卒業研究論文

WBRIP を用いた電力解析攻撃に安全な楕円曲線のスカラー倍算

Fast scalar multiplication of elliptic curves
secure against power analysis attack using WBRIP

学籍番号　14D8102009B
山下　剛永
YAMASHITA TAKANORI

中央大学理工学部情報工学科
趙研究室

2018 年 3 月

概論

既存研究に電力解析攻撃への高い耐性を持つ楕円曲線暗号のスカラー倍算のアルゴリズムがある．本研究ではその高い耐性を維持しつつ, スカラー倍算の処理時間の短縮を目的とする．本研究では, 電力解析攻撃に耐性を持つ複数のアルゴリズムと手法を組み合わせる事で上記の目的を達成するアルゴリズムを提案する．

キーワード

- 楕円曲線暗号
- スカラー倍算
- 電力解析攻撃

目次

第 1 章

序論

近年, インターネット利用者とその普及率が増加しており, インターネットを使用しての通信は, 我々の生活にとって無くてはならない便利なものとなった. しかしインターネットという便利なものが普及する中で, その技術が悪用されるという事例も目立ってきている. 個人情報の流出やプライバシー保護に関する問題が後を絶たない. 様々な事例がある中でも, 通信の暗号化等の専門的分野では, 情報の漏洩や改ざんを防ぐための研究が盛んに行われている. 暗号技術は, 情報化社会となった現代では必要不可欠なものである. 暗号技術は大きく分けて共通鍵暗号方式と公開鍵暗号方式の二つに分類される. 共通鍵暗号方式では暗号化と復号化に同じ鍵を用いる方式である. 共通鍵は公開されておらず, 送信者と受信者のみで共有されるのが特徴である. 一方で, 公開鍵暗号方式では暗号化, あるいは復号化を行う際に公開鍵と秘密鍵という別々の鍵が使用される. 本研究で扱う楕円曲線暗号は公開鍵暗号に分類され, 楕円曲線上の離散対数問題の求解困難性を安全性の根拠としている. 楕円曲線暗号の大きな特徴として, 他の暗号方式と比較すると省メモリで高速に暗号処理を行う事が可能である. 楕円曲線暗号で 160bit の鍵の安全性は, 同じ公開鍵暗号である RSA 暗号の 1024bit の鍵の安全性に相当する. このように楕円曲線暗号は従来の暗号方式に比べ非常に有用性があるため, 暗号処理速度の向上を目指したアルゴリズムや様々な攻撃への耐性を持たせるための手法が数多く考えられている. 楕円曲線暗号は理論上安全なため, 攻撃は数学的な弱点よりも, 暗号システムの実装部分を標的とした攻撃手法が提案されている. 暗号装置が発する電磁波や熱などを外部から観測し, 暗号解読の手がかりを得ようとする攻撃に, サイドチャネル攻撃というものがあり, その中の一つに電力解析攻撃という攻撃がある. 電力解析攻撃は, 暗号装置が処理する演算や内部情報と消費電力に相関がある事を利用し, 秘密鍵を推測する攻撃手法である. 主に SPA(Simple Power Analysis) 方式と, DPA(Differential Power Analysis) 方式の 2 つの方式に分類される.SPA 攻撃は, 暗号処理一回分の消費電力波形を用いて解析を行う. アルゴリズム内で条件分岐が存在する場合に, 処理される命令が異なる事により, 秘密鍵を復元する事が可能である. 対策法としてはアルゴリズム内に条件分岐を使用しない事や, 条件分岐後の命令を同様にする事が挙げられる. 一方で DPA 攻撃は, 複数の消費電力波形を用い統計的に解析を行う. この攻撃には乱数を用いて計算内容をランダム化する事が対策法となっている. この 2 つの攻撃以外にも Doubling Attack や Template Attack 等の電力解析攻撃があり, それら全ての攻撃に耐性を持たせる事は困難であるが, 耐性を持つアルゴリズム同士を組み合わせることで, より強い耐性を持つアルゴリズムに改良する事が可能である. また, 楕円曲線暗号における暗号処理の計算量はスカラー倍算が多くを占める, そのためスカラー倍算の高速化が暗号処理の時間短縮に繋がる. 本研究ではスカラー倍算の高速化と複数の電力解析攻撃への耐性を両立したアルゴリズムを提案する.

第 2 章

群・環・体

2.1 群の定義

2.1.1 群

集合 G の直積集合 $G \times G$ から集合 G への写像が 1 つ与えられているとき，この写像を G の 2 項演算と呼ぶ．このとき，$G \times G$ の元 (a, b) のこの写像による像を a と b の積といい, 集合 G にひとつの 2 項演算が与えられているといい，$(G, \circ)$ と表す．2 項演算が与えられ，次の三つの条件を全てを満足するとき，G はこの演算に関して群という．

1. 結合法則
   $^\forall a, b, c \in G$ に対して常に，$(a \circ b) \circ c = a \circ (b \circ c)$ が成立する．
2. 単位元の存在
   $e \in G$，$^\forall a \in G$ に対して $a \circ e = e \circ a = a$ が成立する．
3. 逆元の存在
   $^\forall a \in G$，$b \in G$ に対して $a \circ b = b \circ a = e$ が成立する．

   2 項演算が結合法則のみを満たす集合 G と演算 $\circ$ の組 $(G, \circ)$ はこの 2 項演算に関する半群という．

4. 交換法則
   $a, b \in G$ に対して，$a \circ b = b \circ a$ を満たす．

上記 4 つの条件を満たすとき，群 G は可換群またはアーベル群であるという．

集合 G が 2 項演算 $\circ$ に対して可換群であるとき，演算 $\circ$ が $+$ で表される場合その群を加法群と呼ぶ．そのとき $x + y$ を x と y の和といい，単位元を 0，x の逆元を $-x$ で表す．

群 $(G, \circ)$ において 2 項演算が明らかな場合，単に群 G ということもある．

また，群 G に属する元の個数を位数といい，位数が有限であるときは G を有限群，そうでないときは無限群という．

可換群 G の任意の元が 1 つの元 a のべき乗で表せるとき, G を a で生成された巡回群といい, $G = \langle a \rangle$ で表す．a を生成元，あるいは原始元という．

2.1.2 部分群

2 項演算 $\circ$ に関して群 G が与えられているとする．G の部分集合 H が 2 項演算 $\circ$ に関して群であるとき H を G の部分群であるという．

群 G の中で位数が最大の部分群は G 自身であり，最小のものは単位元 e からなる位数 1 の e である．この二つの部分群はどの群に対しても定義することができるため自明な部分群という．また，これ以外の部分群を真部分群とよぶ．

2.2 環の定義

2.2.1 環

2 種類の 2 項演算 (加法 $+$ と乗法 $\cdot$) が定義された集合 R が次の条件を満足するとき，$(R, +, \cdot)$ は環であるいう．

1. 加法に関して可換群をなす．
   $(a+b)+c = a+(b+c)$
   $a+b = b+a$
   $0+a = a+0$
   $(-a)+a = a+(-a) = 0$

2. 乗法に関して半群をなし乗法に関する単位元が存在する．
   $a \in R$ に対して $a \circ e = e \circ a = a$ となる $e \in R$ が存在する．

3. 分配法則
   $a, b, c \in R$ に対して，
   $a \cdot (b+c) = a \cdot b + a \cdot c$
   $(a+b) \cdot c = a \cdot c + b \cdot c$
   が成立する．

更に，環 R において，

4. 交換法則
   $a, b \in R$ に対して, $a \circ b = b \circ a$

を満たすとき，群 R は可換環であるといい，そうでないときを非可換環という．
環における単位元は加法単位元 0_R と，乗法単位元 1_R がある． 環の乗法の記号 $\cdot$ は省略されることが多い． すなわち, $x \cdot y$ は xy と書かれる． 以下この記法で書くことにする．
可換環 R において $a \cdot b$=0 $(a, b \in R)$ ならば a=0 または b=0 が成り立つとき, R を整域という．

2.2.2 部分環, イデアル, 商環

環 R の部分集合でそれ自身が R の演算において, 環になるものを R の部分環という．
環 R の部分集合 I において，

1. $a, b \in I$ ならば, $a+b \in I$
2. $a \in I$ と $r \in R$ に対して, $ra \in I$
3. $a \in I$ と $r \in R$ に対して, $ar \in I$

条件 1. と 2. を満たすとき, I は R の左イデアル, 条件 1. と 3. を満たすとき, I は R の右イデアル, 条件 1. から 3. まで全てを満足するとき, I は R の 両側イデアルまたは単にイデアルという． R が可換環の場合は左イデアル, 右イデアル, 両側イデアルは一致する．
環 R の中のイデアル I の生成元の集合とは, I の元の集合であって, I の任意の元がその集合の元の R 係数の有限

な 1 次結合であるものである． イデアルはもし生成元の有限集合をもつなら, 有限生成といわれる． I が元の集合 $\{f_1, \cdots, f_l\} \subset I$ によって生成されるなら, $I = \sum_{i=1}^{l} Rf_i$, または単に $I = (f_1, \cdots, f_l)$ と書く．

環 R の元 x, y がイデアル I を法として合同であるとは, $x + i = y$ となる元 $i \in I$ が存在することであり, このとき $x \equiv y \pmod{I}$ と書く． この関係は同値関係である． 環 R のイデアル I による同値類を $[x]$ と書けば, $[x] = x + I = \{x + i | i \in I\}$ となり, 同値類の集合 R/I に加法と乗法, つまり,

$$[x] + [y] = [x + y] \ , \ \ [x] \cdot [y] = [xy] \tag{2.1}$$

が定義できる． これらの演算に関して同値類の集合 R/I は環になり, I を法とする R の商環または剰余環 という．

2.2.3 多項式環

可換環 R において, x を不定元 (変数) としたとき, R 上の多項式の集合は,

$$\{R[x] := a_n x^n + \cdots + a_1 x + a_0 \mid a_0, \ a_1, \ \cdots, \ a_n \ \in R, \ n \text{ は } 0 \text{ か正の整数} \} \tag{2.2}$$

と定義される． $f(x) = b_n x^n + \cdots + b_1 x + b_0$が R 上の多項式 で $b_n \neq 0$ としたとき, n を多項式 f の次数といい, $\deg f$ と表す． 特に, $n = 0$ のとき, $f(x) = b_0 \in R$ となるが, これを定数と呼ぶ． $0 \in R$ の次数は $-\infty$ とする． また, 最高次の係数が 1 である多項式をモニック多項式という．

x を不定元とする可換環 R 上の多項式全体の集合には, R における 2 項演算を用いて, 次のように 2 項演算を定義することができる． R 上の 2 つの多項式 $f(x) = b_n x^n + \cdots + b_1 x + b_0$, $g(x) = c_m x^m + \cdots + c_1 x + c_0$ に対して,

$$f(x) + g(x) = \sum_{k \geq 0} (b_k + c_k) x^k \tag{2.3}$$

$$f(x) \cdot g(x) = \sum_{k=0}^{n+m} (\sum_{i+j=k} b_i c_j) x^k \tag{2.4}$$

と 2 つの 2 項演算 $+$ と $\cdot$ を定めると, これに関して, x を不定元とする R 上の多項式の全体集合は可換環になる． ただし, $x^0 = 1$, $0 \cdot x = 0$ $(0, \ 1 \in R)$ と定める． こうして得られた環を, R 上の多項式環と呼び, $R[x]$ と表す．

可換環 R 上の多項式 $f(x)$ が, 1 次以上の多項式 $g(x), \ h(x) \in R[x]$ によって, $f(x) = g(x)h(x)$ となるとき, $g(x) | f(x)$, $h(x) | f(x)$ と表し, $g(x)$, $h(x)$ を $f(x)$ の因子と呼ぶ． $f(x) \in R[x]$ が因子を持たないとき, $f(x)$ は R 上既約であるといわれる． $f(x)$ が既約でないとき, 可約であるという．

T を $T \supset R$ であり, R で定義されている 2 項演算 に対して, 環になっているとする． このとき, 不定元 x に T の元 t を代入することにより,

$$f(t) = b_n t^n + \cdots + b_1 t + b_0 \in T, \ b_i \in R \tag{2.5}$$

が得られる． $f(t) = 0 \in R$ となるときの t を, $f(x)$ の零点という．

2.3　体の定義

可換環の 0 でない元の全てに, 乗法に関する逆元が存在するとき, その可換環を体と呼ぶ. 元の個数が有限な体を有限体という. 特にその個数 q の有限体は $\mathbb{F}_q$ と表す. その元の個数を体の位数という. 体 $\mathbb{K}$ の部分集合 k が $\mathbb{K}$ の部分体であるとは, k が空集合ではなく, 加法と乗法 (それぞれの逆演算も含む.) に関して閉じていることをいう. 体 k が体 $\mathbb{K}$ の部分体であるなら, $\mathbb{K}$ は k の拡大体または単に拡大であるという.

体 $\mathbb{K}$ が, $\mathbb{K}$ に係数を持つ全ての多項式が 1 次因子に完全に分解するという性質を持つなら, $\mathbb{K}$ は代数的に閉じているという. これは, $\mathbb{K}$ に係数を持つ全ての多項式が $\mathbb{K}$ に根を持つことと同値である. 代数的に閉じている最小の $\mathbb{K}$ の拡大体は, $\mathbb{K}$ の代数的閉包と呼ばれ, $\overline{\mathbb{K}}$ で表される.

整数環 $\mathbb{Z}$, 素数 p に対して, p の倍数の集合を $p\mathbb{Z}$ とすると, $p\mathbb{Z}$ は $\mathbb{Z}$ のイデアルである. 商環 $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$ は体になり, p 個の元からなる. これと同型な体を $\mathbb{F}_p$ で表す.

もし $\mathbb{K}$ において, 乗法の単位元 1 をそれ自身に加えていっても決して 0 にならないなら, 体の標数は 0 であるという. そうでない場合, $1+1+\cdots+1(p$ 回) が 0 に等しいような素数 p があり, p は体 $\mathbb{K}$ の標数と呼ばれる. その場合, $\mathbb{K}$ は体 $\mathbb{Z}/p\mathbb{Z}$ の写しを含み, これを $\mathbb{K}$ の素体と呼ぶ.

第 3 章

楕円曲線

3.1 楕円曲線の定義

体 $\mathbb{K}$ 上で定義される楕円曲線とは，一般的に

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (a_1,\ a_2,\ a_3,\ a_4,\ a_6 \in \mathbb{K})$$

で与えられる $(x,\ y)$ に関する方程式のことである．係数 a_n が属する体 $\mathbb{K}$ を係数体，変数 $x,\ y$ が属する体を定義体 $\mathbb{L}$ と呼ぶ．$\mathbb{K}$ は $\mathbb{L}$ の部分体であり楕円曲線の $\mathbb{L}$ 有限体とは，この方程式に無限遠点と呼ばれる要素 $\mathcal{O}$ を加えた x , $y \in \mathbb{L}$ である点 $(x,\ y)$ の集合を表す．
もし、K の標数が 2 であるとき，方程式は

$$y^2 + xy = x^3 + ax^2 + b \quad (a,\ b \in \mathbb{K})$$

と変形され，上式を満たす点の集合となる．
また、K の標数が 3 であるときは，方程式は

$$y^2 = x^3 + ax^2 + bx + c \quad (a,\ b,\ c \in \mathbb{K})$$

と変形され，標数が 3 より大きい場合は

$$y^2 = x^3 + ax + b \tag{3.1}$$

と変形される．
(3.1) 式の形で表される曲線を， Weierstrass 型楕円曲線と呼ぶ．

3.2 楕円離散対数問題 (ECDLP)

任意の点 $P \in E(\mathbb{F}_q)$ に対して，$\langle P\rangle = \{\mathcal{O},\ P,\ 2P,\ 3P,\ \ldots\}$ は有限巡回群となる．この巡回群の位数を n とすると任意の $Q \in \langle P\rangle$ に対して，

$$xP = Q \quad x \in (\mathbb{Z}_n)$$

となる x がただ一つ存在する．P と Q が与えられたとき，x を求める問題を楕円離散対数問題という．これは x と P から $xP = Q$ となる Q を求めるのは簡単だが，P と Q から x を求めるのは非常に困難であることに基づいている．

3.3　楕円曲線上の点の加算と 2 倍算

楕円曲線上の有理点において， 射影座標とは異なる点の加算を定義する． 楕円曲線上の点 P, Q があるとき， まず点 P, Q を通る直線を引き楕円曲線との交点 $P*Q$ を見つける． 次に $P*Q$ と無限遠点を通る直線 (垂直線) を引き楕円曲線と交わるもう 1 つの点を楕円曲線における P と Q が加算された点 $P+Q$ とする． また $P=Q$ のときは楕円曲線との接線を引いて，その直線と楕円曲線との交点を $P*Q$ とする．
この加算によって楕円曲線は群構造をなす． 特に，無限遠点同士の加算は無限遠点となる．点 $P=(x,\ y)$ と点 $Q=(x,\ -y)$ の場合第三の交点は無限遠点 となる．これより 点 Q が点 P の逆元 $-P$ になる．

3.3.1　Weierstrass 型楕円曲線における点の加算と 2 倍算の計算方法

Weierstrass 型楕円曲線

$$y^2=x^3+bx+c$$

の各点 P, Q, $P*Q$, $P+Q$ を
$P=(x_1,\ y_1)$， $Q=(x_2,\ y_2)$， $P*Q=(x_3,\ y_3)$， $P+Q=(x_3,\ -y_3)$
と設定する． このときの P と Q を結ぶ直線の方程式は

$$y=\lambda x+\upsilon,\ \left(\lambda=\frac{y_2-y_1}{x_2-x_1},\ \upsilon=y_1-\lambda x\right)$$

となり，これを楕円曲線の式に代入するとそれぞれ

$$x_3=\lambda^2-x_1-x_2$$

$$y_3=\lambda x_3+\upsilon$$

となる．
また， $P=Q$ のとき $\lambda=\frac{f'(x)}{2y}$ を用いて 2 倍点の傾き λ は

$$\lambda=\frac{3x_1^2+b}{2y_1}$$

と表される．

加算と 2 倍算を使って， 楕円曲線上のスカラー倍算を求めることができる．

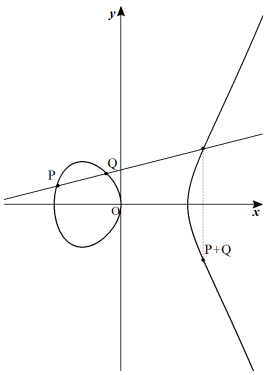

図 3.1　加算

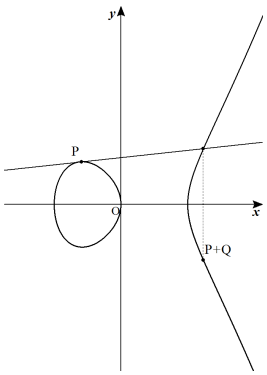

図 3.2　2 倍算

3.4　楕円曲線上の加算と 2 倍算の計算量

楕円曲線上の計算量を，逆元計算 (割算)，乗算，2 乗，加算の演算回数の総和で考える．

- I：逆元計算
- M：乗算
- S：2 乗算
- a：加算

と定義する。

加算 a の計算量は，M と比較すると非常に小さい．

楕円曲線上のスカラー倍算を高速化することは，この計算量を減らすことと同じことと考えられる．

3.4.1　Weierstrass 型楕円曲線における計算量

Weierstrass 型楕円曲線

$$y^2 = x^3 + bx + c$$

の加算の計算量は次のようになる．

$P = (x_1, y_1)$，$Q = (x_2, y_2)$ とし，$P + Q = (x_3, y_3)$ を求めるには，次の計算をおこなう必要がある．

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\upsilon = y_1 - \lambda x_1$$

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda x_3 + \upsilon$$

各演算回数を数えていくと，

$$I + 2M + S + 6a \simeq 22.\ 8M$$

となる．

3.5　楕円曲線上のスカラー倍算

楕円曲線上の点 P に対し，整数 d をかける演算

$$d \times P = \underbrace{P + \cdots + P}_{d\ 個}$$

を，楕円曲線上のスカラー倍算と呼ぶ．

第 4 章

高速化

4.1 射影座標

この章では，楕円曲線上の加算を高速化する上で最も基本となる射影座標 (Projective Coordinates) について解説する．また，この章での楕円曲線は Weierstrass 型楕円曲線 $y^2 = x^3 + ax + b$ を用いることとする．

射影座標とは，$(x,\ y)$ で表現される座標を $(X,\ Y,\ Z)$ の座標に変換して考える手法である．この変換を用いると，途中で逆元演算をせずに加算を行うことができる．

4.1.1 射影座標の定義

楕円曲線上の任意の点 $(x,\ y)$ に対して

$$(x,\ y) \to (X/Z,\ Y/Z)$$

に変換する. このとき Weierstrass 型楕円曲線は

$$Y^2Z = X^3 + aXZ^2 + bZ^3$$

となり，点 P と 点 Q は

$$P = (X_1,\ Y_1,\ Z_1)$$

$$Q = (X_2,\ Y_2,\ Z_2)$$

$$P + Q = (X_3,\ Y_3,\ Z_3)$$

と表される．

4.1.2 変換された座標の固定化による計算時間の高速化

射影座標の変換において変換した変数などを固定値とし，高速化をおこなう場合がある．例えば，Mixed addition という手法では加算の式中の Z_2 を $Z_2 = 1$ と固定し，計算時間を比較している．
他にも，加算では X_2 を $X_2 = 1$ にし, $Z_1 = 1$, $Z_2 = 1$ とするものや，2 倍算でも $Z_1 = 1$ とした方法も考えられている．

4.1.3 楕円曲線の係数の固定化における計算時間の高速化

楕円曲線の点，例えば $y^2 = x^3 + ax + b$ のパラメータ a を固定した値で高速化させる手法が存在する．それには，4.2 で挙げられる jacobian 座標が代表的である．

4.1.4　射影座標を用いた加算の計算量

加算を計算してみると,

$$\begin{aligned} x_3 &= \frac{X_3}{Z_3} \\ &= \lambda^2 - x_1 - x_2 \\ &= \frac{(y_2-y_1)^2}{(x_2-x_1)^2} - x_1 - x_2 \\ &= \frac{(y_2-y_1)^2 - (x_1-x_2)(x_2-x_1)^2}{(x_2-x_1)^2} \\ &= \frac{(y_2-y_1)^2 - (x_2-x_1)^3 - 2x_1(x_2-x_1)^2}{(x_2-x_1)^2} \end{aligned}$$

ここで射影座標を適用し分母を消すと,

$$x_3 = \frac{(Y_2Z_1 - Y_1Z_2)Z_1Z_2 - (X_2Z_1 - Z_1Z_2)^3 - 2X_1Z_2(X_2Z_1 - X_1Z_2)^2}{(X_2Z_1 - X_1Z_2)^2 Z_1Z_2}$$

と表される.

y_3 は最終的に

$$y_3 = \frac{(Y_2Z_1-Y_1Z_2)((X_1Z_2(X_2Z_1-X_1Z_2)^2-(Y_2Z_1-Y_1Z_2)Z_1Z_2-(X_2Z_1-X_1Z_2)^3-2X_1Z_2(X_2Z_1-X_1Z_2)^2))-(X_2Z_1-X_1Z_2)^3Y_1Z_2}{(X_2Z_1-X_1Z_2)^3Z_1Z_2}$$

となる.

計算手順は以下の通りになる.

- $A = Y_2Z_1 - Y_1Z_2$
- $B = X_1Z_2$
- $C = X_2Z_1 - B$
- $D = C^2$
- $E = Z_1Z_2$
- $F = CD$
- $G = A^2E - F - 2BD$
- $X_3 = CF$
- $Y_3 = A(DE - C) - FY_1Z_1$
- $Z_3 = EF$

この射影座標における加算の計算時間は $14M + 2S + 6a$ となり, およそ $15.6M$ である.

この手法だと時間のかかる除算を無視することができるため, Weierstrass 型楕円曲線の計算量 $22.8M$ と比べて計算量が減っていることが分かる.

4.1.5　射影座標を用いた 2 倍算の計算量

2 倍算の場合も加算と同じ考え方である．唯一違う点は，任意の点 (x_1, y_1) ，(x_2, y_2) を $x_1 = x_2$ ，$y_1 = y_2$ と考えればよく，それぞれの式は

$$x_3 = \lambda^2 - 2x_1$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

で表され，射影座標を適用すれば，

$$x_3 = \frac{(a_4 Z_1^2 + 3X_1^2)^2 - 8X_1 Y_1^2 Z_1}{4Y_1^2 Z_1^2}$$

$$y_3 = \frac{(a_4 Z_1^2 + 3X_1^2)(4X_1 Y_1^2 Z_1 - ((a_4 Z_1^2 + 3X_1^2)^2 - 8X_1 Y_1^2 Z_1)) - 8Y_1^4 Z_1^2}{8Y_1^3 Z_1^3}$$

のようになる．

計算手順は以下の通りになる．

- $A = a_4 Z_1^2 + 3X_1^2$
- $B = Y_1 Z_1$
- $C = X_1 Y_1 B$
- $D = A^2 - 8C$
- $E = B^2$
- $X_3 = 2BD$
- $Y_3 = A(4C - D) - 8Y_1^2 E$
- $Z_3 = 8BE$

射影座標における 2 倍算の計算時間は $7M + 5S + 4a$ で, およそ 11M となる．

4.2 jacobian 座標

jacobian 座標とは,

$$(x,\ y) \to (X/Z^2,\ Y/Z^3)$$

の変換をおこなう方法で， このときの楕円曲線は

$$E : Y^2 = X^3 + a_4XZ^4 + a_6Z^6$$

で表現される．以降は, 射影座標と同じように計算し,

$$x_3 = \frac{(Y_2Z_1^3 - Y_1Z_2^3)^2 - (X_2Z_1^2 - X_1Z_2^2)^3 - 2X_1Z_2^2(X_2Z_1^2 - X_1Z_2^2)^2}{(X_2Z_1^2 - X_1Z_2^2)^2Z_1^2Z_2^2}$$

$$y_3 = \frac{(X_1Z_2^2 - X_2Z_1^2)(Z_1 + Z_2)^2 - Z_1^2 - Z_2^2}{2((X_2Z_1^2 - X_1Z_2^2)(Y_1Z_2^3) + (Y_2Z_1^3 - Y_1Z_2^3)((X_1Z_2^2)(X_2Z_1^2 - X_1Z_2^2)^3 - X_3))}$$

と書ける．計算手順は以下の通りになる．

- $A = Z_1^2$
- $B = Z_2^2$
- $C = X_1B$
- $D = X_2A$
- $E = Y_1BZ_2$
- $F = Y_2AZ_1$
- $G = D - C$
- $H = F - E$
- $I = G^2$
- $J = CI$
- $X_3 = 2(-G - 2J + H^2)$
- $Y_3 = G(Z_1 + Z_2)^2 - A - B$
- $Z_3 = 2(-CE + F(J - X_3))$

計算時間は $12M + 4S + 9a$, およそ $15.2M$ と表される．

2 倍算の場合も， 射影座標と同様の方法で計算すると,

$$x_3 = \frac{(3X_1^2 + a_4Z_1^4)^2 - 8X_1Y_1^2}{4Y_1^2Z_1^2}$$

$$y_3 = \frac{3X_1^2 + a_4Z_1^4}{2Y_1Z_1}\left(\frac{X_1}{Z_1^2} - x_3\right) - \frac{Y_1}{Z_1^3}$$

と書けるので，計算手順は以下の通りになる．

- $A = Y^2$
- $B = Z_1^2$
- $C = 4X_1A$
- $D = 3X_1^2 + a_4B^2$
- $X_3 = -2C - D^2$
- $Y_3 = -8A^2 + D(C - X_3)$

- $Z_3(Y+Z_1)^2 - A - B$

計算時間は $4M + 6S + 7a$ で, およそ $8.6M$ である.

$$x_3 = (Z_1Z_2)^2 - (X_1X_2)$$

$$Y_3 = (Z_3 + 2\epsilon(X_1X_2)^2)(Y_1Y_2 - 2\delta X_1X_2Z_1Z_2) + 2\epsilon X_1X_2Z_1Z_2(X_1^2Z_2^2 + Z_1^2X_2^2)$$

ただし， 位数 2 の点が必要であり, $\epsilon = 1$ のとき， 群の位数は 4 で割り切れる.

4.3 Hessian curve

一般の楕円曲線の定義方程式とは違い，特別な形をしている Hessian curves という楕円曲線がある．$q \equiv 2(mod3)$ のとき，$\mathbb{F}_q$ 上に定義された E に対して，Hessian curves を定義できる．ただし， 位数 3 の点が必要である．また，Weierstrass 型楕円曲線との互換性がない.

$$E : y^2 + a_1xy + a_3y = x^3$$

$$\delta = a_1^3 - 27a_3$$

$$\mu = \frac{1}{3}((-27a_3\delta^3)^{\frac{1}{3}} + \delta) \in \mathbb{F}_q$$

Hessian curves

$$H : X^3 + Y^3 + Z^3 = cXYZ$$

$$c = 3\frac{\mu - \delta}{\mu}$$

$(x_1, y_1) \in E(\mathbb{F}_q)$ から $(X_1, Y_1, Z_1) \in H(\mathbb{F}_q)$ への座標変換は

$$X_1 = \frac{a_1(2\mu - \delta)}{3\mu - \delta}x_1 + y_1 + a_3$$

$$Y_1 = \frac{-a_1\mu}{3\mu^\delta}x_1 - y_1$$

$$Z_1 = \frac{-a_1\mu}{3\mu^\delta}x_1 - a_3$$

加算における X_3, Y_3, Z_3 の式は

$$X_3 = X_2Y_1^2Z_2 - X_1Y_2^2Z_1$$

$$Y_3 = X_1^2Y_2Z_2 - X_2^2Y_1Z_1$$

$$Z_3 = X_2Y_2Z_1^2 - X_2Y_2Z_1^2$$

2 倍算における X_3, Y_3, Z_3 の式は

$$X_3 = Y_1(Z_1^3 - X_1^3)$$

$$Y_3 = X_1(Y_1^3 - Z_1^3)$$

$$Z_3 = Z_1(X_1^3 - Y_1^3)$$

4.4 Edwards curve

Edwards curve は, 次の方程式によって定義される.

$$E : x^2 + y^2 = dx^2y^2$$

Edwards curve に射影座標を適用すると, 曲線の方程式は,

$$E : (X^2 + Y^2)Z^2 = Z^4 + dX^2Y^2$$

となる.

加算の計算手順は以下の通りである.

- $A = Z_1Z_2$
- $B = X_1X_2$
- $C = Y_1Y_2$
- $D = (X_1 + Y_1)(X_2 + Y_2) - B - C$
- $E = dBC$
- $X_3 = D(A^2 - E)$
- $Y_3 = (C - B)(A^2 + E)$
- $Z_3 = (A^2 - E)(A^2 + E)$

2 倍算の計算手順は以下の通りである.

- $A = X_1 + Y_1$
- $B = X_1^2 + Y_1^2$
- $C = 2Z_1^2 - B$
- $X_3 = (A^2 - B)C$
- $Y_3 = B(Y_1^2 - X_1^2)$
- $Z_3 = BC$

4.5 既存手法の計算量

本章で紹介した既存手法における楕円曲線上の点の加算と 2 倍算の計算量は, 表 4.1 のようになる.

表 4.1 既存手法の計算量比較

楕円曲線の形	座標系	加算の計算量	2 倍算の計算量
Weierstrass	アフィン	$I + 3M$	$I + 4M$
Weierstrass	射影	$12M + 2S$	$7M + 5S$
Weierstrass	jacobian	$12M + 4S$	$4M + 6S$
Hessian	射影	$12M$	$6M + 3S$
Edwards	射影	$10M + S$	$3M + 4S$

第 5 章

電力解析攻撃

電力解析攻撃は， 実装攻撃であるサイドチャネル攻撃の一種であり， 暗号デバイスが処理する演算や秘密情報と，その消費電力に相関が存在することを利用した攻撃手法である．電力解析攻撃は暗号機能付き IC カードや携帯端末などの暗号デバイスに対し， 現実的な脅威となりうる．

電力解析攻撃は大きく 2 つの方式に分類することができる．暗号処理一回分の消費電力波形を用いる Simple Power Analysis 方式と， 複数の消費電力波形を用い， 統計的に解析することで秘密情報を復元する Differential Power Analysis 方式である．これらは暗号化処理， すなわちスカラー倍算に着目した攻撃であり， 本章で書かれる Algorithm 1，2，3，4, 5 はすべてスカラー倍算を行うものである．

本章ではこれらの攻撃についての概要と対策法を述べる．

5.1 Simple Power Analysis

SPA 攻撃は， 一回の暗号処理の消費電力を観測し， 得られた消費電力波形を用いる選択暗号攻撃である．この攻撃手法は， 秘密情報により暗号デバイスの処理する命令が変化することを利用する．具体例として， Algorithm 1 を挙げる．

表 5.1 Algorithm 1: Binary method

input: $d = (d_{n-1}, \ldots, d_0)_2$, $P \in E(K)$ **output**: dP
$1: Q = \mathcal{O}$ $2:$ **for** $i = n-1$ **down to 0 do** 2. $1: Q = 2Q$ 2. $2:$ **if** $d_i = 1$ 2. 2. $1: Q = Q + P$ $3:$ **end for** $4:$ **Reutrn** Q

このアルゴリズムは SPA 攻撃に対して脆弱性を持つ．秘密鍵である d による条件分岐が存在し， $d_i = 0$ の場合と， $d_i = 1$ の場合で処理される命令が異なるため， 詳細な消費電力波形を観察することで秘密鍵 d を復元することができる．

5.2 SPA 攻撃の対策法

SPA 攻撃の対策法は, 楕円曲線を加算と 2 倍算の電力波形が同じになる曲線を用いる方式で, アルゴリズム中の条件分岐後における命令を同じにする方式の二つがある.

5.2.1 Indistinguishable Point Addition Formulae

Indistinguishable Point Addition Formulae は Weierstrass 型楕円曲線を用いるのではなく Hessian curves 曲線等の特別な曲線を用いることで，加算と倍算が同じ消費電力波形となる．秘密情報に依存した消費電力波形が観測されないために SPA 攻撃に耐性をもつ.

5.2.2 条件分岐による対策法

Algorithm 1 は以下のように改善することができる.

表 5.2 Algorithm 2: Double and always add

input: $d = (d_{n-1}, \ldots, d_0)_2$, $P \in E(K)$ **output**: dP
1 : $Q_0 = \mathcal{O}$
2 : **for** $i = n-1$ **down to 0 do**
2. 1 : $Q_0 = 2Q_0$
2. 2 : $Q_1 = Q_0 + P$
2. 3 : $Q_0 = Q_{d_i}$
3 : **end for**
4 : **Return** Q_0

d_i の値によらずに演算を行うように 2. 2 行の内容を変更した． $d_i = 1$ でも $d_i = 0$ でも， 処理される命令は全く同じであるため， 消費電力波形から d を得ることはできない.

高速にスカラー倍算を行う対策法として Montgomery Ladder がある．x 座標のみを用いて計算を行う手法であり，Double and always add よりも高速に計算を行うことができる.

表 5.3 Algorithm 3: Montgomery ladder

input: $d = (d_{n-1}, ..., d_0)_2, P \in E(K)$ **output**: dP
1 : $R[0] = \mathcal{O}, R[1] = P$
2 : **for** $i = n-1$ **down to** 0 **do**
2.1 : $R[1-d_i] = R[1-d_i] + R[d_i]$
2.2 : $R[d_i] = 2R[d_i]$
4 : **Return** $R[0]$

秘密鍵 d によらず処理される命令はまったく同じであるために，消費電力波形から d を得ることはできない.

5.3 Differential Power Analysis

DPA 攻撃は， 暗号処理の消費電力を多数観測し， 得られた複数の消費電力波形を統計的に解析することで秘密情報を復元する．楕円曲線上のスカラー倍算に対する一般的なこの攻撃の手順は以下のようになる．

1. $(d_{n-1}, \ldots, d_0)_2$ を 2 進展開とし， 攻撃者はその上位にビット $d_{n-1}, \ldots, d_{j+1}$ を知っているものと仮定する．攻撃者は j 番目のビット d_j を推測する．
2. 攻撃者はランダムに m 個の点 $P_1, \ldots, P_m$ を選び， $Q_k = (\sum_{i=j}^{m-1} d_i)P_k$ を $1 \leq k \leq m$ で計算し， その m 回分のサイドチャネル情報 $T(k)$ を観測する．
3. 攻撃者は， m 個 $T(k)$ を 2 つの集合 S_{true} と S_{false} に分けるような， 選択関数 $C(Q_k)$ を定義する．この選択関数は d_i の予想が正しかった場合， 相関が存在するもの同士で S_{true} と S_{false} に振り分け， 予想が間違っていた場合はランダムに振り分けられるように設定する．
4. 選択関数 C により分けられた集合をそれぞれ平均し， その差分を取る．つまり， 以下のようなグラフ g を求める．

$$g = \frac{1}{|S_{true}|} \sum_{k=1,\ldots,\ m|Tk \in S_{true}} Tk - \frac{1}{|S_{false}|} \sum_{k=1,\ldots,\ m|Tk \in S_{false}} Tk$$

ただし， $|S_{ture}| + |S_{false}| = m$ である．もし d_j の予想が間違っていた場合， サイドチャネル情報 $T(k)$ は 2 つの集合へランダムに振り分けられるため， $g \approx 0$ となり， g は平坦なグラフとなる， d_j の予測が正しかった場合には， $T(k)$ は相関のある集合に分配されるため， g に相関を表すピークが現れる． d_j を得られたならば， この操作を繰り返すことで残りの $d_{j-1,\ldots,\ d_0}$ を求めることができる．
例えば， Algorithm 2 において， 最下位ビットである d_{n-1} が $d_{n-1} = 1$ となるならば， $i = n-2$ において， $2P$ が計算されることになり， そうでなければ $2P$ は計算されない．相関を得ることができるように $2P$ の特定のビットに着目し， 選択関数 $C(Q_k)$ を設定する．

5.4 DPA 攻撃の対策法

DPA 攻撃への最も有効な対策法は， 乱数によって演算や, 秘密鍵, 平文やポイント P をランダム化することである．J.S.Coron により 3 つの対策法と, C.Tymen により楕円曲線の同型写像を利用した対策法が示された [9] [12]. Coron の 3 つの対策法を以下に示す．

1. Randomization of the Private Exponent
   $\#\varepsilon$ を用いる曲線の有理点の個数とする． $Q = dP$ のスカラー倍算を以下のように行う．
   (a) ランダムに $r \in K^*$ を選ぶ.
   (b) $d' = d + \#\varepsilon r$
   (c) $Q = d'P$ を計算し, $\#\varepsilon P = \mathcal{O}$ となるため， $Q = dP$ を得る．
2. Base Point Blinding
   楕円曲線 E 上のランダムな点 $R \in E$ を用いることで， P の演算を隠す．$S = dR$ とし， 以下のように演算を行う．
   (a) ランダムに $b \in 0, 1$ を選ぶ.
   (b) $R = (-1)^b 2R$ とする．
   (c) $S = (-1)^b 2S$ とする．
   (d) $Q' = d(P + R)$ を演算する．
   (e) $Q' - S$ から $Q = dP$ を得る．

しかし，このままでは計算量が非常に多くなってしまうため，効率的に運用するためには，あらかじめ R と S を記憶する必要がある．

3. Randomized Projective Coordinates
   点 $P = (x, y)$ と乱数 r を用いて， 点 P ランダムな射影座標上の点 $P' = (rx : ry : r)$ へ移せることを利用する．具体的な処理方法は以下のようになる．
   (a) 乱数 $r \in K^*$ を選択し， $P' = (rx : ry : r)$ とする．
   (b) $Q' = (X : Y : Z) = dP'$ を計算する．
   (c) 得られた Q' を逆変換し， $Q = (X/Z, Y/Z) = dP$ を得る．
   新しい演算を行うたびに新たな乱数 r を選び，異なる射影座標を用いることで DPA 攻撃を防ぐことができる．

Randomized Isomorphic Elliptic Curve
射影座標を用いる対策法と似ているが， この方法はランダムな同型写像を利用する．点 P を同型な曲線上の点 $P' = (r^{-2}x, r - 3y)$ へ写し， その新たな曲線上で演算した後， 逆変換することで $Q = dP$ を得る．処理の手順は以下のようになる．

1. ランダムに $r \in K^*$ を選ぶ.
2. 新たな点 $P' = (r^{-2}x, r - 3y)$ と $a' = r^{-4}a$ を計算する．
3. $Q' = (x', y') = dP'$ を $E(K)$ と同型な楕円曲線 $E'(K)$ で演算する．ただし $E'(K) : y^2 = x^3 + a'x + b'$ とする．
4. 逆変換を $Q = (r^2x', r^3y') = dP$ のように行うことで Q を得る．

P' を射影座標上の点として表現すると $P' = (r^{-2} : r - 3y : 1)$ となり， Z 座標が 1 に等しくなる．これを利用すると J. S. Coron の 3 番目の対策法よりも早く演算を行うことができる．また b' は， このスカラー倍算において関与しない．

5.5 Refined Power Analysis， Zero-value Point Attack

RPA 攻撃は DPA 攻撃を応用した攻撃手法であり， ZPA はそれをさらに一般化したものである．この 2 つの攻撃はスカラー倍算の途中で， 0 の値を持つ特殊な点が現れるように点 P を選び演算させることで， 有効な DPA 対策であるランダムな射影座標， 同型写像を用いた防御法を破ることができる．

RPA 攻撃は L. Goubin により示された， メッセージ選択方式の DPA 型の攻撃である [11]．この攻撃手法は， $(x, 0)$ と $(0, y)$， 射影座標上の表現では $(X : 0 : Z)$ と $(0 : Y : Z)$ という 2 つの点を利用している．これらの点は， J. S. Coron の 3 番目の対策法を用いてランダム化されていても， $(rX : 0 : rZ)$ や $(0 : rY : rZ)$ のように 0 の値は消えることなく残る．また同様に同型写像を用いた防御法も， これらの特徴をランダム化することができない．よって， DPA を用いることで， スカラー倍算において演算される， 上記のような点を感知することができる．

Algorithm 2 の場合， RPA 攻撃は以下のように適用される．d の 2 進展開を $(d_{n-1}, \ldots, d_0)_2$ とし， 攻撃者はその上位ビット $d_{n-1}, \ldots, d_{j+1}$ を知っているものと仮定する．任意の点 P において， i 番目のループの最後で Q_0 は，

$$Q_0 = (\sum_{j=i+1}^{n-1} d_j 2^{j-i} + d_i)P$$

となる．そして， 次の 2 つの状況のどちらかとなる．

- もし $d_i = 0$ ならば， $i+1$ 番目のループにて次のような 2 つの値が計算される．

$$Q_0 = (\sum_{j=i+1}^{n-1} d_j 2^{j-i+1})P$$
$$Q_1 = (\sum_{j=i+1}^{n-1} d_j 2^{j-i+1} + 1)P$$

- もし $d_i = 1$ ならば， $i+1$ 番目のループにて次のような 2 つの値が計算される．

$$Q_0 = (\sum_{j=i+1}^{n-1} d_j 2^{j-i+1} + 2)P$$
$$Q_1 = (\sum_{j=i+1}^{n-1} d_j 2^{j-i+1} + 3)P$$

この説で述べた特殊な点を P_0 とし， 選択する点を P_1 とする．ここで， $d_i = 0$ と予測するならば，

$$P_1 = \{(\sum_{j=i+1}^{n-1} d_j 2^{j-i+1} + 1)^{-1} \mathrm{mod} |\mathrm{E(K)}\} \mathrm{P_0}$$

とし， もし $d_i = 1$ と予測するならば，

$$P_1 = \{(\sum_{j=i+1}^{n-1} d_j 2^{j-i+1} + 3)^{-1} \mathrm{mod} |\mathrm{E(K)}|\} \mathrm{P_0}$$

とする．

dP_1 を m 回演算 した消費電力波形を $T(k)$ と置き， $1 \leq k \leq m$ とする．ここで消費電力波形の平均を取る．

$$g = \frac{1}{m} \sum_{k=1}^{m} T(k)$$

もし d_i の推測が間違っていた場合， $i+1$ 番目に現れる点は P_0 ではなく， それぞれの計算においてランダム化された点になるため， $g \approx 0$ となり， 平坦な波形を得る．d_i の推測があっていた場合， $i+1$ 番目のループにおいて P_0 が現れ， 平均化した波形である g に相関を表すピークが生まれる．

d_i を得ることができたら， 同様の操作を繰り返すことで残りの $d_{i-1}, \ldots, d_0$ を復元できる．この攻撃手法は $(x, 0)$ または $(0, y)$ が存在する楕円曲線に対し有効である．$(x, 0)$ の位数は 2 であるため， $(x, 0)$ は 2 ではない素位数である曲線には存在しない．また， $(0, y)$ は b が平方剰余であれば存在する．

ZPA は T. Akishita らに提案された手法であり， 基本的な構想は RPA 攻撃と変わりはないが， $(x, 0)$ や $(0, y)$ だけでなく， 楕円曲線上の加算， 2 倍算の計算途中にも 0 の値を演算させることができるため， より多くの曲線において RPA 攻撃を適用できることが示された [13]．例えば， Jacobian 座標上で楕円曲線上の加算と 2 倍算は以下のようになる．

Elliptic curve doubling in Jacobian coordinate

$X_3 = T$, $Y_3 = -8Y_1^4 + M(S - T)$, $Z_3 = 2Y_1 Z_1$,
$S = 4X_1 Y_1^2$, $M = 3X_1^2 + aZ_1^4$, $T = -2S + M^2$

Elliptic curve adding in Jacobian coordinate

$X_3 = -H^3 - 2U_1 H^2 + R^2$, $Y_3 = -S_1 H^3 + R(U_1 H^2 - X^3)$, $Z_3 = Z_1 Z_2 H$,
$U_1 = X_1 Z_2^2$, $U_2 = X_2 Z_1^2$, $S_1 = Y_1 Z_2^3$, $S_2 = Y_2 Z_1^3$, $H = U_2 - U_1$, $R = S_2 - S_1$

このように， 2 倍算においては M や $S - T$， 加算においては H や R で， X 座標, Y 座標 が 0 の値を持たなく

ても， 演算途中に 0 の値の計算が起こりうることがわかる．

5.6 RPA 攻撃， ZPA の対策法

4．5 節で述べたように， SPA や DPA に対する有効な防御法の一部は， RPA や ZPA を防ぐことができない．これらの手法に対しては， Coron の 1 番目や 2 番目の対策法である平文や点 P のランダム化が有効である．

ランダムな射影座標や同型写像を用いた対策法は， 特殊な点を隠すことができないため， DPA を防ぐことができても RPA や ZPA により解読することができる．しかし， Coron の 2 番目の対策法のようにランダムな点 R を用いた場合， 計算途中に 0 の値を持った点を出すために選択した， 特殊な点 P_1 が別の点へ移ってしまうために， RPA と ZPA を用いて秘密情報 d を得ることができなくなる．また新しい演算を行うたびに新しいランダムな点を選びなおすことで， 安全性を保つことができる．

5.7 Doubling Attack

Doubling Attack は RPA/ZPA 攻撃と同じくメッセージ選択方式の攻撃であるが，DPA 攻撃とは違い相関関数を用いるのではなく，2 倍算に注目した攻撃である．同じメッセージを二回以上入力，スカラー倍算の途中で秘密鍵の 2 進展開が 0 であった際には 2 倍算が同じ処理になりそれを検知することができることを利用している．
Doubling Attack は RPA/ZPA 攻撃に有効であった平文や点 P のランダム化の対策がされていても 2 倍算が同じ処理になることを検知できるために，これらの対策法を破ることができる．d=78 を例に Doubling Attack の例を示す．
d=78=64+8+4+2 , n=6　なので，

$$(d_0, d_1, d_2, d_3, d_4, d_4, d_6) = (0, 1, 1, 1, 0, 0, 1)$$

k ステップ目のサイドチャネル情報を T_k とするとスカラー倍算のサイドチャネル情報 $T_k(P)$ は

$$T_k(P) = \sum_{i=0}^{k} d_{n-i} 2^{k-i} \times P \tag{5.1}$$

$$= \sum_{i=0}^{k-1} d_{n-i} 2^{k-1-i} \times (2P) + d_{n-k} \times P \tag{5.2}$$

$$= T_{k-1}(2P) + d_{n-k} \times P \tag{5.3}$$

各ビットごとにサイドチャネル情報を見ていくと

ステップ数 k	0	1	2	3	4	5	6
d_{n-k}	1	**0**	**0**	1	1	1	**0**
$T_k(P)$	P	**2P**	**4P**	9P	19P	39P	**78P**
$T_k(P2)$	**2P**	**4P**	8P	18P	38P	**78P**	156P

となり，d_{n-k}=0 のときに $T_k(P)$ と $T_{k-1}(2P)$ が等しくなるとために秘密情報がわかってしまう．

5.8 Doubling Attack の対策法

平文や点 P がランダム化の対策がされていても Doubling Attack を防ぐことはできない．有効な防御法は二回同じ演算をしないことである．よってランダムな射影座標や同型写像をもちいた対策法が有効である．これらの対策法によりランダム化された 2 倍算は異なる計算になるために同じ処理を検知することができない．新しい演算を行うたびに新しい乱数によりこれらをランダム化することで安全性を保つことができる．

5.9 既存の防御手法-Mamiya の方法

ここでは既存の電力解析攻撃に対する防御手法の概要を示す． Algorithm 2 を元にした電力解析攻撃に対する対策法がある．それが下のアルゴリズムの H．Mamiya らにより提案された，SPA 攻撃や DPA 攻撃，RPA 攻撃，ZPA の対策法である [8]．基本的な構想は Coron の点 P のランダム化による対策法を用いている．なお n は秘密鍵 d の 2 進展開でのビット長である．

表 5.4 Algorithm 4: Mamiya 法

input: $d = (d_{n-1}, \ldots, d_0)_2$, $P \in E(K)$ **output**: dP
Pre-computation 1 : **Choose a random point** $R \in E(K) \setminus \mathcal{O}$ 2 : $T_0 = R$, $T_1 = -R$, $T_2 = P - R$
3 : **for** $i = n - 1$ **down to** 0 **do** 3．1 : $T_0 = 2T_0$ 3．2 : $T_0 = T_0 + T_{1+d_i}$ 4 : **Return** $T_0 + T_1$

このアルゴリズムには分岐がないため SPA 攻撃に対して安全である．さらにランダムな点 R を用いることにより，点 P をランダム化し DPA， RPA 攻撃， ZPA に対し安全である．このランダムな点 R を用いることからこの方式は BRIP(Basic SPA-resistant algorithm with Random Initial Point) と呼ばれる．以下 BRIP と呼ぶ．
この方式は秘密鍵の 2 進展開での左端のビットから 1 ビットずつ倍算と加算を繰り返し， スカラー倍算 dP を求める公式である．よって鍵長が 160 ビットの場合， 計算量は加算を 160 回， 2 倍算 160 回行っている．

5.10 Mamiya 法の高速化

BRIP を安全性を保ちつつ高速化のための工夫を加えた手法として， WBRIP(Window-Based Algorithm with RIP) が存在する [8].

5.10.1 WBRIP(Window-Based Algorithm with RIP)

WBRIP は window 法 [3] を BRIP に適用し高速化を図る方式である．具体的なアルゴリズムは以下のとおりである．

表 5.5 Algorithm 5: WBRIP(width-2 of window)

input: $d = (d_{n-1}, \ldots, d_0)_2$, $P \in E(K)$ **output**: dP
Pre-computation 1 : **Choose a random point** $R \in E(K) \setminus \mathcal{O}$ 2 : $T[2] = R$, $T[1] = -R$, $T[0,\ 0] = -3R$, $T[0,\ 1] = P - 3R$, $T[1,\ 0] = 2P - 3R$, $T[1,\ 1] = 3P - 3R$
3 : **for** $i = n - 1$ **down to** 0 **do** 3. 1 : $T[2] = 4T[2]$ 3. 2 : $T[2] = T[2] + T[d_i,\ d_{i-1}]$ 3. 3 : $i = i - 2$ 4 : **Return** $T[2] + T[1]$

ウィンドウ法はスカラー d を m 進数に展開することで計算回数を減らすテクニックである．このアルゴリズムではウィンドウ幅を 2 とし， d を 4 進数としている．計算量は $(\frac{n}{2} + 4)$ 回の加算， n 回の 2 倍算で， 鍵長 160 ビットの場合， 84 回の加算と 160 回の 2 倍算で約 24% の計算量の削減を行える．

第 6 章

既存手法

既存手法は Random Projective Coordinate, Base Point Blinding, Montgomery Ladder を組み合わせることで, 電力解析攻撃への高い耐性を得ている.

6.1 既存手法のアルゴリズム

表 6.1 提案手法

input: $k = (k_{n-1}, ..., k_0)_2, P \in E(K)$
output: kP
Pre-computation
1 : **Choose a random point** $S \in E(K)$
2 : $P(x_1, y_1), S(x_2, y_2) \rightarrow (X_1, Y_1, Z_1), (X_2, Y_2, Z_2)$
3 : **Choose a random number** r
4 : $(X_1, Y_1, Z_1), (X_2, Y_2, Z_2) \rightarrow P'(rX_1, rY_1, rZ_1), S'(rX_2, rY_2, rZ_2)$
5 : $T = P' + S', U[0] = \mathcal{O}, U[1] = T, V[0] = \mathcal{O}, V[1] = S'$
6 : **for** $i = n - 1$ **down to** 0 **do**
6.1 : $U[1 - d_i] = U[1 - d_i] + U[d_i]$
6.2 : $V[1 - d_i] = V[1 - d_i] + V[d_i]$
6.3 : $U[d_i] = 2U[d_i]$
6.4 : $V[d_i] = 2V[d_i]$
8 : $Q' = U[0] - V[0]$
9 : $Q'(X, Y, Z) \rightarrow Q(x, y)$
10 : **Return** Q

6.2 既存手法の耐性

既存手法は SPA 攻撃，DPA 攻撃，RPA 攻撃，ZPA，Doubling Attack，Template Attack に対して耐性をもつ．

対策法	SPA	DPA	RPA/ZPA	Doubling Attack	Template Attack
Montgomery Laddar	○	-	-	-	-
Base Point Blinding	-	○	○	-	-
Random projective Coordinates	-	○	-	○	○
既存手法	○	○	○	○	○

6.3 既存手法の計算量

この手法の加算, 倍算は, 射影座標上で行うので計算量はそれぞれ,

加算 $4M + 2S \simeq 5.6M$

倍算 $3M + 2S \simeq 4.6M$

となる．

また, 既存アルゴリズムの鍵長が 16bit, 32bit, 64bit, 128bit, 256bit, 512bit の場合の加算, 倍算の総回数と総計算量を表 6.2 に記す．

表 6.2　計算回数

鍵長 (bit)	加算 (回)	倍算 (回)	総計算量 (M)
16	33	33	336.6
32	65	65	663.0
64	129	129	1315.8
128	257	257	2,621.4
256	513	513	5,232.6
512	1025	1025	10,455.0

第 7 章

提案手法

本研究では, Random Projective Coordinate と WBRIP(width) を組み合わせることで, 既存手法の持つ電力解析攻撃に対する高い耐性を維持しつつ, 高速に暗号処理を行う事が可能なスカラー倍算アルゴリズムを提案する.

7.1 提案手法のアルゴリズム

表 7.1 提案手法

input: $k = (k_{n-1}, ..., k_0)_2, P \in E(K)$ **output**: kP
Pre-computation 1 : **Choose a random point** $R \in E(K)$ 2 : $P(x_1, y_1), R(x_2, y_2) \rightarrow (X_1, Y_1, Z_1), (X_2, Y_2, Z_2)$ 3 : **Choose a random number** r 4 : $(X_1, Y_1, Z_1), (X_2, Y_2, Z_2)$ $\rightarrow P'(rX_1, rY_1, rZ_1), S'(rX_2, rY_2, rZ_2)$ 5 : $T[2] = R, T[1] = -R$, $T[0,0] = -3R, T[0,1] = P - 3R$, $T[1,0] = 2P - 3R, T[1,1] = 3P - 3R$
6 : **for** $i = n - 1$ **down to** 0 **do** 6.1 : $T[2] = 4T[2]$ 6.2 : $T[2] = T[2] + T[k_i, k_{i-1}]$ 6.3 : $i = i - 2$ 7 : **Return** $T[2] + T[1]$

7.2 提案手法の耐性

提案手法は既存手法と同様に, SPA 攻撃，DPA 攻撃，RPA 攻撃，ZPA，Doubling Attack，Template Attack に対して耐性をもつ.

対策法	SPA	DPA	RPA/ZPA	Doubling Attack	Template Attack
WBRIP	○	○	○	-	-
Random projective Coordinates	-	○	-	○	○
提案手法	○	○	○	○	○

7.3 提案手法の計算量

この手法の加算, 倍算は, 射影座標上で行うので計算量はそれぞれ,

加算 $4M + 2S \simeq 5.6M$

倍算 $3M + 2S \simeq 4.6M$

となる.

また, 提案アルゴリズムの鍵長が 16bit, 32bit, 64bit, 128bit, 256bit, 512bit の場合の加算, 倍算の総回数と総計算量を表 7.2 に記す.

表 7.2 計算回数

鍵長 (bit)	加算 (回)	倍算 (回)	総計算量 (M)
16	15	19	171.4
32	23	35	289.8
64	39	67	526.6
128	71	131	1,000.2
256	135	259	1,947.4
512	263	515	3,841.8

第 8 章

実験

8.1 実装環境

計測に使用した環境は以下のものである.

- CPU : Intel Core i5 2.9GHz
- Memory : 8.0GB
- OS : Macintosh High Sierra
- 言語 : Java

8.2 実装条件

任意の楕円曲線上の点 10,000 点に対して, 既存手法と提案手法でスカラー倍算を行い, その実行時間を計測し平均を求める. 本研究では, 鍵長が 16bit, 32bit, 64bit, 128bit, 256bit, 512bit の場合について検証する. 楕円曲線の標数は NIST(米国立標準技術研究所) が推奨する楕円曲線 P-521 の標数を使用する.

楕円曲線 P-521 のパラメータは以下の通りである.

$$y^2 = x^3 - 3x + b$$

$$b = 1093849038073734274511112390766805569936207598951683748994586394495953116150735$$
$$0160137087375737596232485921322967063133094384525315910129121423274884789859 84$$

標数 q = 6864797660130609714981900799081393217269435300143305409394463459185543183397656
052122559640661454554977296311391480858037121987999716643812574028291115057151

8.3　結果

実験の結果を表 8.1 に記す.

表 8.1　結果

鍵長 (bit)	既存アルゴリズム (ms)	提案アルゴリズム (ms)
16	31.0042	11.2159
32	61.7193	22.2341
64	123.5217	42.8427
128	246.1405	82.9316
256	490.7483	160.5801
512	992.7893	317.3729

表 8.1 の結果をグラフで記す.

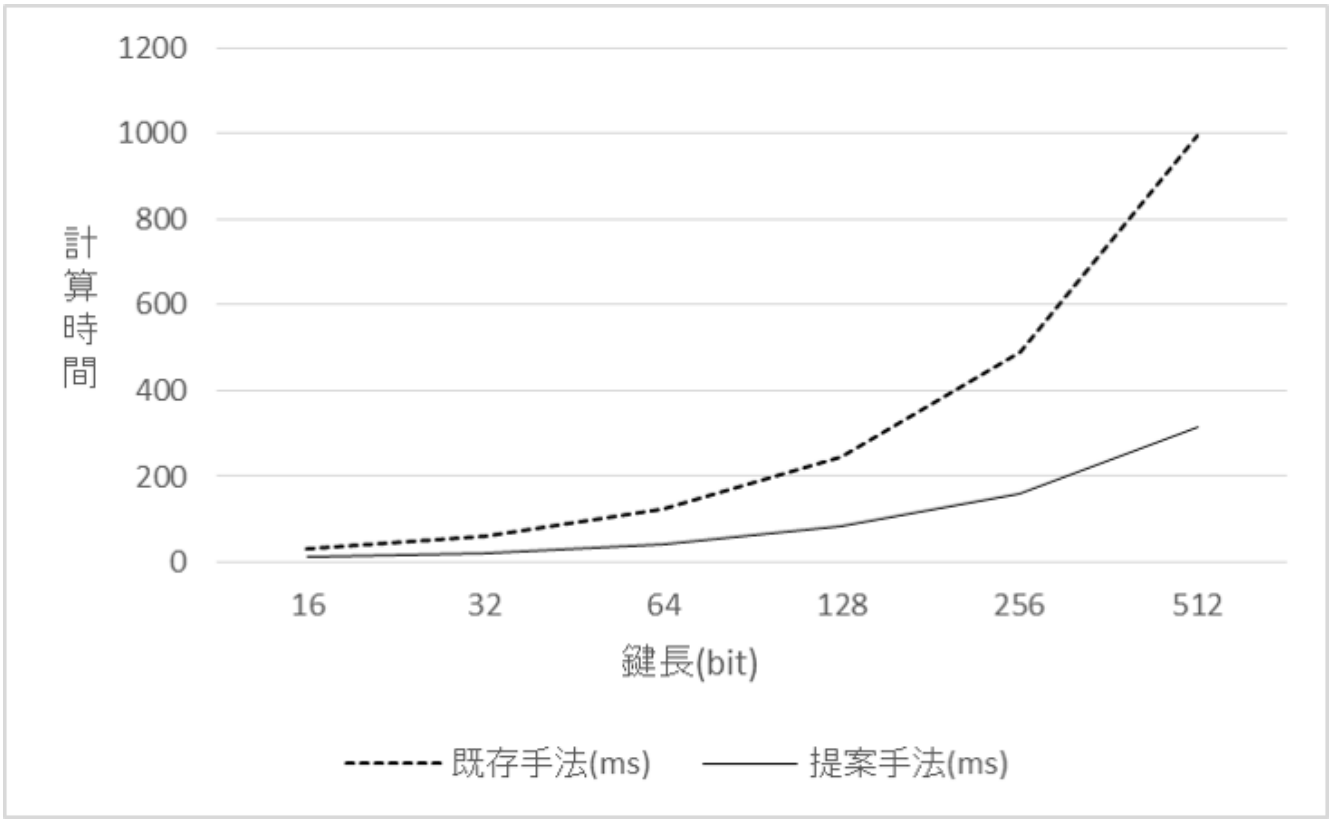

図 8.1　グラフ

第 9 章

結論

提案手法では, 電力解析攻撃への耐性を高いレベルで維持したまま, 既存手法よりも短い計算時間でスカラー倍算を行う事ができた. 計算時間の削減については, 本実験におけるいずれの鍵長の場合においても, 約 30% 前後に削減されていた. この削減率は第 6 章と第 7 章で述べた, 提案手法と既存手法の総計算量と相関がみられる. そのため本実験の整合性を裏付ける事ができた. 今後の課題としては, 各鍵長ごとに最適な WBRIP の幅を調べて検証する事や, さらなる高速化のために Edwards Curve 等の特殊な楕円曲線を用いて実験を行う事が挙げられる.

謝辞

本研究を進めるにあたり，適切な御指導，御助言，御検討を頂いた中央大学 理工学部 趙 晋輝 教授に，深く感謝いたします．また, 日常の議論を通じて多くの知識や示唆を頂いた趙研究室の皆様に深く感謝いたします.

参考文献

[1] 辻井重男, 笠原正雄, 有田正剛, 境隆一, 只木孝太郎, 趙晋輝, 松尾和人, **暗号理論と楕円曲線**, 森北出版, 2008.

[2] 塩野耀, **電力解析攻撃に耐性を持つ楕円スカラー倍算高速化アルゴリズムの提案**, 中央大学理工学部情報工学科卒業論文, 2016.

[3] Bos, J.and Coater, M: *Addition chain heuristics*, Proc. of CRYPTO'89(1989).

[4] D. J. Bernstein, T. Lange: *Analysis and Optimization of Elliptic-curve Single-scalar Multiplication*, Cryptology ePrint Archive, 2007/455, IACR, December 2007.

[5] D. J. Bernstein, T. Lange: *Faster addition and doubling on elliptic curves*, ASIACRYPT, 2007.

[6] Fouque PA., Valette F. : *The Doubling Attack Why Upwards Is Better than Downwards.* , Cryptographic Hardware and Embedded Systems, vol 2779, Berlin, Heidelberg ,2003.

[7] Hideyo Mamiya, Atsuko Miyaji, and Hiroaki Morimoto, *Efficient Countermeasures Against RPA, DPA, and SPA* , Cryptographic Hardware and Embedded Systems, Volume 3156 pp 343-356,Berlin, Heidelberg ,2004.

[8] H.Mamiya, A.Miyaji, and H.Morimoto: *Efficient countermeasure against RPA, DPA, and SPA*, Cryptographic Hardware and Embedded Systems , pp.343-356, Springer-Verlag,Berlin, Heidelberg, 2004.

[9] J.S.Coron:*Resistance against differential power analysis for elliptic curve cryptosystems*,Cryptographic Hardware and Embedded Systems , pp.292-302, Springer-Verlag, 1999.

[10] Lawrence C. Washington (2008) Elliptic Curves Number Theory and Cryptography, US-VA, Taylor & Francis Group

[11] L. Goubin : *A refined power-analysis attack on elliptic curve cryptosystems.* in Proceedings of PKC 2003, LNCS 2567, pp. 199-211. ,Berlin, Heidelberg ,2003.

[12] M.Joye, C.Tymen:*Protections against differential analysis for elliptic curve cryptography*, Cryptographic Hardware and Embedded Systems, pp.377-390, Berlin, Heidelberg, 2001.

[13] T.Akishita, T. Takagi: *Zero-Value Point Attacks on Elliptic Curve Cryptosystem*, International Continence Society 2003, Lecture Notes in Computer Science, pp.218-233, Springer-Verlag, Berlin, Heidelberg, 2003.