Fault Attack and Countermeasures on Pairing Based Cryptography

Santosh Ghosh, Debdeep Mukhopadhyay, and Dipanwita Roy Chowdhury
(Corresponding author: Santosh Ghosh)

Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur
West Bengal 721302, India (Email: {santosh, debdeep, drc}@cse.iitkgp.ernet.in)

(Received July 16, 2009; revised and accepted Oct. 10, 2009 & Feb. 17, 2010)

Abstract

Bilinear pairing is a new and increasingly popular way of constructing cryptographic protocols. This has resulted in the development of Identity Based Encryption (IBE) that is ideally used in identity aware devices. The security of such devices using pairing algorithms against side-channel and fault attack has not been studied extensively. This paper examines the security of existing countermeasures and show their weakness against fault attacks. Subsequently, it proposes a new countermeasure that prevents such kind of attacks. The paper also discusses for the first time fault attacks and countermeasures against bilinear pairing in Edwards coordinates.

Keywords: Edwards coordinates, fault attack, pairing-based cryptography

1 Introduction

Bilinear pairing has attained utmost importance in the field of public key cryptography due to its wide application area. An area of research in this regard has been developed that is known as pairing based cryptography [8, 9, 12, 17, 20]. Pairing based cryptography is well-suited for developing identity based cryptographic schemes [1, 10, 24, 32]. It is mainly used in identity aware and ubiquitous computing devices. In the last decade, an increasingly popular form of attack known as side-channel analyzes [21, 22], which exploits the weakness in implementations, have developed. A related attack method, known as fault attacks have evolved at the same time. Both of them are directly applied on implementations. Fault attack is based on the accidental or intentional introduction of fault in the computations. It exploits the leakage of information through the faulty outputs of the cryptographic device. Boneh *et al.* showed way back in [7] that cryptographic algorithms can be attacked when they output faulty computation. These theoretical findings were applied on both symmetric ciphers [27, 29, 30, 31] and asymmetric ciphers [6] by several researchers.

Earlier implementation techniques for computing the Tate pairing such as Barreto, Kim, Lynn, and Scott (BKLS) algorithm [3] are effectively realized as point multiplication with a fixed multiplier and some auxiliary operations. Thus the security of pairing computations against side-channel and fault attacks were not considered as a new problem area distinct from elliptic curve cryptography. But, the algorithms for Tate pairing by Duursma and Lee [15], and their modification by Kwon [23] are not based on point multiplication algorithm. Thus, the side-channel and fault attacks become new problems in pairing computation. Fault injection attacks on the above pairing algorithms have been explicitly studied by Page and Vercauteren in [28]. The attack exploits the effect of fault at a specific register, which stores the number of iterations of the pairing computations. The paper [28] have been also proposed some countermeasures for resisting fault attacks on respective pairing algorithms.

It is observed that the countermeasures that are proposed in [28] are based on the measuring techniques of elliptic curve scalar multiplication against side-channel and fault attacks. This paper analyzes the security of the existing countermeasures against fault attacks of pairing algorithms. It shows that the countermeasure against side-channel and fault attacks on elliptic curve scalar multiplication are insufficient for protecting the secrets of pairing computation. It also presents a new countermeasure against such kind of fault injection attacks.

Recently, a new representation of the addition law on elliptic curves, introduced by Edwards [16], leads to extremely efficient elliptic curve group operations [5]. Pairing computation in Edwards coordinates are proposed in [13] and [19]. This paper analyzes the security of the pairing computation that is proposed in [19] against fault injection attack. It finds out a weakness of the pairing computation by Miller's algorithm in Edwards coordinates in presence of fault. The paper also proposes a suitable counter measuring technique.

The paper is organized by first describing the pairing and fault injection attack in Section 2. It investigates the pitfalls of existing countermeasures in Section 3. Section 4 presents one new countermeasure for pairing computation against fault attack. It analyzes the security of pairing computation in Edwards coordinates against fault attack in Section 5. Finally, Section 6 concludes the paper.

2 Background

The pioneer work in the field of pairing based encryption is proposed by Boneh and Franklin [8]. The identity based encryption (IBE) scheme proposed in [8] uses the pairing computation as one of the major operations in encryption as well as decryption procedures. The security of the scheme is based on the difficulty to solve well known Bilinear Diffie-Hellman problem. A very good survey on pairing based cryptographic schemes are given by Dutta et al. [14]. This section gives a brief overview of Tate pairing computation and some of the security issues against fault attack on pairing algorithms. Subsequently, it describes the pairing computation in Edwards coordinates [19].

2.1 Tate Pairing

The name bilinear pairing indicates that it takes a pair of vectors as input and returns a number. It performs a linear transformation on each of its input variables. For example, the dot product of vectors is a bilinear pairing. Similarly, for cryptographic application the bilinear pairing (or pairing) operations are defined on elliptic or hyper-elliptic curves. Pairing is a mapping $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$, where $\mathbb{G}_1$ is a curve group on some field $\mathbb{F}_q$, $\mathbb{G}_2$ is another curve group on the lowest extension field $\mathbb{F}_{q^k}$, and $\mathbb{G}_3$ is a subgroup of the multiplicative group of $\mathbb{F}_{q^k}$.

Let, a large odd prime l divides the order of the curve group ($\#E(\mathbb{F}_q)$). Let, the point P be a l-torsion point for a large prime $l|\#E(\mathbb{F}_q)$. Here k is the corresponding embedding degree, often referred to as security multiplier in pairing computation. It is the smallest positive integer such that l divides $q^k - 1$. Then the Tate pairing of order l is a map

$$e_l : E(\mathbb{F}_q)[l] \times E(\mathbb{F}_{q^k})[l] \to \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^l,$$

where $E(\mathbb{F}_q)[l]$ denote the subgroup of $E(\mathbb{F}_q)$ of all points of order dividing l, and similarly for $\mathbb{F}_{q^k}$. The l-Tate pairing on points $P \in E(\mathbb{F}_q)[l], Q \in E(\mathbb{F}_{q^k})[l]$ is given by $e_l(P, Q) = f_{l,P}(D)$. Here $f_{l,P}$ is a function on E whose divisor is equivalent to $l(P) - l(\mathcal{O})$, D is a divisor equivalent to $(Q) - (\mathcal{O})$, whose support is disjoint from the support of $f_{l,P}$. The point $\mathcal{O}$ represents the point at infinity. For more information regarding divisor, we refer the reader to [3, 18]. The formulas for D and $f_{l,P}(D)$ is given in following equations:

$$\begin{aligned} D &= \sum_i a_i P_i \\ f_{l,P}(D) &= \prod_i f_{l,P}(P_i^{a_i}). \end{aligned}$$

And it satisfies following properties:

- Non-degeneracy: For each $P \neq \mathcal{O}$ there exist $Q \in E(\mathbb{F}_{q^k})[l]$ such that $e_l(P, Q) \neq 1$.
- Bilinearity: For any integer n, $e_l([n]P,\ Q) = e_l(P,\ [n]Q) = e_l(P,\ Q)^n$ for all $P \in E(\mathbb{F}_q)[l]$ and $Q \in E(\mathbb{F}_{q^k})[l]$.
- Let $L = hl$. Then $e_l(P,\ Q)^{(q^k-1)/l} = e_L(P,\ Q)^{(q^k-1)/L}$.
- It is efficiently computable.

The value e_l is a representative of an element of the quotient group $\mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^l$. However for cryptographic protocols it is essential to have a unique representative so it is raised to the $((q^k - 1)/l)$-th power, obtaining an l-root of unity. The resulting value is called reduced Tate pairing:

$$E_l(P, Q) = e_l(P, Q)^{(q^k-1)/l}.$$

The point multiplication based algorithm (Algorithm 1) for pairing computation is given by Miller [25]. The algorithm performs doubling for every bit value of l, and it performs addition only if the corresponding bit value of l is 1. Finally it returns the l-Tate pairing. In the algorithm, $l'(Q)$ indicates the divisor of the straight line equation l' connecting two points P_1 and P_2 with respect to point Q. Let the line l' intersects the curve at a third point X. Now $v'(Q)$ is the divisor of the vertical line equation v' through X with respect to Q [18, 19].

Algorithm 1: Miller's Algorithm

Input P an l torsion point $\in E(\mathbb{F}_q)$, $Q \in E(\mathbb{F}_{q^k})$.
Output The Tate pairing $E_l(P, Q)$
Process

```
$i = [\log_2(l)], K \leftarrow P,\ f \leftarrow 1$.
While $i \geq 1$ do
    Compute equations of $l'$ and $v'$ arising in the
    doubling of $K$.
    $K \leftarrow 2K$ and $f \leftarrow f^2 l'(Q)/v'(Q)$
    If the $i$-th bit of $l$ is 1
        Compute equations of $l'$ and $v'$ arising in
        the addition of $K$ and $P$.
        $K \leftarrow P + K$ and $f \leftarrow f l'(Q)/v'(Q)$.
    end
    $i \leftarrow i - 1$.
End While
Return $f^{(q^k-1)/l}$
```

The Tate pairing can only be computed efficiently if the security parameter k is small. Before the work of Miyaji, Nakabayashi and Takano [26], it was assumed that for general curve k was in size of l. Thus, the early curves to be used in pairing based cryptography were super-singular curves, since their security multiplier satisfies $k \leq 6$. Algorithm 2 presents a technique for pairing computation on hyper-elliptic curves [15]. In the algorithm, ρ, σ, and b are known system parameters. The algorithm was further improved by Kown [23] and by Barreto et al. [3].

Algorithm 2: Miller's Algorithm

```
Input   $P = (x_1, y_1),\ Q = (x_2, y_2)$
Output  $f_P(\phi(Q)) \in \mu_l \subset \mathbb{F}^*_{q^6}$
Process

f ← 1
For $i = 1$ to $m$ do
    $x_1 \leftarrow x_1^3$, $y_1 \leftarrow y_1^3$
    $\mu \leftarrow x_1 + x_2 + b$
    $\lambda \leftarrow -y_1 y_2 \sigma - \mu^2$
    $g \leftarrow \lambda - \mu\rho - \rho^2$
    $f \leftarrow f.g$
    $x_2 \leftarrow x_2^{1/3}$, $y_2 \leftarrow y_2^{1/3}$
End For
Return $f^{q^3-1}$
```

2.2 Fault Attack on Tate Pairing

Fault attack on pairing computation tries to exploit erroneous results that are produced by the device in presence of some transient fault at loop bound m [15] of the corresponding algorithm. Page and Vercauteren [28] first studied the security of pairing algorithms against fault attack. They have shown that if an adversary can induce proper transient fault at loop bound m of Duursma-Lee algorithm then the secret point $P(x_1, y_1)$ could be revealed easily. The transient fault on m can be induced through glitch attack, or provoking error in memory or register in where m is stored [2].

Let an adversary induce transient faults into the register that holds the value of loop boundary m. It measures the modified loop boundary and corresponding pairing result. Let us consider it replaces the loop boundary m with $m \pm r$ and $m \pm r + 1$ in two instances. The corresponding pairing results are $R_1 = e_{m\pm r}$ and $R_2 = e_{m\pm r+1}$. The ratio of these two pairings gives

$$R = \frac{R_2}{R_1} = \frac{e_{m\pm r+1}}{e_{m\pm r}} = g_{m\pm r+1}^{q^3-1},$$

where

$$g_i = -y_1^{3^i}.y_2\sigma - \mu_i^2 - \mu_i\rho - \rho^2.$$

The value of g_i from $g_i^{q^3-1}$ can be extracted through root finding algorithm and by solving some linear system of equations [28]. Here σ and ρ are field extension parameters known to the attacker. The attacker can extract the value of x_1 and y_1 from above equation. We refer [28] for further analysis and information regarding fault attacks on pairing algorithm.

2.3 Pairing in Edwards Coordinates

Edwards showed in [16] that every elliptic curve defined over an algebraic number field $\mathbb{F}$ is birationally equivalent to a curve over some extension of $\mathbb{F}$ given by the equation:

$$x^2 + y^2 \quad = \quad c^2(1 + x^2y^2).$$

Thereafter Bernstein and Lange [5] showed that the group operations can be performed most efficiently on the elliptic curves defined in the Edwards coordinates. The equation $x^2+y^2 = 1+dx^2y^2$ is called the Edward curve [4]. It was shown in [4] that an Edwards curve E is birationally equivalent to the elliptic curve $E_d : (1/(1-d))v^2 = u^3 + 2((1+d)/(1-d))u^2 + u$ via the rational map:

$$\begin{aligned} \psi \quad &: \quad E_d \rightarrow E \\ & (u, v) \rightarrow \left(\frac{2u}{v}, \frac{u-1}{u+1}\right). \end{aligned}$$

The addition formulas on Edwards curve is given by:

$$(x_1, y_1), (x_2, y_2) \rightarrow \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2}\right).$$

It is shown in [5] that above addition law is complete when d is not a square. This means that it is defined for all pairs of input points on the Edwards curve with no exceptions for doubling operation, neutral element, etc.

The pairing computation in Edwards coordinates and on Twisted Edwards coordinates [4] are defined by Ionica and Joux [19], and by Das and Sarkar [13], respectively. The doubling and mixed addition steps of Miller's algorithm for pairing computation are redefined in Edwards and Twisted Edwards coordinates in these two papers. It is shown that the computation of pairing f in Edwards coordinates is the most efficient than that of Twisted Edwards coordinates. This paper takes the pairing computation that is given in [19] for analyzing security against fault attack.

3 Analysis of Existing Countermeasures

Page and Vercauteren [28] have given two countermeasures against fault attacks on pairing based cryptography. Both of the countermeasures are based on point blinding technique, which is known as a good defence mechanism against side-channel attacks on point multiplication algorithm. But the principle to attack point multiplication and pairing computation are completely different. In point multiplication, the adversary try to find out the bit value of scalar multiplier that is used as a secret parameter in elliptic curve cryptography. Whereas, in pairing computation the objective of the adversary is finding out the x and y coordinates of the secret point P. Thus, it is essential to study the countermeasures of side-channel and fault attacks on point multiplication and pairing computation independently. Here we investigate the ability of the countermeasures that are given in [28] to protect the secret point against fault attack and show that they can be compromised.

In the fault attack as described in Section 2.2 the fault is injected randomly into the loop boundary m. The attacker can easily measure the faulty value of m through timing or power analyzes. The attacker has to ascertain the loop boundary of the algorithm for which it produces the final pairing result. The attacker collects two pairing results R_1 and R_2 for asserted faulty loop boundary $m \pm r$ and $m \pm r + 1$, respectively, and computes the ratio

$$R = \frac{R_2}{R_1} = \frac{e_{m\pm r+1}(P,Q)}{e_{m\pm r}(P,Q)},$$

which is exploited to compute the x and y coordinates of secret point P. Therefore, in the countermeasure it is essential to take care of the point such that attacker could not ascertain loop boundary for which the algorithm produces final result. Unfortunately, the countermeasures that are given in [28] do not guarantee such a protection. Let us analyze those two countermeasures.

3.1 Countermeasure-1: New Point Blinding Technique [28]

The aim of point blinding technique is randomization of input points so that the attacker could not utilize knowledge of the public point in pairing computation. This countermeasure chooses two integers x, y randomly from $\mathbb{Z}_l^*$ such that $xy \equiv 1 \pmod{l}$. The points P and Q in $e(P,Q)$ computation are blinded by computing xP and yQ. The pairing is computed on xP and yQ as $e(xP,yQ)$ since it is known that

$$\begin{aligned} e(P,Q) &= e(xP,yQ) \\ &= e(P,Q)^{xy}. \end{aligned} \quad (1)$$

In both Duursma-Lee and Kwon-BGOS algorithms, the input points are processed and it produces pairing result after m iterations. Now according to the relationship, which is shown in Equation 1, the pairing result on set of points (P, Q) and (xP, yQ) are equal. However the fault attack exploits the final result, which is remain unchanged in current countermeasure. Thus the fault attack that is defined in [28] should hold on Duursma-Lee algorithm even in presence of this countermeasure. The details are explained underneath.

Let us assume that the random fault is injected into the respective register in the device to alter the value of m. The new point blinding technique does not change any internal operations of pairing computation. It only changes the input points in such a way that the final result remains unchanged. Pairing computation is an iterative algorithm. In presence of the above countermeasure, the algorithm performs same sequence of operations iteratively on different data. It is observed that the power consumption profile and execution time are almost same for computing same operations on different data. The adversary can find out the number of iterations the algorithm performs for computing one pairing result by simple power analysis and simple timing analysis. Thus the altered value of m is recovered irrespective of input points. The attacker repeatedly alters the value of m and aims to collect two pairing results R_1 and R_2 such that:

$$\begin{aligned} R_1 &= e_{m\pm r}(x_1P, y_1Q) = e_{m\pm r}(P,Q), \\ R_2 &= e_{m\pm r+1}(x_2P, y_2Q) = e_{m\pm r+1}(P,Q). \end{aligned}$$

The ratio of R_2 and R_1 is nothing but $g_{m\pm r+1}^{q^3-1}$, where

$$g_{m\pm r+1} = -y_P^{3^{m\pm r+1}}.y_Q\sigma - \mu_{m\pm r+1}^2 - \mu_{m\pm r+1}\rho - \rho^2,$$

and $\mu = x_P + x_Q + b$. The point $Q = (x_Q, y_Q)$ is known to the attacker. Thus, the above equation can be reduced to a equation of unknown point $P = (x_P, y_P)$. Along with the above equation the attacker knows the curve equation, which can be used as second equation for solving the x, y coordinates of the secret point $P = (x_P, y_P)$. Hence, the value of P can be computed easily by applying the attacking procedure described in [28].

3.2 Countermeasure-2: Altering Traditional Point Blinding [28]

The pairing computation on points P, Q is performed by

$$e(P,Q) = e(P,Q+X).e(P,X)^{-1},$$

where X is a random point. It is assumed that P is secret and Q is public. The fault attack described in [28] exploits knowledge of the public point Q. This defence mechanism [28] tries to randomize the public point Q using the random point X. Thus, it computes $e(P,Q+X)$ instead of $e(P,Q)$, and eliminates the surplus by multiplying the inverse of $e(P,X)$.

Internal operations of the algorithm remain same with the countermeasure. So, the straight line instructions of the pairing algorithms runs iteratively in same manner without countermeasure. In summary, it only wraps the input and unwraps the output for producing correct pairing result. The attacker can easily alter the value of m, randomly, and know its altered value through timing or simple power analysis. With the same fault attack described in [28], attacker collects two pairing results R_1 and R_2. These results are collected after (say) $m \pm r$ and $m \pm r + 1$ iterations, which means $R_1 = e_{m\pm r}(P,Q)$ and $R_2 = e_{m\pm r+1}(P,Q)$ for some random fault r and $r+1$. The ratio $\frac{R_2}{R_1}$ can be written in terms of $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ as written in Section 3.1. The faulty results can be exploited in the same way as the fault attack described in [28] for finding out the secret point P.

4 Proposed Countermeasure

This section proposes a suitable countermeasure against fault attack on pairing computation. The underlying principle of fault attack on pairing computation is based on the ability of the attacker to change the value of the loop boundary m. The attacker also has the ability to measure the change from timing or power analysis of the computation. The attacker tries to obtain two pairing computations one for $m + r$ and the other for $m + r$, augmented by 1 through fault induction. Hence, our countermeasure ensures that even if there is a fault the attacker cannot correlate the pairing output with number of iterations. The objective is to disable the attacker from ascertaining the ratio R_2/R_1, as mentioned in Section 2.2.

4.1 Blinding Loop Boundary

The proposed countermeasure blinds the loop boundary m as it is the main factor in fault attacks. It protects the loop boundary so that the attacker cannot guess the number of iterations for which the faulty output is produced. It modifies the Duursma-Lee algorithm for protecting secret point in pairing computation against fault attack. The modified algorithm is shown in Algorithm 3. Other pairing computation procedures, like Kown-BGOS algorithm can be modified by same procedure in order to defend it against fault attack.

Algorithm 3: Modified Duursma-Lee Algorithm

```
Input   $P = (x_1, y_1),\ Q = (x_2, y_2)$
Output  $f_P(\phi(Q)) \in \mu_l \subset \mathbb{F}^*_{q^6}$
Process
Choose $r_1 \in_R \mathbb{Z}^*_{q^6}$, and $r_2 \in_R \mathbb{Z},\ r_2 \leq m$
$f_0 \leftarrow r_1,\ f_1 \leftarrow 1$
$m' \leftarrow m + r_2$
For $i = 1$ to $m'$ do
    $x_1 \leftarrow x_1^3,\ y_1 \leftarrow y_1^3$
    $\mu \leftarrow x_1 + x_2 + b$
    $\lambda \leftarrow -y_1 y_2 \sigma - \mu^2$
    $g \leftarrow \lambda - \mu\rho - \rho^2$
    $f_1 \leftarrow f_1.g$
    $j \leftarrow (i == m)$
    $f_0 \leftarrow f_j$
    $x_2 \leftarrow x_2^{1/3},\ y_2 \leftarrow y_2^{1/3}$
End For
Return $f_0^{q^3-1}$
```

Correctness:

Theorem 1. *The modified Duursma-Lee algorithm produce the correct result.*

Proof. The Algorithm 3 is modified from original Duursma-Lee algorithm (Algorithm 2) for resisting it against side-channel and fault attacks. The original algorithm runs for m iterations and produce the final result after m^{th} iteration. In the modified algorithm, the loop boundary m' is random as $m' \leftarrow m + r_2$, $r_2 \in_R \mathbb{Z}$ and $r_2 \leq m$. It runs for a random number of iterations. However, the intermediate pairing result f_1 is restored into f_0 at the m^{th} iteration only. It is not restored for other iterations. At the end of the execution, i.e. after m' iterations f_0 holds the pairing result of m iterations. Hence, the algorithm produces the correct reduced Tate pairing result. □

Security Against Fault Attack:
Security Assumption. The adversary inject random fault into the loop boundary. But the faulty loop boundary value is not known to the adversary.

Theorem 2. *The modified Duursma-Lee algorithm against fault attack proposed in [28].*

Proof. In the fault attack, the adversary is interested in two pairing results, $R_{m' \pm r'}$ and $R_{m' \pm r' + 1}$. We may consider the following two scenarios.

- Inject fault at m': The adversary can change the value of m' to $m' \pm r'$ (with random r') by injecting fault at m'. Thus, our modified Duursma-Lee algorithm runs for $m' \pm r'$ iterations. If the resultant value $m' \pm r' \geq m$ then the algorithm produces result R_m for m iterations else it produces random value $r_1^{q^3-1}$ as a pairing result. So, the adversary cannot collect two such target outputs by injecting random faults at m' register.

- Inject fault at m: The adversary can inject random fault at m register, and alter m to $m \pm r'$. Thus, the algorithm runs for $m \pm r' + r_2$ iterations. But, it produces result $R_{m \pm r'}$ for $m \pm r'$ iterations only, where r_2 and r' both are random. This result can be collected by the adversary. The adversary can also measure the total number of iterations $m \pm r' + r_2$ by timing or power analysis. But, it could not correlate the outputs and corresponding measured iteration numbers, which are actually not correlated. Thus, it could not find out two useful pairing results. Therefore, the fault attack described in [28] could not be mounted on proposed countermeasure.

□

5 Fault Attack on Pairing Computation in Edwards Coordinates

This section attempts to analyze the security of pairing computation in Edwards coordinates that is defined by Ionica and Joux [19] against fault attack. It finds out a weakness of such algorithm in presence of fault and give a suitable countermeasure.

5.1 Attack Procedure

The fault attack defined in [28] will not work on Miller's algorithm, Algorithm 1, in Edwards coordinates due to the complex nature of the iterative operations. For example, the doubling operation [5] on $K = (X_1, Y_1, Z_1)$ gives $2K = (X_3, Y_3, Z_3)$, and the formulas are:

$$\begin{aligned} X_3 &= 2X_1Y_1(2Z_1^2 - (X_1^2 + Y_1^2)), \\ Y_3 &= (X_1^2 + Y_1^2)(Y_1^2 - X_1^2), \\ Z_3 &= (X_1^2 + Y_1^2)(2Z_1^2 - (X_1^2 + Y_1^2)). \end{aligned}$$

Similarly, during addition K is updated by $K + P$, which is even more complex than doubling [5]. The point K is initialized by the secret point $P = (X_0, Y_0, 1)$.

Algorithm 1 is realized as a point multiplication along with some additional field multiplication for computing pairing value f. The value of f in doubling step of the Miller's algorithm in Edwards coordinates [19] can be computed by $f \leftarrow f^2 l_1$, where in case of even embedding degree and $k > 2$, l_1 can be computed by following equations:

$$\begin{aligned} l_1 = {} & 2X_1Y_1(x/y - y/x)(X_1^2 - Y_1^2)(X_1^2 + Y_1^2 - Z_1^2) \\ & - 2(X_1^2 - Y_1^2)^2(X_1^2 + Y_1^2 - Z_1^2) \\ & - dx^2y^2Z_1^2(X_1^2 + Y_1^2)(2Z_1^2 - X_1^2 - Y_1^2) \\ & + (X_1^2 + Y_1^2)(2Z_1^2 - X_1^2 - Y_1^2)(X_1^2 + Y_1^2 - Z_1^2). \end{aligned}$$

The Tate pairing $E_l(P, Q)$ is computed by Miller's algorithm on points P, Q such that P is an l-torsion point on the curve $E(\mathbb{F}_q)$ and $Q \in E(\mathbb{F}_{q^k})$. In order to mount fault attack on Miller's algorithm in Edwards coordinates, we assume that the adversary has ability to inject fault at the register l. We further assume that the adversary can obtain the pairing result $E_l(P, Q)$ for $l = 2$. This may be possible by adopting some powerful fault injection procedure or from a number of trial with the help of timing and simple power analysis [2, 11, 28]. If $l = 2$ then the Miller's algorithm runs for only one iteration and it executes only doubling part of Algorithm 1. In such a scenario the pairing output $f = l_1$ and $K = P$. So, f will be a function of X_0, Y_0, x, y, and d, which can be deduced from the equation of l_1 by replacing X_1 by X_0, Y_1 by Y_0, and Z_1 by 1. We can assume that the value of d (curve parameter) and $Q = (x, y)$ are known to the attacker. Thus, f has been simplified and represented by the following equation:

$$\begin{aligned} f = {} & a_1X_0^6 + a_2Y_0^6 + a_3X_0^5Y_0 + a_4X_0Y_0^5 + a_5X_0^2Y_0^4 \\ & + a_6X_0^4Y_0^2 + a_7X_0Y_0^3 + a_8X_0^3Y_0 + a_9X_0^2Y_0^2 \\ & + a_{10}X_0^4 + a_{11}Y_0^4 + a_{12}X_0^2 + a_{13}Y_0^2, \end{aligned}$$

for constants $a_1, \cdots, a_{13}$. Here $a_1, \cdots, a_{13}$ are constants as they can be expressed interms of known values, x, y, and d. We can linearize the above equation by using a number of variables. The public point Q could be changed for obtaining a number of such equations. Hence, X_0, Y_0 could be solved by solving the set of linear equations.

- **Practical Implication of Above Fault Attack:** Let us assume l is a large prime (say 256 bits long in practice). Then the probability of setting $l = 2$ by random fault injection [2] is very less ($\approx 2^{-256}$ for a 256-bit l). Hence a random fault in register l has vary less probability of success. However, we propose a different strategy.

  The requirement of our fault attack is satisfied by inverting the least-significant-bit of l, $l[1]$, and setting $i = 1$. Note that since l is a odd prime, $l[1]$ is 1. Now, if l is 256 bits long then i is of $\lceil \log_2(256) \rceil = 8$ bits. Hence the probability of setting $i = 1$ by random fault injection is at least 2^{-8}. The algorithm runs for only one iteration as $i = 1$, and it executes only the doubling part as $l[i] = 0$. Thus the probability of success of the attacker is 2^{-9}. Hence we expect that after 512 trials the attacker will be successful at least once.

5.2 Countermeasure

In order to resist the above fault attack it is ensured that the Miller's algorithm does not produce a valid pairing result for $l = 2$, and for the condition that $i = 1$ and $l[1] = 0$. In general, l is a odd prime in l-Tate pairing computation, which means $l[1] = 1$. But for mounting the above fault attack it is essential to alter the value of $l[1]$ from 1 to 0. Thus we suggest modified Miller's algorithm that is shown in Algorithm 4 for defending against fault attack.

Algorithm 4: Fault Attack Resistant Miller's Algorithm

Input P an l torsion point $\in E(\mathbb{F}_q)$, $Q \in E(\mathbb{F}_{q^k})$
Output The Tate pairing $E_l(P, Q)$
Process

```
$i = [\log_2(l)], K \leftarrow P,\ f \leftarrow 1.$
If $l[1] = 0$ then
    Return 0.
End If
While $i \geq 1$ do
    Compute equations of $l'$ and $v'$ arising in the
    doubling of $K$.
    $K \leftarrow 2K$ and $f \leftarrow f^2 l'(Q)/v'(Q)$.
    If the $i$-th bit of $r$ is 1 then
        Compute equations of $l'$ and $v'$ arising in
        the addition of $K$ and $P$.
        $K \leftarrow P + K$ and $f \leftarrow fl'(Q)/v'(Q)$.
    End If
    $i \leftarrow i - 1$.
End While
Return $f^{(q^k-1)/l}$.
```

Correctness:

Theorem 3. *The fault-attack resistant Miller's algorithm produce the correct result for cryptographic pairing computation.*

Proof. The modified Miller's algorithm performs correctly for cryptographic pairing computation. It is automatically aborted if l is even. It returns zero if least significant bit (LSB) of l is zero, i.e., $l[1] = 0$. But, pairing computation for cryptographic applications chooses l as a large odd prime. Thus, the LSB of l is one, i.e., $l[1] = 1$. In this case our proposed modified Miller's algorithm executes exactly same operations with its original form (Algorithm 1). Thus it produces correct pairing value for cryptographic applications. □

Security:

Theorem 4. *The fault-attack resistant Miller's algorithm is secure against fault attack described in Section 5.*

Proof. The fault attack described in Section 5 believes that the attacker has ability to inject fault at particular variables during execution. It injects fault at variables i and l. In order to mount the fault attack in pairing computation in Edwards coordinate it is necessary to sets $i = 1$ and $l[1] = 0$. Let us assume that the adversary has successfully injected the required fault. Now for performing the attack on the pairing computation it is also necessary to get the correct result for faulty values of i and l. But the proposed fault-attack resistant Miller's algorithm does not execute the pairing with above fault. It will simply return zero. Thus the proposed countermeasure is secure against the fault attack described in Section 5. □

6 Conclusion

The paper has described the security issues of pairing algorithms in presence of fault. It has shown that the existing countermeasures, which are based on the point blinding technique, are not sufficient for resisting fault attack on pairing algorithms. It has proposed a new countermeasure that resists such kind of fault attacks. A weakness of Miller's algorithm in Edwards coordinates in presence of fault has been also described in this paper. The paper has proposed a suitable countermeasure against such an attack.

Acknowledgements

The authors would like to thank the anonymous reviewers for their critical suggestions that greatly improved the quality of this paper.

References

[1] A. K. Awasthi and S. Lal, "ID-based Ring Signature and Proxy Ring Signature Schemes from Bilinear Pairings," *International Journal of Network Security,* vol. 4, no. 2, pp. 187-192, Sept. 2007.

[2] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The Sorcerers apprentice guide to fault attacks," *Cryptology ePrint Archive*, Report 2004/10, 2004.

[3] P. S. L. M. Barreto, H. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," *CRYPTO 2002,* LNCS 2442, pp. 354-368, Springer-Verlag, 2002.

[4] D. J. Bernstein, P. Birkner, M. Joye, T. Lange, and C. Peters, "Twisted Edwards curves," *AFRICACRYPT 2008,* LNCS 5023, pp. 389-405, Springer, 2008.

[5] D. J. Bernstein and T. Lange. "Faster addition and doubling on elliptic curves," *ASIACRYPT 2007,* LNCS 4833, pp. 29-30, Springer, 2007.

[6] I. Biehl, B. Meyer, and V. Muller, "Differential fault analysis on elliptic curve cryptosystems," *CRYPTO 2000*, LNCS 1880, pp. 131-146, Springer, 2000.

[7] D. Boneh, R. A. DeMillo, and R. J. Lipton, "On the importance of checking cryptographic protocols for faults," *EUROCRYPT 1997,* LNCS 1233, pp. 37-51, Springer, 1997.

[8] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," *CRYPTO 2001,* LNCS 2139, pp. 213-229, Springer, 2001.

[9] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *ASIACRYPT 2001,* LNCS 2248, pp. 514-532, Springer, 2001.

[10] X. Cheng, L. Guo, and X. Wang, "An identity-based mediated signature scheme from bilinear pairing," *International Journal of Network Security,* vol. 2, no. 1, pp. 29-33, Jan. 2006.

[11] M. Ciet and M. Joye, "Elliptic curve cryptosystems in the presence of permanent and transient faults," *Designs, Codes and Cryptography,* vol. 36, pp. 33-43, 2005.

[12] M. L. Das, A. Saxena, D. B. Phatak, "Proxy signature scheme with effective revocation using bilinear pairings," *International Journal of Network Security,* vol. 4, no. 3, pp. 312-317, Nov. 2007.

[13] M. P. L. Das and P. Sarkar, "Pairing computation on twisted Edwards form elliptic curves," *Pairing 2008,* LNCS 5209, pp. 192-210, Springer, 2008.

[14] R. Dutta, R. Barua, and P. Sarkar, "Pairing-based cryptographic protocols: A survey," *Cryptology ePrint Archive*, Report 2004/064, 2004.

[15] I. Duursma and H. Lee, "Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$," *ASIACRYPT 2003,* LNCS 2894, pp. 111-123, Springer, 2003.

[16] H. M. Edwards, "A normal form for elliptic curves," *Bulletin of the American Mathematical Society*, vol. 44, no. 3, pp. 393-422, 2007.

[17] C. Gu and Y. Zhu, "New efficient searchable encryption schemes from bilinear pairings," *International Journal of Network Security,* vol. 10, no. 1, pp. 25-31, Jan. 2010.

[18] J. Hoffstein, J. Pipher, and J. H. Silverman, *An Introduction to Mathematical Cryptography,* Springer, 2008.

[19] S. Ionica and A. Joux, "Another approach to pairing computation in Edwards coordinates," *INDOCRYPT 2008*, LNCS 5365, pp. 400-413, 2008.

[20] A. Joux, "A one round protocol for tripertite Diffie-Hellman," *Procedings of ANTS 4,* LNCS 1838, pp. 385-394, 2000.

[21] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, RSA, DSS and other systems," *CRYPTO 1996*, LNCS 1109, pp. 104-113, 1996.

[22] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," *CRYPTO 1999,* LNCS 2139, pp. 388-397, Springer, 1999.

[23] S. Kwon, "Efficient tate pairing computation for supersingular elliptic curves over binary fields," *Cryptology ePrint Archive,* Report 2004/303, 2004.

[24] R. Lu and Z. Cao, "Group oriented identity-based deniable authentication protocol from the bilinear pairings," *International Journal of Network Security,* vol. 5, no. 3, pp. 283-287, Nov. 2007.

[25] V. S. Miller, "The weil pairing, and its efficient calculation," *Journal of Cryptology,* vol. 17, no. 4, pp. 235-261, 2004.

[26] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions on elliptic curve traces for FR-reduction," *IEICE Transactions on Fundamentals,* vol. E-84, no. 5, pp. 1234–1243, 2001.

[27] D. Mukhopadhyay, "An improved fault based attack of the advanced encryption standard," *AFRICACRYPT 2009,* LNCS 5580, pp. 421-434, 2009.

[28] D. Page and F. Vercauteren, "A fault attack on pairing-based cryptography," *IEEE Transactions on Computers,* vol. 55, no. 9, pp. 1075-1080, 2006.

[29] S. P. Skorobogatov and R. J. Anderson, "Optical fault induction attacks," *CHES 2002,* LNCS 2523, pp. 2-12, Springer-Verlag, 2002.

[30] J. Takahashi, T. Fukunaga, and K. Yamakoshi, "DFA mechanism on the AES schedule," *Proceedings of 4th International Workshop on Fault Detection and Tolerance in Cryptography,* pp. 62-72, IEEE Computer Society, 2007.

[31] J. Takahashi and T. Fukunaga, "Improved differential fault analysis on CLEFIA," *Proceedings of the 2008 5th Workshop on Fault Diagnosis and Tolerance in Cryptography,* pp 25–34, FDTC, IEEE Computer Society, 2008.

[32] Z. Zhao, "ID-based weak blind signature from bilinear pairings," *International Journal of Network Security,* vol. 7, no. 2, pp. 265-268, 2008.

Santosh Ghosh received his B.Tech degree in Computer Science and Engineering from Haldia Institute of Technology, Haldia, WB, India in 2002. He received his M.S. degree in Computer Science and Engineering from Indian Institute of Technology, Kharagpur, India in 2008. He is currently working towards his PhD degree in the department of Computer Science and Engineering at Indian Institute of Technology Kharagpur. His research interests include cryptography and network security, hardware implementation of cryptosystems, side-channel analysis, fault analysis, VLSI Design and Testing.

Debdeep Mukhopadhyay is presently working as an Assistant Professor in the Computer Science and Engineering Department, Indian Institute of Technology Kharagpur from June 2008. Prior to this he worked as an Assistant Professor in the the Computer Science and Engineering Department, Indian Institute of Technology Madras. Debdeep obtained his B.Tech from the Dept of Electrical Engg, Indian Institute of Technology Kharagpur in 2001. Subsequently he obtained his M.S. Degree in 2004 and PhD in 2007 from the Dept of Computer Sc and Engg, Indian Institute of Technology Kharagpur. He has authored about 10 Journal and 50 Conference papers and has served in the Program Committee and as Reviewers of several International Conferences and Journals. Debdeep has been awarded the Indian Semiconductor Association (ISA) TechnoInventor award for the best PhD Thesis in 2008.

Dipanwita Roychowdhury is a Professor in the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India. She received her B.Tech and M.Tech degrees in Computer Science from University of Kolkata in 1987 and 1989 respectively, and the PhD degree from the Department of Computer Sc and Engg, Indian Institute of Technology, Kharagpur in 1994. Her current research interests are in the field of Cryptography, Error Correcting Code, Cellular Automata, and VLSI Design and Testing. She has published more than 125 technical papers in International Journals and Conferences. Dr. Roy Chowdhury has supervised 8 PhD and 6 MS thesis and she is the Principal Investigator of several R&D projects. She is the recipient of INSA Young Scientist Award and Associate of Indian Academy of Science.