Resistance against attacks on practical elliptic curve cryptosystems

Uday Banerjee
Department of Electrical & Computer Engineering,
Oregon State University, Corvallis, Oregon 97331 -USA.
E-mail: banerjee@ece.orst.edu

Abstract— **The security provided by elliptic curve (EC) cryptography is based on the difficulty of the elliptic curve discrete logarithm problem (ECDLP). But of late, a new breed of attacks, namely Differential Power Analysis (DPA) and timing attacks have rendered several cryptosystems, including Elliptic Curve cryptosystems, vulnerable. This paper generalizes side channel attacks to elliptic curve cryptosystems and elucidates past and present techniques used to attack and defend them. In particular, a method of preventing any access to the computations being processed by the system, using both hardware and algorithmic approaches, is presented . This system circumvents the need to use more complicated algorithms to mask the computations being processed and achieves higher performance and greater security.**

I. Introduction

Since the introduction of elliptic curves in cryptography by Miller [1] and Koblitz [2] in 1985, the scope of applications of elliptic curves in cryptography has increased tremendously. Elliptic curves are defined by group structures and the arithmetic used is finite field arithmetic. This makes it possible to translate the discrete logarithm (DL) problem utilized by conventional cryptosystems (eg., RSA) to the elliptic curve discrete logarithm problem (ECDLP). The discrete logarithm problem is described as: given a prime p, y and x, compute a from the equation $y \equiv x^a \bmod p$. The elliptic curve version of the DL problem, ie., the elliptic curve discrete logarithm problem is as follows: given an elliptic curve E defined over the finite field F_q, a point $P \in \mathrm{E}(F_q)$ of order n, and a point $Q \in \mathrm{E}(F_q)$, determine the integer $l, 0 \leq l \leq n-1$, such that $Q = lP$. Conventional algorithmic attack on elliptic curve cryptosystems attempt to invert the elliptic curve discrete logarithmic problem in sub-exponential time, and very few of these are able toperform at practical speeds. The one attack of most significance yet is the MOV reduction technique, formulated by Menezes, Okamoto and Vanstone in 1991 [3]. This method attempts to reduce the elliptic curve discrete logarithm problem to the discrete logarithm problem in F_{q^k}, for some integer k. But the applications of this technique are limited only to a particular class of curves known as *supersingular* curves ($k \leq 6$). There are several other attacks, but steps can be easily taken to guard againsht them Thus, since there are no known sub-exponential time algorithms for non-supersingular elliptic curves, much smaller key lengths can be used. A generally agreed upon number is about 160 bits. In this paper, we concentrate on a new class of attacks, namely power attacks, on elliptic curve implementations of smart-cards. The Differential Power attacks, first described by Kocher *et al.* in [4] is a powerful tool that essentially derives secret information by monitoring the power consumption of devices . This paper is organized as follows. We take a brief look at elliptic curve operations in section 2, section 3 shows how they are susceptible to power attacks, section 4 describes a key recovery scheme for an elliptic curve cryptosystem using power attacks, and finally, we suggest a solution for the described attack.

II. Elliptic Curve Operations

An elliptic curve is the curve described by the set of points (x, y) satisfying the equation:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \qquad (1)$$

where

$$a_i \in F_q$$

For practical elliptic curve cryptography, we concern ourselves with a restricted form of the above equation, which is defined over a finite field. Described below is the elliptic group mod p

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

together with a point at infinity O, where a, b satisfy the relation

$$4a^3 + 27b^2 (mod\ p) \neq 0;\ a, b < p$$

If

$$P = (x_1, y_1)\ and\ Q = (x_2, y_2)\ with P \neq -Q$$

then

$$P + Q = (x_3, y_3)$$

where

$$x_3 \equiv \lambda^2 - x_1 - x_2 (mod\ p)$$

$$y_3 \equiv \lambda(x_1 - x_3) - y_1 (mod\ p)$$

The value of λ is given by:

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } P \neq Q \\ \frac{3x_1^2 + a}{2y_1} & \text{if } P = Q \end{cases}$$

It should be noted that multiplication ($Q = dP$) is defined as repeated addition.

III. Differential Power Analysis (DPA)

The Differential Power Analysis technique has risen to fame in recent times for its unconventional and straightforward approach to compromising secret information stored in tamperproof devices such as smartcards. This section will give the reader an insight into this ingenious technique.

DPA analysis makes use of power consumption measurements made during the operation of a system to find out if a speculated key value is correct or not. The technique makes use of a function known as the DPA selection function, which we will denote as D. The attacker records and observes the encryption process for several samples, and compiles a differential trace for them by finding the difference between the average of the traces for which the function D equals zero and the average of the traces for which Z equals one. What is inferred from this differential trace is that, if the speculated key is incorrect, then this differential value will tend to zero as the number of samples tends to infinity. Also, if the speculated key is correct, the DPA selection function D will be *correlated* to the bit that was computed by the device. Thus statistical correlation is used to determine the secret information stored. This approach virtually eliminates the need to cryptanalyze the algorithm. [4]

IV. Susceptibility of elliptic curve cryptosystems to DPA

Since we are discussing the DPA attack on elliptic curve cryptosystems, let us assume that the given cryptosystem is resistant to simple power analysis (SPA), and thus we use the double and add resistant against SPA algorithm.[5] This algorithm is illustrated below:

Algorithm:

input P

$$Q[0] \leftarrow P$$

for i from $l-2$ to 0 do

$$Q[0] \leftarrow 2Q[0]$$

$$Q[1] \leftarrow Q[0] + P$$

$$Q[0] \leftarrow Q[d_i]$$

output Q[0]

We assume here that the algorithm executes in constant time, else it might be vulnerable to timing and simple power attacks. Upon analysis of the above algorithm, we can see that at step k, the partial result Q is dependent only on the first bits $(d_{l-1}, \ldots, d_k)$ of the exponent d. We could perform several sets of computations, for the cases when we assume the bits to be represented in binary, two's complement, etc. This will further our chances of narrowing down the key search. Assuming that we do know the type of representation of points in memory during processing, we choose a particular bit of this representation. When this point Q is processed, power consumption will be correlated to this specific bit of Q, but also, no correlation will be observed with a point that is not computed by the card. To make this clearer, we must first understand that the algorithm presented earlier proceeds in a most significant digit first fashion. Thus, if we want to guess the second most significant digit d_{l-2} of the exponent, we compute the correlation between power consumption and any specific bit of the binary representation of $4P$, ie., 100. If $d_{l-2} = 0$, we know that $4P$ is calculated once during the execution of the above algorithm and thus power consumption is correlated with any specific bit of the binary representaion of $4P$. If $d_{l-2} = 1$, we know that $4P$ is never calculated during execution, and thus the power consumption will not correlate with $4P$. Thus d_{l-2} is recovered. Subsequent bits of the exponent d can be recovered in a similar fashion.

V. Recovery of the private exponent d in $Q = dP$

Assume the algorithm is executed r times with several values of P to compute the corresponding values of Q $(Q_1, \ldots, Q_r)$. Let $C_i(t)$ be the i-th iteration's $(1 \leq i \leq r)$ power consumption and s_i be any specific bit in the binary representation of $4P_i$ for $1 \leq i \leq r$. The correlation function $c(t)$ between $C_i(t)$ and s_i is given as:

$$c(t) = < C_i(t) >_{i=1,2,\ldots,k:s_i=1} - < C_i(t) >_{i=1,2,\ldots,k:s_i=0} \quad (2)$$

Assume that the points $4P_i$ are computed at a time $t = t_1$. The power consumption $C_i(t_1)$ will then be correlated with the specific bit s_i of the binary representation of $4P_i$. The average power consumed for points where the specific bit $s_i = 1$ will differ from the consumption where the specific bit $s_i = 0$. The correlation function $c(t)$ will have a maxima at $t = t_1$, thus telling us that the points $4P_i$ have been computed. If the points $4P_i$ have not been computed, no maxima will be observed in the correlation $c(t)$. This information leads to the deducing of the bits of the private exponent d by examination of the power correlation characteristic.

The above described attack can be applied to various elliptic curve public key protocols including conventional encryption and key exchange, thus serious countermeasures must be taken in order to prevent these kinds of attacks.

The following graphs show the correlation between the power consumption and the computation of specified points.

Figure 1: Correlation function $c(t)$ between the consumption function $C_i(t)$ and the points $4P_i$. the figure clearly shows a spike indicating that the point $4P_i$ was computed.

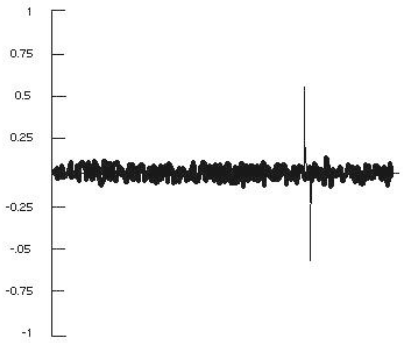
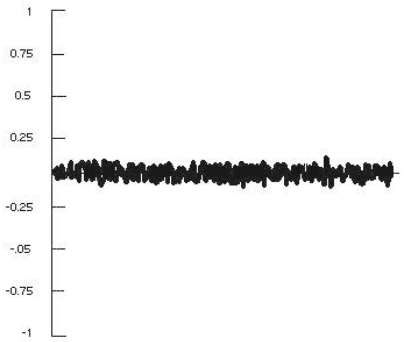


Figure 2: Correlation function $c(t)$ between the consumption function $C_i(t)$ and the points $4P_i$. The figure does not show a spike indicating that the point $4P_i$ is not computed.



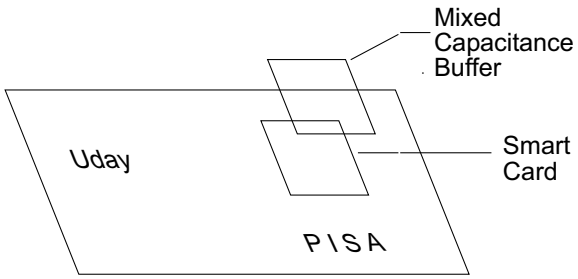
VI. Secure elliptic curve cryptosystems

Since the power attacks on smartcards happen to deal with externally observing the power signatures of the computations, a more pragmatic approach to countering this problem than tweaking algorithms would be to physically protest the smartcard from unknowingly leaking information.

To this extent, with the advancements in capacitive buffering technology, we have been able to come up with a simple and unique solution to the power attack problem. We propose an addition to the smartcard architecture in the form of a yet to be patented Mixed Capacitance Buffer (MCB) chip. This chip efficiently acts as a buffer between the power source and the chip itself. Its working is simple: micro sensors on board this chip track electromagnetic radiation (using the on-board capacitance) in its immediate surroundings as well as any residual radiation from its last operation and produce what are known as "chaons". These chaons are randomly produced particles, and are thought to be a new form of positrons. They essentially interact with the power source and obfuscate any power information about the actual working of the smartcard.

A figure showing an actual implementation of the MCB chip on a smartcard is depicted below.

Figure 3: Proposed addition to existing smartcards - The Mixed Capacitance Buffer chip.



Thus the problem of DPA attacks on elliptic curve implementations of smartcards can be subverted in a most cost efficient way.

The security of elliptic curve cryptosystems for a given key length is supposedly much more than conventional algorithms like RSA, etc. A set of tables are presented below to show the reader the comparison between the computational effort for Cryptanalysis of Elliptic Curve Cryptograpy when compared to RSA. [6]

Key Size	MIPS-Years
150	3.8×10^8
205	7.1×10^{18}
234	1.6×10^{28}

Table 1: Elliptic curve logarithms using the Pollard Rho Method

Key Size	MIPS-Years
512	3×10^4
768	2×10^8
1024	3×10^{11}
1280	1×10^{14}
1536	3×10^{16}
2048	3×10^{20}

Table 2: Integer Factorization using the General Number Field Sieve

Elliptic curve cryptosystems seem to be the standard for the future and with more research, hopefully, will reach the status that AES enjoys today.

References

[1] V. S. Miller, "Use of elliptic curves in cryptography," *Proceedings of Crypto 85*, vol. Lecture Notes in Computer Science (LNCS), no. 218, pp. 417–426, 1986.

[2] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, Jan. 1987.

[3] T. Okamoto A. Menezes and S. Vanstone, "Reducing elliptic curve logarithms in a finite field," *IEEE Trans. Inf.Theory*, , no. 5, pp. 1639–1646, 1993.

[4] Joshua Jaffe Paul Kocher and Benjamin Jun, "Differential power analysis," *White Paper*, pp. 1–10, http://www.cryptography.com.

[5] Jean-Sebastien Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," *Cryptographic Hardware and Embedded Systems (CHES)*, vol. Lecture Notes in Computer Science (LNCS), no. 1717, pp. 292–302, 1999.

[6] William Stallings, *Cryptography and Network Security*, Pearson Education Asia, 1999.