SCIS 2003 The 2003 Symposium on Cryptography and Information Security
Hamamatsu, Japan, Jan.26-29, 2003
The Institute of Electronics, Information and Communication Engineers

楕円曲線暗号の実装攻撃
Attacks on Implementations of Elliptic Curve Cryptsystems

小林 鉄太郎*† 星野 文学* 今井 秀樹†
Tetsutaro KOBAYASHI Fumitaka HOSHINO Hideki IMAI
kotetsu@isl.ntt.co.jp fhoshino@isl.ntt.co.jp imai@iis.u-tokyo.ac.jp

Abstract— 実装環境によって公開鍵暗号を攻撃することができることが明らかとなっている。実装に依存する攻撃法のひとつに、演算時間を測定することによる攻撃法 (Timing Attack) がある。本稿では、楕円曲線暗号系に対する Timing Attack について考察する。

Keywords: Timing attack, Elliptic curve cryptosystem, ECDH

1 はじめに

現実世界で暗号を利用するためには、何らかのソフトウェアやハードウェアなどに実装する必要があり、それらの実装は、多くの場合、暗号処理途中に処理時間や消費電力といった数学的な入出力以外の情報、いわゆるサイドチャネル情報が漏洩する。このようなサイドチャネル情報を用いて暗号を解読する手法のひとつとして 1996 年 P. C. Kocher により提案された Timing Attack [1] がある。また、中国人の剰余定理を用いて RSA 実装に対する攻撃法 [2] なども提案されている。

一方、楕円曲線を用いた方式に関しては、消費電力情報を用いて攻撃する Power Analysis による攻撃法が提案されている [3]。

本稿では、楕円曲線を用いた ECDH などのプロトコルに対して Timing Attack を行う新しい攻撃法を提案する。

2 攻撃対象とするモデル

本論文では、図 1に示す ECDH プロトコルを攻撃対象とする。ただし、ゼロ知識証明を応用したプロトコルでは ECDH に類似したやりとりを行うことがあるため、 ECDH 鍵共有以外のプロトコルの攻撃にも応用できる場合がある。

* NTT 情報流通プラットフォーム研究所, 〒 239-0847 神奈川県横須賀市光の丘 1-1
NTT Laboratories, 1-1 Hikarinooka, Yokosuka-shi, Kanagawa-ken, 239-0847 Japan
† 東京大学生産技術研究所 大学院情報学環, 〒 153-8505 目黒区駒場 4-6-1
Institute of Industrial Science Interfaculty Initiative for Information Studies, University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505 Japan.

共有パラメータ	楕円曲線 E, ベースポイント P $E/\mathrm{GF}(p) : y^2 = x^3 + ax + b$ $P \in E(\mathrm{GF}(p))$
A の秘密鍵	$x_A \in \mathbf{Z}$
B の秘密鍵	$x_B \in \mathbf{Z}$

Step 1: A から B に $Q = x_A P$ を送る。
Step 2: B は $K = x_B Q$ を計算する。
Step 3: B から A に $Q' = x_B P$ を送る。
Step 4: A は $K = x_A Q'$ を計算し、 B に ack を返す。
Step 5: A, B は鍵 K を共有する。

図 1: 攻撃対象のプロトコル (ECDH)

2.1 攻撃シナリオ

本論文において攻撃者及び攻撃対象実装を次のように仮定する。

- 攻撃者を B とし、 B が A の秘密鍵 x_A を求めることが出来るとき、攻撃成功とする。
- B は、最大 q 回このプロトコルを行う。そのたびごとに Step 3 で異なる Q' を A に対して送っても良い。
- A は、 q 回のプロトコル中, 同じ x_A を使いつづける。
- A は、 Q' が実際に E 上の点となっているかどうかを検査しない。
- A は、楕円曲線上のスカラー倍演算法として 2 進演算法を使う。

Step 1: $P_1 = 0$ ならば $P_3 = P_2$ として終了。
Step 2: $P_2 = 0$ ならば $P_3 = P_1$ として終了。
Step 3: $Y_1 + Y_2 = 0$ ならば $P_3 = 0$ として終了。
Step 4: $P_1 \neq P_2$ ならば、以下の演算を行なう。

$$\begin{aligned} R &= (Y_2 - Y1)/(X_2 - X_1) \\ X_3 &= R^2 - X_1 - X_2 \\ Y_3 &= R(X_1 - X_3) - Y_1 \end{aligned}$$

Step 5: $P_1 = P_2$ ならば、以下の演算を行なう。

$$\begin{aligned} R &= (3X_1^2 + a)/(2 * Y_1) \\ X_3 &= R^2 - X_1 - X_2 \\ Y_3 &= R(X_1 - X_3) - Y_1 \end{aligned}$$

図 2: 楕円加算アルゴリズム

- 楕円演算以外にかかる時間は固定とし、本論文では無視する。

3 攻撃方法

3.1 基本アイデア

楕円曲線上の点 $P_3 : (X_3, Y_3) = P_1 : (X_1, Y_1) + P_2 : (X_2, Y_2)$ を求めるアルゴルズムを図 2に示す。楕円曲線のパラメータのうち、b の値を使わないため、E とは別の楕円曲線 $E' : y^2 = x^3 + ax + b', (b \neq b')$ 上の点が入力された場合も正しく演算することができる。

本論文の攻撃の基本戦略は、B が楕円曲線 E 上に存在しない点 Q' を A に対して送り、A が Q' を E 上の点として処理することによって、サイドチャネル情報をもらしてしまうことを利用する。

3.2 攻撃手法

詳細な攻撃手順を示す。

Step 1: B は 適当な整数 w にたいし位数が $2^w | \#E'(\mathrm{GF}(p))$ を満たす $\mathrm{GF}(p)$ 上定義される楕円曲線 $E' : y^2 = x^3 + ax + b'$ を用意する。
Step 2: B は、Step 3 において Q' として O を A に送り、A の演算時間 T_0 を測定する。
Step 3: B は、位数 2 の点 $Q_1 \in E'(\mathrm{GF}(p))$ を求め、Step 3 において Q' として Q_1 を A に送り、A の演算時間 T_1 を測定する。
Step 4: B は、位数 4 の点 $Q_2 \in E'(\mathrm{GF}(p))$ をそれぞれ求め、Step 3 において Q' として Q_2 を A に送り、A の演算時間 T_2 を測定する。

$\vdots$

Step 5: B は、位数 2^w の点 $Q_w \in E'(\mathrm{GF}(p))$ を求め、Step 3 において Q' として Q_w を A に送り、A の演算時間 T_w を測定する。

x_A の二進表現 x_i を以下のように定義する。ただし、$0 \leq x_A < 2^m$ とする。

$$x_A = \sum_{i=0}^{m-1} x_i, \qquad (x_i \in \{0, 1\})$$

A が楕円加算を行うのにかかる時間を以下のように定義する。

$A_{P_1P_2}$: $P_1 + P_2$ を求めるのにかかる時間
D_{P_1} : $2P_1$ を行うのにかかる時間

A が 2 進算法で演算を行うとすると、

$$T_0 = \left(\sum_{i=0}^{m-1} x_i \right) A_{OO} + (m-1) D_O \tag{1}$$

という関係が成り立つ。B は式 (1) から鍵 x_A ハミング重みの情報を知ることができる。

$$\sum_{i=0}^{m-1} x_i = (T_0 - (m-1)D_O)/A_{OO}$$

次に、T_1 を用いて

$$\begin{aligned} T_1 &= \left(\sum_{i=0}^{m-1} x_i \right) A_{OP} + \left(\sum_{i=1}^{m-1} x_i \right) D_P + \left(\sum_{i=1}^{m-1} (1 - x_i) \right) D_O \\ &= (T_0 - (m-1)D_O)/A_{OO}(A_{OP} + D_P - D_O) \\ &\quad + ((m-1)A_{OO} - x_0(D_P - D_O)) \end{aligned} \tag{2}$$

式 (2) から B は x_0 を求める。

次に、

$$\begin{aligned} T_2 &= \left(\sum_{i=0}^{m-1} x_i (x_{i+1} A_{OP} + (1 - x_{i+1} A_{2P,P})) \right) \\ &\quad + \left(\sum_{i=1}^{m-1} x_i x_{i+1} \right) D_{3P} \\ &\quad + \left(\sum_{i=1}^{m-1} (1 - x_i) x_{i+1} \right) D_{2P} \\ &\quad + \left(\sum_{i=1}^{m-1} x_i (1 - x_{i+1}) \right) D_P \\ &\quad + \left(\sum_{i=1}^{m-1} (1 - x_i)(1 - x_{i+1}) \right) D_O \end{aligned}$$

の関係から B は $\sum_{m=0}^{i-1} x_i x_{i+1}$ を求める。

以下同様に、B は $\sum_{m=0}^{i-1} x_i ... x_{i+w}$ を求める。以上より、B は長さ w のすべてのビットパターンについて x_A 中に出現する回数を確定させることができる。オイラーパス探索問題を解くことにより x_A を求める。

4 本解読法に対する対策

本論文の方式は、Timing Attack の一種であるため、従来から知られている乱数を用いて演算時間を隠蔽する方法で防ぐことができる。また、送られてきた点が本当に楕円曲線上にのっているかどうかをチェックしないで演算に

用いてしまう場合に攻撃者がより多くの情報を入手できることが利用できてしまうため、受けった点を用いて演算を行なう前に楕円曲線上の正しい点となっていることを確認することも対策となる。

5 まとめ

サイドチャネル攻撃のひとつである、 Timing Attack の楕円曲線方式への適用を考察した。 ECDH に対する攻撃法を実際に示した。本論文の攻撃法は、 ECDH 以外にも ECDLP ベースのプロトコルに適用できる可能性がある。したがって、 Timing Attack を許すような環境下でこのようなプロトコルを実行する場合は、乱数を用いて時間情報をかく乱するなどの適切な対策をとる必要がある。

参考文献

[1] P. C. Kocher. Timing Attacks on Implementiations of Diffie-Hellman, RSA, DSS,and Other Systems, In *Advances in Cryptology — CRYPTO'96*, LNCS 1109, pp. 104–113. Springer-Verlag, Berlin, Heidelberg, New York, 1996.

[2] W. Schindler. A Timing Attack against RSA with the Chinese Remainder Theorem In *Cryptographic Hardware and Embedded Systems — CHES'2000*, LNCS 1965, pp. 109–124. Springer-Verlag, Berlin, Heidelberg, New York, 2000.

[3] K. Okeya, K. Sakurai. Power Analysis Breaks Elliptic Curve Cryptosystems even Secure against Timing Attack, In *Advances in Cryptology — INDOCRYPTO'00*, LNCS 1977, pp. 178–190. Springer-Verlag, Berlin, Heidelberg, New York, 2000.