An algorithm to select a secure twisted elliptic curve in cryptography

Dinh Tien Thanh, Nguyen Quoc Toan, Nguyen Van Son, Nguyen Van Duan

***Abstract*—Fault attack is a powerful adjacency channel attack technique to break cryptographic schemes. On elliptic curve cryptography (ECC), fault attacks can be divided into three types: safe-error attacks, weak-curve-based attacks, and differential fault attacks. In the paper [1], the author has presented the fault attack on the elliptic curve cryptosystem based on the quadratic twist curve and Proposed criteria to resist elliptic fault attack on the elliptic curve. In this paper, we propose an algorithm to choose a twist secure elliptic curve and evaluate the paths published in cryptographic standards around the world.**

***Tóm tắt*— Tấn công gây lỗi là một kỹ thuật tấn công kênh kề mạnh nhằm phá vỡ các lược đồ mật mã. Tấn công gây lỗi lên mật mã đường cong elliptic (ECC) có thể được chia thành ba loại: tấn công safe-error, tấn công dựa trên đường cong yếu và tấn công gây lỗi vi sai. Trong bài báo [1], nhóm tác giả đã làm tường minh tấn công gây lỗi lên ECC dựa vào đường cong xoắn và đề xuất tiêu chí để chống lại tấn công gây lỗi trên ECC. Bài báo này nhóm tác giả đề xuất thuật toán lựa chọn đường elliptic an toàn xoắn và đánh giá an toàn xoắn cho các đường cong elliptic đã công bố trong một số chuẩn mật mã.**

Keywords—fault attacks; twist curve; Montgomery ladder; elliptic curve cryptosystem.

Từ khóa—tấn công fault, đường cong xoắn, thang Montgomery, mật mã đường cong Elliptic Introduction.

The Montgomery scale method is known to be an efficient scalar multiplication algorithm resistant to some adjacency channel attacks. In FDTC 08, Fouque et. [10] described a failure attack based on the twisted curve implemented on the Montgomery scale method. The curve does not use *y*-coordinates in order to countermeasure against the point verification method. Considering the elliptic curve E defined in $\mathbb{F}_p$, a random value $x \in \mathbb{F}_p$, corresponding to the coordinates of a point either on the elliptic curve E or on its twisted curve E' with a probability of approximately ½.

While not working with *y*-coordinates, The Montgomery algorithm works on E elliptic curves like on E' elliptic curves. The goal of the attack is to solve the discrete logarithm problem (DLP) on the twisted curve E' (which is a weak curve), from which it is easy to solve the DLP problem on the original curve E (which is a strong curve).

In the article [1], the authors have clarified the relationship formula between the order of the initial elliptic curve E and its twisted curve E'. And then, the authors presented how to solve the small problem of DLP on the twisted curve E', from which it is easy to get the result of the DLP problem on the original curve E, to get the secret key.

The fault attack model based on the twisted curve E' is as follows:

- The attacker modifies the x coordinate of point P to form point P' such that $P' \in E'$.

- The targeted attack is to implement Montgomery scale scalar multiplication when the y coordinate is not used. Performing scalar multiplication with the point P' gives an error $Q' = dP' \in E'$.

- Compute $d \bmod ord(P')$ by solving the DLP in the group $\langle P' \rangle$.

There are two types of twisted curve-based attacks. The basic attack without countermeasures is when the adversary has the ability to choose the input point P and implements no end-point verification of the scalar multiplication algorithm. The second attack assumes that the attacker cannot choose P and implements the resulting point verification at the end of the scalar multiplication. In this case, the attacker needs to insert two errors. First, an error is inserted into the *x*-coordinate of the base point P to form the point P'. Then, $P' \in E'$ with probability 1/2. Second, at the end of the

calculation, an error is inserted into the x coordinate of dP just before the point verification. The attacker then passes the point verification step with a probability 1/2. The error output can be obtained and pass the point verification with probability 1/4.

When obtained $Q' = dP'$ over E', if $\text{ord}_{E'}(P')$ smooth or small, an attacker can solve the discrete logarithm problem using algorithms such as Pohlig-Hellman analysis, Shank's baby-step-giant-step method, Pollar$-\rho$ method to compute $d \bmod \text{ord}(P')$. Repeat the process with enough different points, and using the Chinese Remainder Theorem, we get the value $d \bmod \text{ord}(E')$ where the time complexity is the square root of the largest factor of the order of the twisted curve. One of the criteria for dealing with a fault attack is to select an E curve that ensures that it is difficult to solve DLP problem on its twisted curve E' to be able to convert to DLP on the original curve E.

The rest of paper is organized: we first present some basic algorithms in Section II. Section III proposes an algorithm to select a secure twisted elliptic curve. Elliptic curve will be evaluate to meet secure twisted curve criteria in Section IV. We conclude the paper with a conclusion in section V.

I. SOME FOUNDATION ALGORITHMS

A. *Elliptic Curve Discrete Logarithm Problem (ECDLP)*

Let E be an elliptic curve over a finite field $\mathbb{F}_p$ and point $G \in E$ (order n) produces a cyclic group <G>. Let point $P \in < G >$, find a positive integer x such as $P = xG$.

To solve ECDLP, it is necessary to check all values of $x \in [2, n-2]$. If G is chosen carefully with very large n the solution of the ECDLP is considered infeasible. Solving the ECDLP is considered to be more difficult than solving the DLP on the finite field. Currently, there is no effective algorithm to solve this problem in polynomial time. The current best algorithm for solving ECDLP is Pollard's rho algorithm.

Pollard's rho method [17]

Suppose G be a finite group with order N. Divide G to s discrete subset $S_1, S_2, ..., S_s$ of approximately the same size. A good choice for s is 20. Pick 2s random integers a_i, $b_i (\text{mod } N)$. Set: $M_i = a_i P + b_i Q$.

Definition $f(g) = g + M_i : g \in S_i, \overline{i = 1...s}.$

Finally, pick randomly integers a_0, b_0 and set $P_0 = a_o P + b_0 Q$ be the starting point for the random run. While calulating the points P_j, We also record how these points P and Q are represented. If $P_j = u_j P + v_j Q$ and $P_{j+1} = P_j + M_i$, then $P_{j+1} = (u_j + a_i)P + (v_j + b_i)Q$, so $(u_{j+1}, v_{j+1}) = (u_j, v_j) + (a_i, b_i)$.

If found $P_{j_0} = P_{i_0}$ then we have: $u_{j_0} P + v_{j_0} Q = u_{i_0} P + v_{i_0} Q.$ Therefore $(u_{i_0} - u_{j_0})P = (v_{j_0} - v_{i_0})Q.$ If $\gcd(v_{j_0} - v_{i_0}, N) = d$, we have:

$$k \equiv (v_{j_0} - v_{i_0})^{-1}(u_{i_0} - u_{j_0}) \quad (\text{mod } \frac{N}{d})$$

This gives d a way to choose the value of k. Normally d will be small, so we can try all possible values until we find $Q = kP$. In cryptographic applications, N must usually be prime, then $d = 1$ or $d = N$. If $d = N$, e get a trivial relation (the coefficients of both P and Q are multiples of N), then we have to start over. If $d = 1$, k will be found.

The effectiveness of Pollard's rho method:

Pollard's method is applicable to any finite group of class N with negligible memory requirements. Average computation time of the Pollard – rho algorithm is $\sqrt{\pi N/2}$. There are several ways to improve the computation time of this algorithm, making the average computation time of the algorithm much lower. J. Bernstein et al. [6] implemented Pollard's rho algorithm using inverse mapping with the average time of the algorithm being $\sqrt{\pi N/4}$.

ECDLP can to be unbreakable if the degree of the base point *N* must be large enough that the ECDLP is secure against the Pollard' rho method.

Currently the most powerful supercomputer in the world Fugaku in Japan has 442,010 petaflop (1 petaflop is equivalent to 10^{15} calculations/second) is equivalent to performing approximately 2^{84} calculation in one year. Safecurve [14] request *N* such as $\sqrt{\pi N/4} > 2^{100}$, to break the ECDLP it is necessary to have minimum of 2^{100} computations, and ECDLP is secure in the near future.

B. Elliptic curve conversion algorithm

Let $p > 2$ be a large prime and $\mathbb{F}_p$ is a finite field of *p* elements. Edwards twisted curve $E_{a,d}$ over $\mathbb{F}_p$ is defined by the equation:

$$E_{a,d} : ax^2 + y^2 = 1 + dx^2y^2$$

in which $a,\ d \in \mathbb{F}_p$ and $a(a-d) \neq 0$ [13].

Let $p > 2$ be a large prime and $\mathbb{F}_p$ is a finite field of *p* elements. Montogomery curve $M_{A,B}$ over $\mathbb{F}_p$ is defined by the equation:

$$M_{A,B} :\ By^2 = x^3 + Ax^2 + x$$

in which $A \in \mathbb{F}_p \setminus \{-2,\ 2\}$ and $B \in \mathbb{F}_p \setminus \{0\}$ [13].

Let $p > 3$ be a large prime and $\mathbb{F}_p$ is a finite field of *p* elements, elliptic curve in short Weierstrass form over $\mathbb{F}_p$ is defined by the equation:

$$\mathrm{W}_{a,b} : y^2 = x^3 + a_{\mathrm{w}}x + b_{\mathrm{W}}$$

with $a,\ b \in \mathbb{F}_p$ and $4a^3 + 27b^2 \neq 0$ [13].

Mapping between twisted Edwards curves and Montgomery curves

Every twisted Edwards curve $E_{a,d}$ over $\mathbb{F}_p$ is birationally equivalent over to a Montgomery curve $M_{A,B}$ over $\mathbb{F}_p$ [13] with parameters:

$$A = 2\frac{a+d}{a-d} \text{ và } B = \frac{4}{a-d}$$

Conversely, every Montgomery curve over $\mathbb{F}_p$ is birationally equivalent over $\mathbb{F}_p$ to a twisted Edwards curve [13] with parameters:

$$a = \frac{A+2}{B} \quad \text{and} \quad d = \frac{A-2}{B}$$

Mapping a point (x_M, y_M) over $M_{A,B}$ to the point (x_E, y_E) on $E_{a,d}$ as follows:

$$(x_E, y_E) = \left(\frac{x_M}{y_M}, \frac{x_M - 1}{x_M + 1}\right)$$

Conversely:

$$(x_M,\ y_M) = \left(\frac{1+y_E}{1-y_E},\ \frac{1+y_E}{(1-y_E).x_E}\right)$$

Mapping between Montgomery curves and short Weierstrass

Any elliptic curve can be represented in Weierstrass form. In particular, the elliptic curve is in the form of Montgomery. The Montgomery curve mapping can be done in the following way: Divide $M_{A,B}$ to B^3, then let $x = Bu$ and $y = Bv$, we have:

$$v^2 = u^3 + \frac{A}{B}u^2 + \frac{1}{B^2}u$$

Let $u = t - \frac{A}{3B}$, substituting into the above equation we get the equation:

$$v^2 = \left(t - \frac{A}{3B}\right)^3 + \frac{A}{B}\left(t - \frac{A}{3B}\right)^2 + \frac{1}{B^3}\left(t - \frac{A}{3B}\right)$$

Transforming the above equation, we get the following equation:

$$v^2 = t^3 + \frac{(3-A^2)}{3B^2}t + \frac{2A^3 - 9A}{27B^3}$$

Therefore, to map an elliptic curve in form Montgomery $M_{A,B}$ to short Weierstrass form $\mathrm{W}_{a,b}$, we need to use the formula:

$$a = \frac{3-A^2}{3B^2} \text{ and } b = \frac{2A^3 - 9A}{27B^3}$$

And mapping a point (x_M, y_M) over $M_{A,B}$ to a point (x_W, y_W) over $W_{a,b}$ as follow:

$$(x_W, y_W) = (\frac{x_M}{B} + \frac{A}{3B}, \frac{y_M}{B}).$$

Algorithm 1: Mapping between Montgomery curves and short Weierstrass

Input: $A, B, p, \boldsymbol{G(x_M, y_M)}$

Output: a_w, b_w, p, G_W

1. Compute $a_w = \frac{3-A^2}{3B^2}$
2. Compute $b_w = \frac{2A^3 - 9A}{27B^3}$
3. Compute $x_w = \frac{x_M}{B} + \frac{A}{3B}$
4. Compute $y_w = \frac{y_M}{B}$
5. Set $G_w = (x_w, y_w)$
6. Return (a_w, b_w, p, G_W).

Mapping between twisted Edwards curves and Weierstrass curves

From the above two transformations, we have the formula to map an elliptic curve in Edwards form $E_{a,d}$ to short Weierstrass form $W_{a,b}$ as following:

$$(a_w, b_w) = \begin{pmatrix} -\frac{1}{48}(a^2 + 14ad + d^2), \\ \frac{1}{864}(a+d)(-a^2 + 34ad - d^2) \end{pmatrix}$$

Mapping point (x, y) over curve $E_{a,d}$ to point (x_w, y_w) over curve $W_{a,b}$ as follow:

$$(x_w, y_w) = \left(\frac{5a + ay - 5dy - d}{12 - 12y}, \frac{a + ay - dy - d}{4x - 4xy}\right)$$

Algorithm 2: Mapping between twisted Edwards curves and short Weierstrass

Input: $a, d, p, \boldsymbol{G(x, y)}$

Output: $\boldsymbol{a_w, b_w, p, G_W}$

1. Compute $\boldsymbol{a = -\frac{1}{48}(a^2 + 14ad + d^2)}$
2. Compute $\boldsymbol{d = \frac{1}{864}(a+d)(-a^2 + 34ad - d^2)}$
3. Compute $\boldsymbol{x_w = \frac{5a+ay-5dy-d}{12-12y}}$.
4. Compute $\boldsymbol{y_w = \frac{5a+ay-5dy-d}{12-12y}}$
5. Set $\boldsymbol{G_w = (x_w, y_w)}$
6. Return $(\boldsymbol{a_w, b_w, p, G_W})$.

II. III- AN ALGORITHM TO SELECT A SECURE TWISTED ELLIPTIC CURVE IN CRYPTOGRAPHY

Theorem 1: [Hasse's theorem] [**Theorem 13.28,12**]. Let E be an elliptic curve defined on $\mathbb{F}_p$. We have:

$$\left|E(\mathbb{F}_p)\right| = p + 1 - t$$

In which $|t| \leq 2\sqrt{p}$ and t is called the trace of the Frobenius mapping at p.

We have $p + 1 - 2\sqrt{p} \leq \left|E(\mathbb{F}_p)\right| \leq p + 1 + 2\sqrt{p}$. The algorithm is currently considered to be the most efficient for calculating $\left|E(\mathbb{F}_p)\right|$, that is SEA (Schoof-Elkies-Atkin) algorithm [17.2.2, 12]. This algorithm has a time complexity of $O(\log^6 p)$, and can be reduced $O(\log^4 q)$.

Proposition 1 [1]: Let E be the elliptic curve defined over $\mathbb{F}_p$ and E' the second order twisted curve of E. Then, $\left|E(\mathbb{F}_p)\right| + \#\left|E'(\mathbb{F}_p)\right| = 2p + 2$.

Secure twisted curve criteria: Twisted curve E' must be secure as initial curve E. It means that the degree of twisted curve is a prime or there are at least a prime $q' > 2^{200}$ as in safecurve criteria [14]. In addition, E' can resit MOV attacks.

Most of the cryptographic standards on elliptic curves today such as ANSI X9.62 [2], SEC1 version 2.0 [4], NIST FIPS 800-186 [13], Brainpool [8]… do not mention the security standard for elliptic curves. In addition to providing standards, some standards also provide elliptic curves that they consider secure and

recommend to use such as elliptic curves in ANSI X9.62 [2], NIST [13], Brainpool [8], GOST [11], SECv2 [5], NUMS (Microsoft Nothing Up My Sleeve) [3],… When using these curves, it is not possible to guarantee that it will not be attacked by fault attack on twisted curves. Therefore, before using these curves, it is necessary to verify whether it meets secure twisted curves. Based on proposition 1 and the security criteria to deal with the fault attack based on the above twisted curve, we propose an algorithm to choose a secure twisted elliptic curve as follows:

Algorithm 3: Selecting a secure twisted elliptic curve

Input: Parameters of the curve E over $\mathbb{F}_p$

Output: Return $(\text{ord}(E),\ \ell,\ \text{ord}(E'),\ p',\ \text{YES})$

if the curve satisfies the secure twisted curve criteria.

Return $(\text{ord}(E),\ \ell,\ \text{ord}(E'),\ p',\ \text{NO})$ if the curve do not satisfy the secure twisted curve criteria

1. 1. If E is not a short Weierstrass curve, mapping the E curve to a short Weierstrass curve
2. Compute $\text{ord}(E)$
3. Parse $\text{ord}(E)$ to: $\text{ord}(E) = h\ell$, in which ℓ is a large prime and h is a co-factor
4. If $\ell \le 2^{200}$ the stop.
5. Compute $\text{ord}(E') = 2(p+1) - \text{ord}(E)$
6. Parse $\text{ord}(E')$ to product of prime numbers: $\text{ord}(E') = \prod q'_i$
7. Compute $q' = \max\{q'_i\}$
8. If $q' \le 2^{200}$ then return $(\text{ord}(E),\ \text{ord}(E'),\ p',\ \text{NO})$
9. Return $(\text{ord}(E),\ \text{ord}(E'),\ p',\ \text{YES})$.

The program to select the safe twisted elliptic curve is built on the Visual Stduio C++ 2015 language using the Miracle large number calculation program library [15]. The algorithm for calculating the degree of the curve we use is the SEA algorithm included in the Miracle library. Besides, we use the open source application yafu v1.33 [16] to perform integer parsing.

Based on the proposed algorithm we evaluate the curves of NIST [13], GOST [11], Bainpool [8], SECv2 [5], NUMS [3], and the curves published in [2] . The results of some curves are detailed in the appendix:

Based on the evaluation results, we make some evaluation as follows:

1. The Weierstrass curves given in Brainpool all have orders that are prime but the orders of the twisted curves are all composite, of which only the Brainpool P512t1 curve has the order of the twisted curve having a prime divisor greater than 2^{200}, the remaining curves are all smaller than 2^{200}.

2. NIST published 12 curves (including 7 Weierstrass curves, 2 Montgomery curves, 3 Edwards curves). There are 5 NIST Weierstrass curves with prime level, but only the NIST P-384 curve has the order of twisted curve as prime, the rest of the curves all have the order of twisted curve which is composite. curve NIST P-256, NIST P-521 has the order of twisted curve with greatest prime divisor greater than 2^{200}, the curves NIST P-192, NIST P-224 are less than 2^{200}. Weierstrass W-255, Montgomery NIST Curve25519, NIST Edwards25519 curves have form $8q$ and twisted curve have form $4q'$, Weierstrass curves W-448, Montgomery NIST Curve448, NIST E448, NIST Edwards448 whose order is $4q$ and the order of the twisted curve are both $4q'$, where q and q' are all primes greater than 2^{200}.

3. GOST Weierstrass curves (GOST R 34.10- 2001/2012-256, GOST R 34.10- 2001-512 GOST, id-tc26-gost-3410-12-512-paramSetA, id-tc26-gost-3410-12- 512-paramSetB) whose order is prime but the order of the twisted curve is composite. In the Weierstrass curves of GOST, only the id-tc26-gost-3410-12-512-paramSetB curve has the greatest prime divisor < 2^{200}, the remaining curves all have the greatest prime divisor > 2^{200}.

As for the GOST Edwards curve (id-tc26-gost-3410-2012-256-paramSetA, id-tc26-gost-3410-12-512-paramSetC) there is a order of the twisted curve of the form 4*q*, where *q* is a prime > 2^{200}.

4. In the NUMS standard [13], there are 3 Weierstrass curves (numsp256d1, numsp384d1, numsp512d1) and 3 twisted Edwards curves (numsp256t1, numsp384t1, numsp512t1). The order of Weierstrass curves and twisted curves are both prime. The order of Edwards curves and twisted curves are of the form 4*q*, where *q* is a large prime and greater than 2^{200}.

5. The curves published in [2] include: 3 Montogomey curves (M-221, M-383, M-511), 3 Edwards curves (E-222, E-382, E-521). In which the order of the curves M-221, M-383, M-511 has the form 8*q* and the order of the twisted curve has the form 4*q'*, the order of the curves E-222, E-382, E-521 has the form 4*q* and order of the twisted curve of the form 4*q'*, where *q* and *q'* are all primes greater than 2^{200}.

General Evaluation:

Most of the Weierstrass curves given in the standards (with the exception of the NIST P-384 curve and the NUMS curves) have the order of the twisted curve as a composite, making these curves vulnerable to fault attack on ECC based on twisted curve.

The order of the Montgomery curve and the Edwards curve has the form $h \cdot q$, with h is a small cofactor that is divisible by 4 and q is a large prime. To avoid attacks on subgroups on elliptic curves, cofactors h should be chosen as small as possible. Montgomery and Edwards curves are given in standards of order $h \cdot q$ and order of twisted curve is $h' \cdot q'$. h and h' is chosen depending on the prime number p. All standards are selected $h = h' = 4$ and $p \equiv 3 \bmod 4$ and select $h = 8$ and $h' = 4$ with $p \equiv 1 \bmod 4$ or $p = 5 \bmod 8$. We explain the selection of values h and h' as follows: For the Edward twisted curve if $p \equiv 1 \bmod 4$ then $\left|E(\mathbb{F}_p)\right| + \#\left|E'(\mathbb{F}_p)\right| = 2p + 2 \equiv 4 (\bmod 8)$, therefore either of the curves E or E' will have degrees divisible by 8 while the other will have degrees divisible by 4, it means $h = 4$ and $h' = 8$ or $h = 8$ and $h' = 4$. Same for case $p \equiv 3 \bmod 4$ then $\left|E(\mathbb{F}_p)\right| + \#\left|E'(\mathbb{F}_p)\right| = 2p + 2 \equiv 0 (\bmod 8)$, so select $h = h' = 4$, as for $p \equiv 5 \bmod 8$ then $\left|E(\mathbb{F}_p)\right| + \#\left|E'(\mathbb{F}_p)\right| = 2p + 2 \equiv 12 (\bmod 16)$, thus select $\{h, h'\} = \{4, 8\}$ or $\{h, h'\} = \{8, 4\}$.

III. EVALUATION THE PROBABILITY OF A EDWARDS TWISTED CURVE SATISFIED THE SECURE TWISTED CURVE STANDARD

In this section, we analyze the probability of Edwards twisted curves satisfied the secure twisted curve standard.

We first analyze how many random curves have small cofactors ($h = 1$, $h = 4$, $h = 8$).

We consider curves with large enough primes p (at least 224 bits), these curves with small cofactors as above will satisfy the requirements $\sqrt{\pi l / 4} > 2^{100}$. In other words, this requirement that the curve must have a small cofactor replaces the requirement that the curve meet the secure requirements against Pollard's rho method.

Let $\pi(x)$ be the number of elements in the segment $[2, x]$. The prime number theorem states that $x / \log x$ is a good approximation to $\pi(x)$, in the sense that the limit of the quotient between the two functions $\pi(x)$ and $x / \log x$ if x increases infinitely, equals 1, where $\log x$ is the natural logarithm of x. In 1930, Hoheisel proved that: the number of primes in the segment $I = [x - y, x]$ where $\sqrt{x} < y < x$ such as:

$$\pi(I) = \pi(x) - \pi(x - y) \sim \frac{y}{\log x}$$

Number of integers in I have forms q, $4q$, $8q$; where is a prime number, equal to the number of integers in the intervals $I_1 = [x - y, x]$

, $I_4 = [(x-y)/4, x/4]$ and $I_8 = [(x-y)/8, x/8]$, as follows:

$$\pi(I_1) + \pi(I_4) + \pi(I_8) \sim \frac{y}{\log x} + \frac{y/4}{log(x/4)} + \frac{y/8}{log(x/8)} = \sum_{h\in[1,4,8]} \frac{y/h}{log(x/h)}$$

According to Theorem 1 (Hasse's Theorem) the order of an elliptic curve over a finite field $\mathbb{F}_p$ within $\left[p+1-2\sqrt{p},\ p+1+2\sqrt{p}\right]$, called about Hasse. Let $x = p+1+2\sqrt{p}$ and $y = \sqrt{4p}$ we have the number of integers in the Hasse interval of the form q, $4q$, $8q$, where q is a prime:

$$\sum_{h\in\{1,4,8\}} \left[\frac{4\sqrt{p}/h}{log\left(\left(p+1+2\sqrt{p}\right)/h\right)} \right]$$

Otherwise, the total number of integers in the Hasse interval will be approximately equal to $4\sqrt{p}$. Therefore, the probability of having an integer is of the form q, $4q$, $8q$ in the Hasse interval will be approximately:

$$\sum_{h\in\{1,4,8\}} \frac{1}{h\cdot log\left(\left(p+1+2\sqrt{p}\right)/h\right)} \quad (1)$$

According to proposition 1, the order of an upper elliptic curve $\mathbb{F}_p$ is $\left|E(\mathbb{F}_p)\right| = (p+1)-t$ and the order of its twisted curve is $\left|E'(\mathbb{F}_p)\right| = (p+1)+t$. We cannot expect that if $p+1-t = hq$, where q is a prime then $p+1+t = h'q'$, where q' is also a prime. Due to the order of the twisted curve of an elliptic curve over a finite field $\mathbb{F}_p$ is also in the Hasse interval, so the probability that the order of the twisted curve has the form q', $4q'$, $8q'$, where q is prime will also be approximately:

$$\sum_{h'\in\{1,4,8\}} \frac{1}{h'\cdot log\left(\left(p+1+2\sqrt{p}\right)/h'\right)} \quad (2)$$

Hence the probability that an elliptic curve over $\mathbb{F}_p$ and its twisted curve with small cofactor will be approximately:

$$\sum_{h\in\{1,4,8\}} \frac{1}{h\cdot log\left(\left(p+1+2\sqrt{p}\right)/h\right)} \times \sum_{h'\in\{1,4,8\}} \frac{1}{h'\cdot log\left(\left(p+1+2\sqrt{p}\right)/h'\right)} \quad (3)$$

In the paper [1], there is a probability that a random elliptic curve over $\mathbb{F}_p$ and its twisted curve is secure bounded below by $\frac{1}{2\log^2 p}$ and is bounded above by $\frac{5}{\log^2 p}$. We conduct experiments to obtain data on twisted Edwards curves from which to evaluate the probability that a twisted Edward curve and its twisted curve both have a small cofactor.

We analyze how many random twisted Edwards curves have order 4ℓ,, where is ℓ, a large prime number. To collect data, we perform the following experiment: Let $p = 2^{256} - 189$ and compute the order of the curves. We proceed on values $d \in \{1, -1, 2, -2, ..., 100000\}$ and about 20000 curves are obtained. The algorithm used is as follows:

Alogrithm 4: Create a Edwards twisted curve satisfied the secure twisted curve standard

Input: Prime p with b bit length

Output: $(a, d, \text{ord}(E), \ell, \text{ord}(E'), \ell')$ are parameters of the E curve satisfied the secure twisted curve standard

Step 1: Set $a = 1$.

Step 2: For $d \in \{1, -1, 2, -2, ..., \}$ do

2.1. Compute N, which is order of the curve $x^2 + y^2 = 1 + dx^2y^2$

2.2. Compute $\text{ord}(E)$

2.3. Verify $\ell = \frac{\text{ord}(E)}{4}$ is prime. If not, try d with other value.

2.4. Compute $\text{ord}(E') = 2(p+1) - \text{ord}(E)$

2.5. Verify $\ell' = \frac{\text{ord}(E')}{4}$ is prime. If not, try d with other value.

Step 3: Return (a, d, ord(E), ℓ, ord(E'), ℓ')

In this experiment, we find the probability 0,00236 for $h = 4$ and probability 0,0024 for $h' = 4$. According to formulas (1) and (2), the probability that an elliptic curve on the field $\mathbb{F}_p$, with $p = 2^{256} - 189$ have the same coefficient $h = 4$ or $h' = 4$ will be 0.0014205, this probability is much smaller than the probability that we find over 20000 curves.

In the experiment, we obtained 4 curves with $h = h' = 4$, corresponds to the probability of getting 0.00002, That is, we have to search about 50000 curves to get a curve with $h = h' = 4$. According to the formula (3), the probability that an elliptic curve has $p = 2^{256} - 189$ and $h = h' = 4$ will be approximately: 0.000002017, this probability is much smaller than the probability that we find. According to article [1], the probability that a random elliptic curve is above $\mathbb{F}_p$ will be in the range $[0.0000161412, 0.00016142]$. We see that the probability that we computed when we performed over 20000 curves lies within the range given by article [1].

IV. CONCLUSION

In this paper, we have proposed an algorithm to select a secure twisted elliptic curve. In addition, we conduct evaluation of elliptic curves given in cryptographic standards. We also performed a probability assessment to generate a twisted Edwards curve on $\mathbb{F}_p$ and its twisted curve is secure. The secure twisted curve condition is necessary for elliptic curves to ensure that they are secure against twisted curve-based fault attack, and in the finite field $\mathbb{F}_p$ there are always elliptic curves that satisfy the secure twisted curve condition.

In this paper, we have proposed a secure twisted elliptic curve selection algorithm. At the same time, also conduct evaluation of elliptic curves given in cryptographic standards in the world. We also performed a probabilistic evaluation to generate a Edwards twisted curve over $\mathbb{F}_p$ and its twsited curve is secure. The secure twisted curve condition is necessary for elliptic curves to ensure that it is secure against twisted curve based fault attack and finite-field $\mathbb{F}_p$ always has elliptic curves satisfying the secure twsited curve condition.

REFERENCES

[1] Dinh Quoc Tien, Do Dai Chi, "*Về tấn công gây lỗi trên hệ mật đường cong elliptic dựa vào đường cong xoắn*", Journal Journal of Science and Technology on Information security, No 2. 2016.

[2] Accredited Standards Committee X9. "American National Standard X9.62-2005, Public key cryptography for the financial services industry: the elliptic curve digital signature algorithm (ECDSA)." 2005

[3] B. Black, J. Bos, C. Costello, P. Longa, M. Naehrig, "*Elliptic Curve Cryptography (ECC) Nothing Up My Sleeve (NUMS) Curves and Curve Generation*", https://datatracker.ietf.org/doc/html/draft-black-numscurves-02

[4] Certicom Research. " SEC 1: Elliptic Curve Cryptography, Version 2.0." May 21, 2009.

[5] Certicom Research, "SEC 2: Recommended Elliptic Curve Domain Parameters", 2010.

[6] Daniel J. Bernstein, Tanja Lange, And Peter Schwabe, "On the correct use of the negation map in the Pollard rho method" , https://eprint.iacr.org/2011/003.pdf

[7] Diego F. Aranha, Paulo S. L. M. Barreto, Geovandro C. C. F. Pereira, And Jefferson E. Ricardini, "*A note on high-security general-purpose elliptic curves*", 2013, https://eprint.iacr.org/2013/647.pdf

[8] ECC Brainpool, "ECC Brainpool Standard Curves and Curve Generation", 2010.

[9] Hoheisel, G., Primzahlprobleme in der Analysis.

Sitz. Preuss. Akad. Wiss. 33 (1930), 580—588.

[10] P.-A. Fouque, R. Lercier, D. Réal and F. Valette, "Fault attack on elliptic curve Montgomery ladder implementation, Fault Diagnosis and Tolerance in Cryptography", 2008. FDTC'08. 5th Workshop on, IEEE,, 2008.

[11] RFC 7836, "Guidelines on the Cryptographic Algorithms to Accompany the Usage of Standards GOST R 34.10-2012 and GOST R 34.11-2012", https://datatracker.ietf.org/doc/rfc7836/

[12] Roberto M. Avanzi, Henri Cohen, Christophe Doche, Gerhard Frey, Tanja Lange, Kim Nguyen, Frederik Vercauteren, "*Handbook of Elliptic and Hyperelliptic Curve Cryptography*", 2005.

[13] U.S. Department of Commerce/National Institute of Standards and Technology, "Recommendations for Discrete Logarithm-Based Cryptography: Elliptic Curve Domain Parameters", FIPS-186-6 (draft) 2019.

[14] https://safecurves.cr.yp.to/

[15] Miracl, Miracl Cryptographic SDK, https://github.com/miracl/MIRACL, Accessed on 10/9/2020

[16] B. Buhrow. (2010) yafu. Available: http://sourceforge.net/projects/yafu/

[17] Edlyn Teske (2000), "On Random Walks for Pollard's Rho Method", *Mathematichs of Computation*, Vol. 70, No. 234, pp. 809-825.

ABOUT THE AUTHOR

Dinh Tien Thanh

Workplace: Vietnam Academy of Cryptography Techniques

Email: thanhdt@actvn.edu.vn

Education: Master's degree in Electronic and Communication Techniques

Recent research direction: Cryptography and Information security.

Nguyen Quoc Toan

Workplace: Vietnam Government of Information Security Commission

Email: nqtoan@bcy.gov.vn

Education: Ph.D's degree in Institute of Military Science and Technology

Recent research direction: Public key Cryptography, Security parameters for cryptosystems, Digital signature scheme

Nguyen Van Son

Workplace: Vietnam Academy of Cryptography Techniques

Email: sonnguyenhvktmm@gmail.com

Education: Undergraduate in Electronic and Communication Techniques

Recent research direction: Cryptography and Information security

N**guyen Van Duan**

Workplace: Vietnam Academy of Cryptography Techniques

Email: Duannguyen.kmm@gmail.com

Education: Master's degree in Electronic and Communication Techniques

Recent research direction: Cryptography and Information security