビット有限体における乗算を指す.

fied Jacobian 座標を表す. この場合の実計算ステージの計算量は $(4w+11)kM + (4w+5)kS$ である. ただし, M, S はそれぞれ有限体上の乗算, 2 乗算の計算量である. 前計算ステージでは前計算点を \mathcal{J}^c で格納するので, $(16 \cdot 2^{w-1} - 11)M + (9 \cdot 2^{w-1} - 3)S$ の計算量が必要である.

$S = 0.8M$ を仮定すると¹⁶⁾, 160 ビットでの計算量は, $w = 4$ のときは $1884.2M$ であり, $w = 7$ のときは $2975.6M$ である. そのため, 大ウィンドウ幅防御法では $1091.4M$ だけ計算量が増加している. さらに, 安全性の確保のためビット長を 170 ビットとすると, 有限体上の演算の計算量増加およびビット長が長くなったことによる楕円演算の回数が増加し, その分だけ計算量が増大する. その計算量は, 170 ビットの有限体上の乗算 3106.4 回である. したがって, 残念ながら, 防御法は大変遅いといえる. 提案攻撃法に対する高速防御法は, 議論すべき課題として残っている.

注意 6 提案防御法は n 階 DPA 攻撃 ($n \geq 3$) に対しては脆弱となるかもしれない. 提案防御法が高階 DPA 攻撃に対して耐性を有するか否かは未解決である.

以上議論してきたことをまとめると, 表 4 となる.

参考文献

- 1) Brier, É. and Joye, M.: Weierstrass Elliptic Curves and Side-Channel Attacks, *Public Key Cryptography (PKC 2002)*, LNCS2274, pp.335–345 (2002).
- 2) Cohen, H., Miyaji, A. and Ono, T.: Efficient Elliptic Curve Exponentiation Using Mixed Coordinates, *Advances in Cryptology — ASIACRYPT '98*, LNCS1514, pp.51–65 (1998).
- 3) Coron, J.S.: Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems, *Cryptographic Hardware and Embedded Systems (CHES'99)*, LNCS1717, pp.292–302 (1999).
- 4) National Bureau of Standards: Data Encryption Standard, *Federal Information Processing Standards Publication 46 (FIPS PUB 46)* (1977).
- 5) ECC Challenge. Available at http://www.certicom.com/resources/ecc_chall/challenge.html
- 6) Feller, W.: An Introduction to Probability Theory and Its Applications, Vol.I, 3rd edition, Wiley, New York (1968).
- 7) Fischer, W., Giraud, C., Knudsen, E.W. and Seifert, J.P.: Parallel scalar multiplication on general elliptic curves over \mathbf{F}_p hedged against Non-Differential Side-Channel Attacks, *International Association for Cryptologic Research (IACR), Cryptology ePrint Archive 2002/007* (2002). Available at <http://eprint.iacr.org/>
- 8) Hasan, M.A.: Power Analysis Attacks and Algorithmic Approaches to Their Countermeasures for Koblitz Curve Cryptosystems, *Cryptographic Hardware and Embedded Systems (CHES 2000)*, LNCS1965, pp.93–108 (2000).
- 9) Izu, T. and Takagi, T.: A Fast Parallel Elliptic Curve Multiplication Resistant against Side Channel Attacks, *Public Key Cryptography (PKC 2002)*, LNCS2274, pp.280–296 (2002).
- 10) Joye, M. and Quisquater, J.J.: Hessian elliptic curves and side-channel attacks, *Cryptographic Hardware and Embedded Systems (CHES 2001)*, LNCS2162, pp.402–410 (2001).
- 11) Joye, M. and Tymen, C.: Protections against Differential Analysis for Elliptic Curve Cryptography — An Algebraic Approach, *Cryptographic Hardware and Embedded Systems (CHES 2001)*, LNCS2162, pp.377–390 (2001).
- 12) Koblitz, N.: Elliptic curve cryptosystems, *Math. Comp.* 48, pp.203–209 (1987).
- 13) Kocher, C.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems, *Advances in Cryptology — CRYPTO '96*, LNCS1109, pp.104–113 (1996).
- 14) Kocher, C., Jaffe, J. and Jun, B.: Introduction to Differential Power Analysis and Related Attacks. Available at <http://www.cryptography.com/dpa/technical/index.html>
- 15) Kocher, C., Jaffe, J. and Jun, B.: Differen-

- tial Power Analysis, *Advances in Cryptology — CRYPTO '99*, LNCS1666, pp.388–397 (1999).
- 16) Lim, C.H. and Hwang, H.S.: Fast implementation of Elliptic Curve Arithmetic in $GF(p^m)$, *Public Key Cryptography (PKC 2000)*, LNCS1751, pp.405–421 (2000).
- 17) Liardet, P.Y. and Smart, N.P.: Preventing SPA/DPA in ECC systems using the Jacobi form, *Cryptographic Hardware and Embedded System (CHES 2001)*, LNCS2162, pp.391–401 (2001).
- 18) Messerges, T.S.: Using Second-Order Power Analysis to Attack DPA Resistant Software, *Cryptographic Hardware and Embedded System (CHES 2000)*, LNCS1965, pp.238–251 (2000).
- 19) Miller, V.S.: Use of elliptic curves in cryptography, *Advances in Cryptology — CRYPTO '85*, LNCS218, pp.417–426 (1986).
- 20) Möller, B.: Securing Elliptic Curve Point Multiplication against Side-Channel Attacks, *Information Security (ISC2001)*, LNCS2200, pp.324–334 (2001).
- 21) Oswald, E. and Aigner, M.: Randomized Addition-Subtraction Chains as a Countermeasure against Power Attacks, *Cryptographic Hardware and Embedded Systems (CHES'01)*, LNCS2162, pp.39–50 (2001).
- 22) Okeya, K., Miyazaki, K. and Sakurai, K.: A Fast Scalar Multiplication Method with Randomized Projective Coordinates on a Montgomery-form Elliptic Curve Secure against Side Channel Attacks, *The 4th International Conference on Information Security and Cryptology (ICISC 2001)*, LNCS2288, pp.428–439 (2002).
- 23) Okeya, K. and Sakurai, K.: Power Analysis Breaks Elliptic Curve Cryptosystems even Secure against the Timing Attack, *Progress in Cryptology — INDOCRYPT 2000*, LNCS1977, pp.178–190 (2000).
- 24) Okeya, K. and Sakurai, K.: On Insecurity of the Side Channel Attack Countermeasure using Addition-Subtraction Chains under Distinguishability between Addition and Doubling, *The 7th Australasian Conference in Information Security and Privacy (ACISP 2002)*, LNCS2384, pp.420–435 (2002).
- 25) Pollard, J.M.: Monte Carlo methods for index computation (mod p), *Math. Comp.* 32, pp.918–924 (1978).
- 26) Rivest, R.L., Shamir, A. and Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Comm. ACM*, Vol.21, No.2, pp.120–126 (1978).
- 27) Shanks, D.: Class number, a theory of factorization and genera, *Proc. Symp. Pure Math.* 20, pp.415–440 (1971).
- 28) Teske, E.: Square-root Algorithms for the Discrete Logarithm Problem (A Survey), *Public-Key Cryptography and Computational Number Theory*, pp.283–301, Walter de Gruyter (2001).

(平成 14 年 11 月 29 日受付)

(平成 15 年 6 月 3 日採録)



桶屋 勝幸 (正会員)

1994 年富山大学理学部数学科卒業。1996 年九州大学大学院数理学研究科博士前期課程修了。1998 年 (株) 日立製作所入社。現在, システム開発研究所第 7 部 (セキュリティシステム研究部) 研究員。暗号, 情報セキュリティ技術の研究に従事。電子情報通信学会, 日本数学会, 応用数学会各会員。



櫻井 幸一 (正会員)

1988 年九州大学工学研究科応用物理専攻修士課程修了。同年三菱電機 (株) 入社。現在, 九州大学大学院システム情報科学研究院情報工学部門教授。1997 年 9 月より 1 年間コロンビア大学計算機科学科客員研究員。2001 年 4 月より九州大学システム LSI 研究センター併任。暗号理論・情報セキュリティ・社会情報工学の研究に従事。博士 (工学)。2000 年情報処理学会坂井特別記念賞受賞。2000 年情報処理学会論文賞受賞。電子情報通信学会, 日本数学会, ACM 各会員。