反復回数改ざんを利用した楕円曲線暗号に対するフォールト攻撃

Fault attacks to elliptic curve cryptosystems by tampering the number of iterations

12D8101003C　渡邊　千尋
中央大学理工学部情報工学科 趙研究室
2016 年 3 月

要約　フォールト攻撃は暗号装置の実装上の脆弱性をつく強力な攻撃である．本研究では, Montgomery ladder などを用いる楕円曲線暗号において, スカラー倍算の反復回数を改ざんする新しいフォールト攻撃手法を示す.

キーワード　楕円曲線暗号, フォールト攻撃

1　序論

現在, あらゆる情報機器に暗号が実装されており, 暗号の安全性がセキュリティの根拠となっている．フォールト攻撃とは, 暗号装置への物理的干渉により意図的に機器にエラーを起こさせ, 秘密情報を推測する攻撃手法である．本研究では, 楕円曲線暗号におけるスカラー倍算中のループ回数をフォールトの導入対象とした新しい攻撃手法を提案する．さらに, NIST 推奨パラメータを用いた楕円曲線暗号への攻撃を行い, 攻撃の有効性を示した.

2　楕円曲線

2.1　楕円曲線の定義

標数が 3 より大きい有限体 $\mathbb{F}_q$ 上に定義される Weierstrass の標準型の楕円曲線 E は以下の式で表される.

$$E : y^2 = x^3 + ax + b \quad (a, b \in \mathbb{F}_q) \qquad (1)$$

2.2　楕円曲線上の点のスカラー倍算

楕円曲線上の点のスカラー倍算とは, 楕円曲線上の任意の点 $P \in E(\mathbb{F}_q)$ と, 任意の整数 $n \in \mathbb{Z}$ に対して

$$nP = \begin{cases} P + P + \cdots + P \ (n \text{ 回}) & (n > 0) \\ -P - P - \cdots - P \ (-n \text{ 回}) & (n < 0) \end{cases}$$

と定義される.

Algorithm 1: Double and add は秘密鍵 k を $k = (k_{n-1}, ..., k_0)_2$ として表し, スカラー倍算 kP を計算するアルゴリズムである．また, Algorithm 2: Double and add always は, Algorithm 1 を改良した, 各演算ステップの計算量が等しいスカラー倍算アルゴリズムである．さらに, Algorithm 3: Montgomery ladder[1] は Algorithm 1, Algorithm 2 を改良した, y 座標を使用せずに x 座標を計算可能なスカラー倍算アルゴリズムである.

2.3　楕円曲線上の離散対数問題 (ECDLP)

楕円曲線上の 2 点 $P, Q = kP$ からスカラー k を求めることを, ECDLP を解くという．楕円曲線暗号の安全性は, ECDLP の求解の困難性を根拠としている.

表 1　Algorithm 1:Double and add

input: $k = (k_{n-1}, ..., k_0)_2, P \in E(K)$ **output**: kP
1 : $Q \leftarrow \mathcal{O}$ 2 : **for** $i = n - 1$ **down to 0 do** 2.1 : $Q \leftarrow 2Q$ 2.2 : **if** $k_i = 1$ 2.2.1 : $Q \leftarrow Q + P$ 3 : **end for** 4 : **return** Q

表 2　Algorithm 2:Double and add always

input: $k = (k_{n-1}, ..., k_0)_2, P \in E(K)$ **output**: kP
1 : $Q[0] \leftarrow \mathcal{O}$ 2 : **for** $i = n - 1$ **down to 0 do** 2.1 : $Q[0] \leftarrow 2Q$ 2.2 : $Q[1] \leftarrow Q[0] + P$ 2.3 : $Q[0] \leftarrow Q[k_i]$ 3 : **end for** 4 : **return** $Q[0]$

表 3　Algorithm 3: Montgomery ladder

input: $k = (k_{n-1}, ..., k_0)_2, P \in E(K)$ **output**: The x-coordinate of kP
1 : $Q[0] \leftarrow \mathcal{O}, Q[1] \leftarrow P$ 2 : **for** $i = n - 1$ **down to 0 do** 2.1 : $Q[1 - k_i] \leftarrow Q[0] + Q[1]$ 2.2 : $Q[k_i] \leftarrow 2Q[k_i]$ 3 : **end for** 4 : **return** $Q[0]$

3　フォールト攻撃

フォールト攻撃は暗号装置が秘密鍵を用いた演算中に起こる誤りを利用する攻撃である．攻撃者は誤りにより得られる不正な出力値を基に秘密鍵の情報を得る.

3.1　Skipping Attack

Skipping Attack は, Schmidt と Herbst によって提案された [2]．この攻撃は RSA 暗号に対する実用的なフォールト攻撃であるが, 楕円曲線暗号にも拡張できる．スカラー倍算アルゴリズム中の for ループの j 回目の処理をスキップした結果を基にビット毎に秘密鍵 $d = (d_i)$ を取り出す.

$$\hat{y_j} = \begin{cases} \hat{y}_{j-1} & (d_j = 0) \\ x^{2^{j-1}} \hat{y}_{j-1} & (d_j = 1) \end{cases}$$

3.2　ペアリング暗号に対するフォールト攻撃

ペアリング暗号に対するフォールト攻撃は, El Mrabet によって提案された [3]．この攻撃は, Miller algorithm の Miller ループの反復回数を標的としており,

その回数を変更した際の出力を得ることで秘密点 P を見つける.

4 提案手法

4.1 前提

定義体を素体 $K = \mathbb{F}_q$ とし, K 上に定義された楕円曲線を $E : y^2 = x^3 + ax + b$ とする. 点の高速スカラー倍算 Double and add, Double and add always, Montgomery ladder アルゴリズムに対し, それぞれアルゴリズム中のループの反復回数を変更する.

4.2 反復回数の変更

ループ回数は秘密鍵 k のビット数 n によって定まる. リバースエンジニアリングによって反復回数のカウンターに属しているフリップフロップを見つけ, クロック周期をカウントすることで n の値が分かる. kP と n の値を記録しておき, レーザー等を用いて障害を起こし, アルゴリズム中の反復回数を変更する.

4.3 秘密鍵 k の復元

反復回数を変更して得られた出力を集め, 秘密鍵 k を復元する.

1 回ずつ減らす方法 反復回数を 1 回減らすと, k の値は先頭のビットが抜かされ $k = (k_{n-1}, k_{n-2}, \ldots, k_0)_2$ から $k' = (0, k_{n-2}, \ldots, k_0)_2$ へと変化する. もし kP と $k'P$ が同じ点ならば, k_{n-1} は 0 であることが分かり, 異なる点ならば, k_{n-1} は 1 であることが分かる. これを繰り返し, k を 1 ビットずつ復元していく.

2 回ずつ減らす方法 フォールトの挿入回数はより少ない方が好ましいため, 反復回数を 2 回ずつ減らし, 1 回ずつ減らす方法に比べてフォールト挿入回数を減らす. 2 回減らすと, k の値は $k = (k_{n-1}, k_{n-2}, k_{n-3}, \ldots, k_0)_2$ から, $k' = (0, 0, k_{n-3}, \ldots, k_0)_2$ へと変化する. もし kP と $k'P$ が同じ点ならば, $k_{n-1} = 0$ かつ $k_{n-2} = 0$ であることが分かる. もし異なる点ならば, (k_{n-1}, k_{n-2}) は, $(0,1), (1,0), (1,1)$ のいずれかであるので一つずつ調べる.

4.4 防御策

この攻撃手法に対する防御策は, ペアリング暗号に対するフォールト攻撃と同様に, フォールトに耐性のあるカウンターを用いる, 反復をランダムな回数実行する, 中間値の点が元の楕円曲線上にあるかチェックする, などが挙げられる.

5 実験

提案する攻撃アルゴリズムの数値実験を行った. 実験プログラムには Magma を用いた.

5.1 実験条件

以下の条件において, 試行回数に対して攻撃成功となる確率を測定する.

- 各スカラー倍算アルゴリズムの for ループの反復回数を変更する.
- 秘密鍵 k が復元できたとき, 攻撃成功とする.
- 試行回数 m は 100 回とする.
- 使用する曲線は, NIST が推奨する曲線 P-192 と P-256[4] を用いる.

表 4 NIST P-192 Curve(フォールト挿入回数 (n-1) 回)

アルゴリズム	成功率 (%)	実行時間 (秒)
Double and add	100	27.190
Double and add always	100	41.435
Montgomery Ladder	100	36.633

表 5 NIST P-256 Curve(フォールト挿入回数 (n-1) 回)

アルゴリズム	成功率 (%)	実行時間 (秒)
Double and add	100	72.038
Double and add always	100	104.035
Montgomery Ladder	100	93.511

表 6 NIST P-192 Curve(フォールト挿入回数 (n-1)/2 回)

アルゴリズム	成功率 (%)	実行時間 (秒)
Double and add	100	31.824(+4.634)
Double and add always	100	55.395(+13.96)
Montgomery Ladder	100	41.636(+5.003)

表 7 NIST P-256 Curve(フォールト挿入回数 (n-1)/2 回)

アルゴリズム	成功率 (%)	実行時間 (秒)
Double and add	100	82.432(+10.394)
Double and add always	100	142.225(+38.19)
Montgomery Ladder	100	106.566(+13.055)

6 結論

本研究では, 新しいフォールト攻撃手法を提案し, NIST 推奨曲線に対する攻撃の成功確率を示した. 実験結果としては, 既存手法よりも高い成功確率が得られた. これは提案手法の攻撃の強力さを示すとともに, 楕円曲線暗号の実装における, フォールト攻撃への対策の必要性を強調するものである.

謝辞

本研究を進めるにあたり，適切な御指導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.

参考文献

[1] Montgomery, P.L.: *Speeding the Pollard and Elliptic Curve Methods of Factorization.* Mathematics of Computation, Volume 48, pp.243-264, 1987.

[2] Jörn-marc Schmidt and Christoph Herbst. *A Practical Fault Attack on Square and Multiply*, Fault Diagnosis and Tolerance in Cryptography, 2008. FDTC '08. 5th Workshop on, pp. 53-58, IEEE, 2008.

[3] Nadia El Mrabet. *What about vulnerability to a fault attack of the Miller algorithm during an Identity Based Protocol ?*, Advances in Information Security and Assurance, pp. 122-134, Springer Berlin Heidelberg, 2009.

[4] Federal Information Processing Standards Publication FIPS 186-2. Digital Signature Standard (DSS), appendix 6: *Recommended Elliptic Curves for Federal Government Use*, Technical report, NIST, January 27, 2000.