6-2014

Fault attacks on hyperelliptic curve discrete logarithm problem over binary field

Mingqiang WANG

Haiyang XUE
Singapore Management University, haiyangxue@smu.edu.sg

Tao ZHAN

Citation

WANG, Mingqiang; XUE, Haiyang; and ZHAN, Tao. Fault attacks on hyperelliptic curve discrete logarithm problem over binary field. (2014). *SCIENCE CHINA Information Sciences*. 57, (3), 1-17.
Available at: https://ink.library.smu.edu.sg/sis_research/9189

SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2014, Vol. 57 032116:1–032116:17
doi: 10.1007/s11432-013-5048-6

Fault attacks on hyperelliptic curve discrete logarithm problem over binary field

WANG MingQiang[1]*, XUE HaiYang[2] & ZHAN Tao[1]

[1]*School of Mathematics, Shandong University, Jinan* 250100*, China;*
[2]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093*, China*

Received October 5, 2013; accepted December 28, 2013

Abstract In this paper, we present invalid-curve attacks that apply to the hyperelliptic curve scalar multiplication (HECSM) algorithm proposed by Avanzi et al. on the genus 2 hyperelliptic curve over binary field. We observe some new properties of the HECSM. Our attacks are based on these new properties and the observation that the parameters f_0 and f_1 of the hyperelliptic curve equation are not utilized for the HECSM. We show that with different "values" for curve parameters f_0, f_1, there exsit cryptographically weak groups in the Koblitz hyperelliptic curve. Also, we compute the theoretical probability of getting a weak Jacobian group of hyperelliptic curve whose cardinality is an smooth integer.

Keywords hyperelliptic curve, discrete logarithm, binary field, genus, cryptosystem

Citation Wang M Q, Xue H Y, Zhan T. Fault attacks on hyperelliptic curve discrete logarithm problem over binary field. Sci China Inf Sci, 2014, 57: 032116(17), doi: 10.1007/s11432-013-5048-6

1 Introduction

The discrete logarithm problem (DLP) is the keystone for the security of cryptosystems based on elliptic curves and on Jacobian groups of more general algebraic curves. The performance of low-genus hyperelliptic curves has been shown to be competitive with that of elliptic curves(see [1] and reference there in). The outcome is that for implementing cryptographic primitives, curves of genus 3 or higher have clearly practical disadvantages over curves of genus 2 and elliptic curves. In this paper, we are concerned with the security of curves of genus 2 defined over finite filed of characteristic 2.

In 1996 a fault analysis attack was introduced by Boneh et al. [2]. This attack is based on a fault injection in a device performing an RSA [3] or Rabin [4] digital signature. Biehl et al. [5] proposed the first fault-based attack on elliptic curve cryptography (ECC) [6,7]. Their basic idea is to change the input points, elliptic curve parameters, or the base field in order to perform the operations in a weaker group where solving the elliptic curve discrete logarithm problem (ECDLP) is feasible. A basic assumption for this attack is that one of the two parameters of the governing elliptic curve equation is not involved in point operations formulas. The authors [8] find that fault-based attack algorithm on elliptic curve is subexponent. Later, Ciet et al. [9] have shown how to recover the secret key by applying the same principle of invalid curves but using a less restrictive assumption of unknown but fixed faulty input point. Karabina et al. [10] demonstrated that invalid-curve attacks can be successfully mounted on

*Corresponding author (email: wangmingqiang@sdu.edu.cn)

protocols based on genus 2 hyperelliptic curves if the appropriate public-key validation is not performed. Recently, Domiinguze-Oviedo et al. [11] presented fault-based attacks that apply to the Montgomery ladder algorithm on curves defined over the binary field and a computation after a fault may leave the original group and be in a twist of the original elliptic curve. The authors [12] extend this method to hyperelliptic curve. They based their work on the fact that the y-coordinate is not used for the elliptic curve scalar multiplication (ECSM). A number of protections against active fault attacks have been reported in [5,9,13–17].

Anderson and Kuhn reported a practical fault attack [18] by producing faults in instructions rather than in data. Its technique consists of applying a high frequency glitch into the clock or power supply signals. Due to different delays in the processors internal signal paths, this glitch might affect only some signals. Varying the timing and duration of the glitch, the attacker can possibly help execute different wrong instructions which might compromise some sensitive information. Skorobogatov et al. [19] introduced a new way to induce faults into a single bit using a laser beam. This is called optical fault induction attack. They used a low-cost laser to change the contents of any single RAM bit. In this way, according to the principles of differential fault analysis, it is possible to mount an inexpensive attack against many microcontrollers used today in constrained devices. Recently, Kim et al. [20] showed how general propose microcontrollers can be targets of a so-called double-fault attack. Their fault injection method is based on inducing a glitch which makes a transient fault with a voltage spike. These glitches are used to corrupt data transferred between registers and memory or to prevent the execution of the code. They mount successfully this attack on a microcontroller computing the Chinese remainder theorem (CRT) based RSA signature generation algorithm.

The invalid-curve attacks presented by Biehl et al. [5], Ciet et al. [9] and Karabina et al. [10] apply to situations where the above-mentioned parameter is not used for the group formulas. In this paper, we extend the notion of invalid elliptic curves proposed by Domiinguze-Oviedo et al. [11] to genus 2 curves. Our work takes advantage of the fact that the resulted u_h is independent of part parameter of v_g for the hyperelliptic curve scalar multiplication (HECSM), where $[u_h, v_h] = k[u_g, v_g]$. Some numerical examples will be shown in this paper by taking Koblitz hyperelliptic curve over $\mathbb{F}_{2^m}$ as the target curve.

In Section 2, some basic knowledge about hyperelliptic curve and hyperelliptic discrete logarithm problem are described.

In Sections 3, we investigate the hyperelliptic curve scalar multiplication (HECSM) algorithm proposed by Avanzi et al. in [1] on the genus 2 hyperelliptic curve over binary field. We provide some useful properties on which our attack method is based.

There are two ways to represent a divisor in a Jacobian group of a curve. In Section 4, we present two invalid-curve-based attacks on the target algorithm according to the representation of a given divisor.

In Section 5 and Section 6, we first describe our fault attack in detail based on the observations in Section 3. If the validation check of the divisor(points) is omitted in a hyperelliptic curve based cryptographic scheme, our attack model really does work. Next, some numerical examples are provided by taking the Koblitz hyperelliptic curve as the target curve. The implemental results show that the fault attack method is efficient. There is no parallel result in elliptic curve.

In Section 7, we analyze the efficiency of our attack method. Also, we obtain theoretical probability of getting a weak Jacobian group of hyperelliptic curve whose cardinality is a smooth integer. Our experimental results substantiate our claim. As an example, for Koblitz hyperelliptic curve over $\mathbb{F}_{2^{113}}$, the probability of running our attack algorithm to get a invalid hyperelliptic curve of which the cardinality of the Jacobian group is a 2^{75} smooth integer is at least 0.96227. In Section 8, we conclude this paper.

2 Preliminaries

2.1 Hyperelliptic curve

A hyperelliptic curve $\mathcal{H}$ of genus 2 over a finite field $\mathbb{F}_q$ of characteristic 2 can be defined by the following non-singular Weierstrass equation: $\mathcal{H} : y^2 + h(x)y = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$, where $\deg(h) \leqslant 2$.

Let $\mathcal{H}$ be an affine hyperelliptic curve of genus 2 with function field $\mathbb{F}_q(\mathcal{H})$ and coordinate ring $\mathcal{O} = \mathbb{F}_q[\mathcal{H}]$. The group of $\mathcal{O}$-ideal classes is denoted by $\mathrm{Cl}(\mathcal{O})$. The Jacobian $J_{\mathcal{H}}(\mathbb{F}_q)$ of $\mathcal{H}$ over $\mathbb{F}_q$ is the quotient group of the degree zero divisors by the group of principal divisors defined over $\mathbb{F}_q$.

Lemma 1 ([1])**.** We use the notation as above. There exists a surjective homomorphism from $J_{\mathcal{H}}(\mathbb{F}_q)$ to $\mathrm{Cl}(\mathcal{O})$.

Lemma 2 ([1])**.** Let $\mathcal{H}$ be a hyperelliptic curve over finite field $\mathbb{F}_q$ of genus g and let ω denote the nontrivial automorphism of $\mathbb{F}_q(\mathcal{H})$ over $\mathbb{F}_q(x)$ with an $\mathbb{F}_q$-rational Weierstrass point P_∞ lying over the place x_∞ of $\mathbb{F}_q[x]$. Let $\mathcal{O} = \mathbb{F}_q[x,y]/(y^2 + h(x)y - f(x))$.

1) In every nontrivial ideal class c of $\mathrm{Cl}(\mathcal{O})$ there is exactly one ideal $I \subseteq \mathcal{O}$ of degree $t \leqslant g$ with the property: the only prime ideal that could divide both I and $\omega(I)$ are those resulting from Weierstrass points.

2) Let I be as above. Then $I = \mathbb{F}_q[x]u(x) + \mathbb{F}_q[x](v(x) - y)$ with $u(x), v(x) \in \mathbb{F}_q[x]$, u monic of degree t, $\deg(v) < t$ and u divides $v^2 + h(x)v - f(x)$.

3) The polynomials $u(x)$ and $v(x)$ are uniquely determined by I and hence by c. So $[u, v]$ can be used as coordinates for c.

The divisor classes $\overline{D} \in J_{\mathcal{H}}(\mathbb{F}_q)$ are in one-to-one correspondence with the pairs of polynomials (u, v) with $u, v \in \mathbb{F}_q[x]$, $\deg(v) < \deg(u) \leqslant g$, u monic, and $u|(v^2 + hv - f)$. The pair $[u, v]$ is called the Mumford representation of the divisor D.

In this paper, we consider the hyperelliptic curves of genus 2 defined over finite field $\mathbb{F}_q$ of characteristic 2 which is given by the following Weierstrass equation:

$$\mathcal{H} : y^2 + (h_1x + h_0)y = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0. \tag{1}$$

Koblitz hyperelliptic curves C_a of genus 2 defined over the binary field $\mathbb{F}_{2^m}$, $C_a : y^2 + xy = x^5 + ax^2 + 1$ are hyperelliptic curves of form (1). Günther et al. [21] found that there is fast scalar mulitplication algorithm on such curves and $J_{C_1}(\mathbb{F}_{2^m})$ are almost prime, where $m \in \{61, 103, 113\}$.

In the following sections, we use $\mathcal{H}$ to represent a hyperelliptic curve of form (1) defined over binary field $\mathbb{F}_q$ unless otherwise specified. $[u, v]$ denotes the Mumford representation of a divisor in $J_{\mathcal{H}}(\mathbb{F}_q)$. P, Q, R denote points in $\mathcal{H}(\mathbb{F}_q)$. $[u_D, v_D]$ denotes the Mumford representation of a given divisor D in $J_{\mathcal{H}}(\mathbb{F}_q)$, where $u_D = x^2 + u_{D1}x + u_{D0}$, $v_D = v_{D1}x + v_{D0}$. If the divisor D can be represented by $D =: \langle R_1\rangle + \langle R_2\rangle - 2\langle\infty\rangle$, by the property of Mumford representation of a divisor, we have $u_D = (x - x_{R_1})(x - x_{R_2})$, $v_D = \frac{y_{R_1} - y_{R_2}}{x_{R_1} - x_{R_2}}x - \frac{x_{R_1}y_{R_2} - x_{R_2}y_{R_1}}{x_{R_1} - x_{R_2}}$, where $R_i = (x_{R_i}, y_{R_i})$, $i = 1, 2$.

2.2 Hyperelliptic curve discrete logarithm problem

Let $\mathcal{H}$ be a hyperelliptic curve of genus 2 defined over a finite field $\mathbb{F}_q$ of characteristic 2, and $g \in J_{\mathcal{H}}(\mathbb{F}_q)$. The discrete logarithm problem is: given $h \in \langle g\rangle$, find an integer k such that $h = [k]g$.

If the order of the divisor g contains only small prime factors, then it is possible to use the Silver-Pohlig-Hellman algorithm [22] to solve the DLP as presented in Algorithm 1. Let n be the order of the base point g with the prime factorization $n = \prod_{i=0}^{j-1} p_i^{e_i}$, where $p_i < p_{i+1}$.

Without loss of generality, we assume that the order of the base point g for which we want to solve the DLP is a large prime number.

3 Arithmetic of hyperelliptic curve of form (1)

Let $[u_i, v_i]$, $i = 1, 2, 3$ be the Mumford representation of divisors in $J_{\mathcal{H}}(\mathbb{F}_q)$. If $\deg(u_i) = 2$, define $u_i = x^2 + u_{i1}x + u_{i0}$, $v_i = v_{i1}x + v_{i0}$, if $\deg(u_i) = 1$, define $u_i = x + u_{i0}$, $v_i = v_{i0}$.

We will use the affine formulae over binary fields for the group law as described in [1,23,24], and refer to these formulae as F_{2a} (see Appendix A) throughout the paper. Karabina et al. [10] claimed that the output of the formulae F_{2a} is independent of f_1 and f_0. However, they did not give the proof. Our experiment results show that this claim is right. For completeness, we prove the following result.

Algorithm 1 Silver-Pohlig-Hellmans algorithm for solving the DLP

Input: $g \in J_{\mathcal{H}}(\mathbb{F}_q)$, $h \in \langle g \rangle$, $n = \prod_{i=0}^{j-1} p_i^{e_i}$, where $p_i < p_{i+1}$.

Output: An integer k with $h = [k]g$

1. For $i = 0$ to $j - 1$ do
   - 1.1 $h' \leftarrow \mathcal{O}$, $k_i \leftarrow 0$.
   - 1.2 $g_i \leftarrow (n/p_i)g$.
   - 1.3 For $t = 0$ to $(e_i - 1)$ do
     - 1.3.1 $h_{t,i} \leftarrow (n/p_i^{t+1})(h + h')$.
     - 1.3.2 $W_{t,i} \leftarrow \log_{g_i} h_{t,i}$. {DLP in a subgroup of order $\mathrm{ord}(g_i)$.}
     - 1.3.3 $h' \leftarrow h' - W_{t,i} p_i^t g$.
     - 1.3.4 $k_i \leftarrow k_i + p_i^t W_{t,i}$.
2. Use the CRT to solve the system of congruences $k \equiv k_i \bmod p_i^{e_i}$.
   This gives us $k \bmod n$
3. Return (k)

Lemma 3. Let $\mathcal{H}$ be a hyperelliptic curve of genus 2 defined over finite field $\mathbb{F}_q$ with equation $\mathcal{H}$: $y^2 + h(x)y = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$. Then the addition and double over the Jacobian group of $\mathcal{H}$ are independent of f_1 and f_0.

Proof. Book [1] has given the explicit formulae of adding and doubling over $\mathcal{H}$. We give them in Appendix A. It is obvious that no formulae utilize the parameters f_1 and f_0.

Furthermore, by the formulae F_{2a}, we have the following results. The proof of Lemma 4 and Lemma 5 will be given in Appendix B.

Lemma 4. Let $[u_i, v_i]$ be the Mumford representation of divisors in $J_{\mathcal{H}}(\mathbb{F}_q)$, for $i = 1, 2, 3$, and satisfy $[u_3, v_3] = [u_1, v_1] + [u_2, v_2]$. Suppose that $\deg(u_2) = 2$. Then $u_3, v_{31}$and $v_{30} - v_{20}$ can be represented by $v_{10} - v_{20}$ with coefficients independent of v_{10}, v_{20}.

Lemma 5. Let $[u_i, v_i]$ be the Mumford representation of the divisors in $J_{\mathcal{H}}(\mathbb{F}_q)$, for $i = 1, 2$, and satisfy $[u_2, v_2] = [2][u_1, v_1]$. Suppose $\deg(u_1) = 2$. Then u_2, v_{21} and $v_{20} - v_{10}$ are independent of v_{10}.

Theorem 1. Let $[u_g, v_g]$ and $[u_h, v_h]$ be the Mumford representation of given divisors g, h respectively and satisfy $h = [k]g$ with $\deg(u_g) = 2$. Then u_h, v_{h1} are independent of v_{g0}.

Proof. Assume that g is a reduced divisor. Let $[u_i, v_i] = [i][u_g, v_g]$, $i = 1, 2, \ldots, k$. By Lemma 5 and $[u_2, v_2] = [2][u_g, v_g]$, u_2, v_{21} and $v_{20} - v_{g0}$ are independent of v_{g0}. By Lemma 4 and $[u_3, v_3] = [u_2, v_2] + [u_g, v_g]$, u_3, v_{31} and $v_{30} - v_{g0}$ are rational functions of $v_{20} - v_{g0}$ with coefficients independent of v_{20}, v_{g0}. Then u_3, v_{31} and $v_{30} - v_{g0}$ are independent of v_{30}, v_{g0}. Iteratively, we find that u_k and v_{k1} are independent of v_{g0}. That means u_h and v_{h1} are independent of v_{g0}. This completes the proof of this theorem.

4 Fault attack models on F_{2a}

Consider a crptosystem that uses a strong hyperelliptic curve $\mathcal{H}$ of form (1) defined over finite field $\mathbb{F}_{2^m}$, where m is an odd number. Since this algorithm F_{2a} does not utilize the curve parameters f_1 and f_0, we can insert a fault in the input points so that the computation is carried out exactly in another curve $\widehat{\mathcal{H}}$ with f_1 and f_0 different. The discrete logarithm problem over $\mathcal{H}$ is transfered to that over $\widehat{\mathcal{H}}$. The discrete logarithm over $\mathcal{H}$ can be solved, if $\widehat{\mathcal{H}}$ is a weaker curve with which we can compute the HEC discrete logarithm using the Silver-Pohlig-Hellman algorithm in the cryptographically weaker group $J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$. This work adopts the same single-bit flip fault model as that proposed in [2], which has been shown to be practical [19].

Definition 1. [10] Let $\mathcal{H}$ be a hyperelliptic curve of genus 2 defined over $\mathbb{F}_q$ with equation $\mathcal{H}: y^2 + h(x)y = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$. An invalid curve relative to $\mathcal{H}$ and F_{2a} is a hyperelliptic curve over $\mathbb{F}_q$ with equation $\widehat{\mathcal{H}}: y^2 + h(x)y = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + \widehat{f}_1x + \widehat{f}_0$, where $(f_1, f_0) \neq (\widehat{f}_1, \widehat{f}_0)$.

Let g be the input divisor in $J_{\mathcal{H}}(\mathbb{F}_q)$ which can be represented by $g = \langle P_1\rangle + \langle P_2\rangle - 2\langle\infty\rangle$, where $P_i = (x_{P_i}, y_{P_i}), i = 1, 2$ and $P_1 \neq \pm P_2$. Let $[u_g, v_g]$ be the Mumford representation of g, where u_g, v_g can be written as $u_g = x^2 + u_{g1}x + u_{g0},\ v_g = v_{g1}x + v_{g0}$. From the two representations of the input divisor g, we have the following two fault attack models.

- **Fault Model 1.** Assume that the adversary can inject a flip fault (single bit) into u_{g1} (or u_{g0}) that might occur at random locations of the input divisor $[u_g, v_g]$ of a device computing the HECSM utilizing F_{2a}. Suppose that the resulting Mumford representation after the fault injection is known and is expressed as $\widetilde{u}_g = x^2 + \widetilde{u}_{g1}x + u_{g0},\ v_g = v_{g1}x + v_{g0}$ or $\widetilde{u}_g = x^2 + u_{g1}x + \widetilde{u}_{g0},\ v_g = v_{g1}x + v_{g0}$. Let the Mumford representation of divisor $\widetilde{g}$ be $[\widetilde{u}_g, v_g]$. Suppose that the result $\widetilde{h} = [k]\widetilde{g}$ carried out in $J_{\mathcal{H}}(\mathbb{F}_q)$ is released.
- **Fault Model 2.** Assume that the adversary can inject a random flip fault (single or multiple bit) into the x-coordinate of the input point $P_i = (x_{P_i}, y_{P_i}),\ i = 1, 2$, of a device computing the HECSM by using F_{2a}. Without loss of generality, we assume that the adversary can inject a flip fault in P_1. Suppose that the resulting point after the fault injection is known, denoted by $\widetilde{P}_1 = (x_{\widetilde{P}_1}, y_{P_1})$, satisfying $\widetilde{P}_1 \neq \pm P_2$. Let $\widetilde{g} = \langle\widetilde{P}_1\rangle + \langle P_2\rangle - 2\langle P_\infty\rangle$. Consider that the resulting $\widetilde{h} = [k]\widetilde{g}$ carried out in $J_{\mathcal{H}}(\mathbb{F}_q)$ is released.
- **How to avoid these attacks.** If there is no validation check of the divisor(points) in the hyperelliptic curve($J_{\mathcal{H}}(\mathbb{F}_q)$), our attack really dose work. We want to emphasize the importance of validation check of the divisor(points).

5 Attack algorithm on Model 1

5.1 Attack algorithm by injecting a fault in u_{g1}

By fault Model 1, we can get $[\widetilde{u}_g, v_g]$ by injecting a fault in u_{g1}, where $\widetilde{u}_g = x^2 + \widetilde{u}_{g1}x + u_{g0},\ v_g = v_{g1}x + v_{g0}$. Assume that there exist two different elements $\widetilde{x}_i \in \mathbb{F}_q,\ i = 1, 2$, such that $\widetilde{u}_g(\widetilde{x}_i) = 0,\ i = 1, 2$. Such elements exist with a probability of about 1/2 [11].

Let $[u_{\widetilde{h}}, v_{\widetilde{h}}]$ be the Mumford representation of divisor $\widetilde{h} = [k][\widetilde{u}_g, v_g]$. The scalar multiplication is carried out in $J_{\mathcal{H}}(\mathbb{F}_q)$ by using F_{2a}.

Our attack idea of Model 1 is motivated by the following result.

Theorem 2. Let $\mathcal{H}$ be a genus 2 hyperelliptic curve of form (1) defined over a finite field $\mathbb{F}_q$ of characteristic 2, and $[u_g, v_g]$ be the Mumford representation of the divisor $g \in J_{\mathcal{H}}(\mathbb{F}_q)$. Let $[\widetilde{u}_g, v_g]$ and $[u_{\widetilde{h}}, v_{\widetilde{h}}]$ be defined as above. Then there exists a hyperelliptic curve $\widehat{\mathcal{H}}$ defined over $\mathbb{F}_q$ and divisors $\widehat{g}, \widehat{h} \in J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$ satisfying $u_{\widehat{h}} = u_{\widetilde{h}}$, and $\widehat{h} = k\widehat{g}$. Moreover $u_{\widehat{g}} = x^2 + \widetilde{u}_{g1}x + u_{g0},\ \ v_{\widehat{g}} = v_{g1}x + v_{\widehat{g}0}$, where $v_{\widehat{g}0}$ is an element in $\mathbb{F}_q$.

Proof. Let $u_{g1}(\widetilde{x}_1 - \widetilde{x}_2) = \alpha$. For any $y_{\widehat{P}_1} \in \mathbb{F}_q$, define $y_{\widehat{P}_2} = y_{\widehat{P}_1} - \alpha$. Consider the following linear equation set:

$$\begin{cases} \widehat{f}_1\widetilde{x}_1 + \widehat{f}_0 = y^2_{\widehat{P}_1} + h(\widetilde{x}_1)y_{\widehat{P}_1} - \widetilde{x}_1^5 - f_4\widetilde{x}_1^4 - f_3\widetilde{x}_1^3 - f_2\widetilde{x}_1^2, \\ \widehat{f}_1\widetilde{x}_2 + \widehat{f}_0 = y^2_{\widehat{P}_2} + h(\widetilde{x}_2)y_{\widehat{P}_2} - \widetilde{x}_2^5 - f_4\widetilde{x}_2^4 - f_3\widetilde{x}_2^3 - f_2\widetilde{x}_2^2. \end{cases} \tag{2}$$

By the assumption, the rank of the coefficient matrix $\begin{pmatrix} \widetilde{x}_1 & 1 \\ \widetilde{x}_2 & 1 \end{pmatrix}$ is 2. Therefore, there is a unique solution of $\widehat{f}_1$ and $\widehat{f}_0$ in the above equations set.

Let $\widehat{\mathcal{H}}$ be a hyperelliptic curve over finite field $\mathbb{F}_q$ represented by the following Weierstrass equation:

$$\widehat{\mathcal{H}}: y^2 + h(x)y = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + \widehat{f}_1x + \widehat{f}_0. \tag{3}$$

Let $\widehat{P}_1 =: (\widetilde{x}_1,\ y_{\widehat{P}_1}),\ \widehat{P}_2 =: (\widetilde{x}_2, y_{\widehat{P}_2}),\ \widehat{g} =: \langle\widehat{P}_1\rangle + \langle\widehat{P}_2\rangle - 2\langle\widehat{P}_\infty\rangle$. Obviously $\widehat{P}_1,\ \widehat{P}_2 \in \widehat{\mathcal{H}}(\mathbb{F}_q)$, $\widehat{g} \in J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$. By the definition of $\widehat{g}$, and α, the Mumford representation $[u_{\widehat{g}}, v_{\widehat{g}}]$ of $\widehat{g}$ satisfy $u_{\widehat{g}} = \widetilde{u}_g$ and $v_{\widehat{g}1} = v_{\widetilde{g}1}$.

By Theorem 1, $u_{\widehat{h}}$ is independent of $\widehat{f}_1, \widehat{f}_0$ and $v_{\widehat{g}0}$. Hence we have $u_{\widetilde{h}} = u_{k\widehat{g}}$. Suppose $k\widehat{g}$ can be represented by $k\widehat{g} = \langle\widehat{Q}_1\rangle + \langle\widehat{Q}_2\rangle - 2\langle\widehat{Q}_\infty\rangle$, where $\widehat{Q}_1, \widehat{Q}_2 \in \widehat{\mathcal{H}}(\mathbb{F}_q)$ and $\widehat{Q}_i = (x_{\widehat{Q}_i}, y_{\widehat{Q}_i})$. By the fact that $u\widetilde{h} = u_{k\widehat{g}}$, $x_{\widehat{Q}_i}$ can be obtained by $u_{\widetilde{h}}$. Since $\widehat{Q}_1, \widehat{Q}_2 \in \widehat{\mathcal{H}}(\mathbb{F}_q)$, we can determine $y_{\widehat{Q}_i}$ by equations $\widehat{\mathcal{H}}(x_{\widehat{Q}_i}, y) = 0$, for $i = 1, 2$.

Therefore, we can find a divisor $\widehat{h} \in J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$ such that $\widehat{h} = k\widehat{g}$. This completes the proof of the theorem.

With the divisors pair $\widehat{g}, \widehat{h} \in J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$, one can obtain $k \bmod n$, where $n = \mathrm{ord}(\widehat{g})$. This would be possible if all the prime factors of n are small. The completed attack procedure is presented as Algorithm 2.

There are several remarks on Algorithm 2.

Remark 1. In Algorithm 2, if $\mathrm{ord}(\widehat{g})$ is larger than $\mathrm{ord}(g)$ and all of the prime factor of $\mathrm{ord}(\widehat{g})$ are smaller than ω, then we can get the whole secret integer k.

Remark 2. In Algorithm 2, we chose parameter w according to its practical computation ability. If $\mathrm{ord}(\widehat{g})$ is not an ω smooth integer, we can modify Step 3 in Algorithm 2 as follows: write $\mathrm{ord}(\widehat{g}) = n'n''$, where n' is w smooth integer, compute $(n''\widehat{h}, n''\widehat{g})$. One can get $k \bmod n'$ from $(n''\widehat{h}, n''\widehat{g})$ by using Algorithm 1.

Remark 3. In Algorithm 2, if $\mathrm{ord}(\widehat{g}) < \mathrm{ord}(g)$, we let 2^e be the exhaustive search space. If there is an integer $r \leqslant 2^e$ such that $\mathrm{Lcm}(\mathrm{ord}(\widehat{g}), r) \geqslant \mathrm{ord}(g)$, we can uniquely determine k by solving the system of congruences:

$$\begin{cases} x \equiv k \bmod n', \\ x \equiv l \bmod r, \end{cases}$$

where $l \leqslant r$. Let k_l be the solution of the above congruence. For each k_l, we compute $D = [k_l]g$. If $D = h$, we have $k = k_l$.

5.2 Implemental results of Algorithm 2

We have implemented Algorithm 2 using C++ library NTL[1)]. In this subsection, we give some numerical results by running Algorithm 2. The hyperelliptic curve $\mathcal{H}$ is the Koblitz curve represented by $y^2 + xy = x^5 + x^2 + 1$, which is defined over $\mathbb{F}_{2^m}$ given by a polynomial $f(x)$, where $m \in \{61, 103, 113\}$. Let us represent the elements of $\mathbb{F}_{2^m}$ in hexadecimal form. g is the input divisor in $J_{\mathcal{H}}(\mathbb{F}_{2^m})$ whose Mumford representation is $[u_g, v_g]$. By implementing Algorithm 2, we obtain fault divisors $\widetilde{g}, \widetilde{h}$, invalid curve $\widehat{\mathcal{H}}$, and $\widehat{g}, \widehat{h} \in J_{\widehat{\mathcal{H}}}(\mathbb{F}_{2^m})$ whose Mumford representations are $[u_{\widehat{g}}, v_{\widehat{g}}]$, $[u_{\widehat{h}}, v_{\widehat{h}}]$ respectively, satisfying $\widehat{h} = [k]\widehat{g}$. We list our numerical results in Table 1. Note that the numerical results are obtained by injecting only one bit in u_{g1}.

By injecting one bit in u_{g0}, we can get similar results as above. The attack procedure and the numerical results are presented in Appendix C (see Algorithm C1 and Table C1).

6 Attack algorithm on Model 2

6.1 Attack algorithm by injecting a fault in x_{P_1}

By Model 2, we get divisor $\widetilde{g}$ by injecting one bit fault in x_{P_1}, where $\widetilde{g} = \langle\widetilde{P}_1\rangle + \langle P_2\rangle - 2\langle\infty\rangle$. Then the corresponding polynomial of $u_{\widetilde{g}}, v_{\widetilde{g}}$ can be written as $u_{\widetilde{g}} = (x - x_{\widetilde{P}_1})(x - x_{P_2}),\ v_g = \frac{y_{P_1} - y_{P_2}}{x_{\widetilde{P}_1} - x_{P_2}} x - \frac{x_{\widetilde{P}_1} y_{P_2} - x_{P_2} y_{P_1}}{x_{\widetilde{P}_1} - x_{P_2}}$. Assume that $x_{\widetilde{P}_1} \neq \pm x_{P_2}$. Let $\widetilde{h} = [k]\widetilde{g}$, which can be represented by $\widetilde{h} = \langle\widetilde{Q}_1\rangle + \langle\widetilde{Q}_2\rangle - 2\langle\infty\rangle$, where the computation is carried out in $J_{\mathcal{H}}(\mathbb{F}_q)$ by using F_{2a}.

Our attack on Model 2 is based on the following theorem.

1) Victor Shoup. NTL: A Library for doing Number Theory. http://www.shoup.net/ntl/.

Algorithm 2 Attack algorithm by injecting a fault in u_{g1}

Input: Hyperelliptic curve $\mathcal{H}$, the Mumford representation $[u_g, v_g]$ of a divisor $g \in J_{\mathcal{H}}(\mathbb{F}_q)$, w a parameter.

Output: Scalar k partially with a probability.

1. Inject a fault in u_g for obtaining $\widetilde{u}_g = x^2 + \widetilde{u}_{g1}x + u_{g0}$.
2. Solve $\widetilde{u}_g$ to get $\widetilde{x}_1, \widetilde{x}_2$, if $\widetilde{x}_1, \widetilde{x}_2 \in \mathbb{F}_q$ goto step 3, otherwise goto step 1.
3. Let $\alpha =: v_{g1}(\widetilde{x}_1 - \widetilde{x}_2)$, for any $y_{\widehat{P}_1} \in \mathbb{F}_q$, $y_{\widehat{P}_2} =: y_{\widehat{P}_1} - \alpha$.
   3.1 Given $y_{\widehat{P}_i}$ solve the equation set (2) get $\widehat{f}_1, \widehat{f}_0$.
   3.2 Define $\widehat{\mathcal{H}} : y^2 + h(x)y = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + \widehat{f}_1x + \widehat{f}_0$.
   3.3 Let $\widehat{P}_i = (\widetilde{x}_i, y_{\widehat{P}_i})$, $\widehat{g} =: \langle\widehat{P}_1\rangle + \langle\widehat{P}_2\rangle - 2\langle\infty\rangle$.
   3.4 Compute $n = \mathrm{ord}(\widehat{g})$ in $J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$.
4. If all the prime factors of n are smaller than w, then
   4.1 Compute $\widetilde{h} = [k][\widetilde{u}_g, v_g]$ carried out in $J_{\mathcal{H}}(\mathbb{F}_q)$ by F_{2a}
   4.2 Decompose $u_{\widetilde{h}}$, get the roots $x_{\widehat{Q}_i}, i = 1, 2$.
   4.3 Compute $y_{\widehat{Q}_i} = v_g(x_{\widehat{Q}_i}), i = 1, 2$.
   4.4 Let $\widehat{Q}_i = (x_{\widehat{Q}_i}, y_{\widehat{Q}_i})$, $\widehat{h} =: \langle\widehat{Q}_1\rangle + \langle\widehat{Q}_2\rangle - 2\langle\infty\rangle$.
   4.5 Utilize Algorithm 1 on $J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$ with $(\widehat{g}, \widehat{h}, n)$ to obtain $k \bmod n$.
5. Return $(k \bmod n)$

Actually, in step 4.2, $\widetilde{h}$ is in $J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$ and $u_{\widetilde{h}}$ may have one or two roots in $\mathbb{F}_q$.

Theorem 3. Let $\mathcal{H}$ be a hyperelliptic curve defined over a finite field $\mathbb{F}_q$, $g \in J_{\mathcal{H}}(\mathbb{F}_q)$, and $\widetilde{g}, \widetilde{h}$ be defined as above. There exists a hyperelliptic curve $\widehat{\mathcal{H}}$ defined over $\mathbb{F}_q$ and divisors $\widehat{g}, \widehat{h} \in J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$ such that $\widehat{h} = [k]\widehat{g}$.

See Appendix A for more detail, the proof of Theorem 3 is similar to that of Theorem 2.

With the points pair $\widehat{g}, \widehat{h} \in J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$, one can obtain $k \bmod n$, where $n = \mathrm{ord}(\widehat{g})$. The completed attack procedure is presented in Algorithm 3.

There are several remarks on Algorithm 3 similar to Algorithm 2.

6.2 Implemental examples of Algorithm 3

In this subsection, we give some examples by implementing Algorithm 3. The hyperelliptic curve $\mathcal{H}$ is the Koblitz curve represented by $y^2 + xy = x^5 + x^2 + 1$, which is defined over $\mathbb{F}_{2^m}$ and given by a polynomial $f(x)$, where $m \in \{61, 103, 113\}$. Let us represent the elements of $\mathbb{F}_{2^m}$ in hexadecimal form. g is the input divisor in $J_{\mathcal{H}}(\mathbb{F}_{2^m})$ which can be represented by $g = \langle P_1\rangle + \langle P_2\rangle - 2\langle\infty\rangle$, where $P_i = (x_{P_i}, y_{P_i}),\ i = 1, 2$. By implementing Algorithm 3, we obtain fault divisors $\widetilde{g}, \widetilde{h}$, invalid curve $\widehat{\mathcal{H}}$, and $\widehat{g}, \widehat{h} \in J_{\widehat{\mathcal{H}}}(\mathbb{F}_{2^m})$ where the Mumford representations of $\widetilde{g}, \widetilde{h}$ and $\widehat{g}, \widehat{h}$ are the same, satisfying $\widehat{h} = [k]\widehat{g}$. We list our examples in Table 2. Note that the numerical results are obtained by injecting only one bit in x_{P_i}.

7 Efficiency of the attack method

In this section, we analyze the efficiency of our attack method.

7.1 Success probability of this attack

Most of the computational cost of Algorithm 2 and Algorithm 3 is involved in obtaining k by partially using the Silver-Pohlig-Hellman algorithm (Algorithm 1) and the exhaustive search in Remark 3. Silver-Pohlig-Hellman algorithm need to compute one HEC discrete logarithm. This operation can be performed with a fast algorithm for HECDLP such as Pollard's rho algorithm [25] with an expected number of point operations about $3\sqrt{p_{t-1}}$, where p_{t-1} is the largest prime divisor of n. An efficient algorithm was provided in [26] to compute the order of Jacobian group for Hyperelliptic curve of characteristic 2. The

Table 1 Insert a flip fault in u_{g1}

Curve specification $m = 61$, $p(x) = x^{61} + x^5 + x^2 + x + 1$
$u_g = x^2 + \mathrm{0x5003d8b67eb7d6f}\,x + \mathrm{0xa8ee05ac09be989}$
$v_g = \mathrm{0x23820d5e5fa3048}\,x + \mathrm{0x074c4c18be9e74b}$
$\mathrm{ord}(g) = 2658455988447243530986550320280662477$
$k = 434798374983234574983$
$u_{\widetilde{g}} = x^2 + \mathrm{0x4003d8b67eb7d6f}\,x + \mathrm{0xa8ee05ac09be989}$
$v_{\widetilde{g}} = \mathrm{0x23820d5e5fa3048}\,x + \mathrm{0x074c4c18be9e74b}$
$u_{\widetilde{h}} = x^2 + \mathrm{0x8703af391365c41}\,x + \mathrm{0x3b7ccd02439f5b8}$
$v_{\widetilde{h}} = \mathrm{0x169f56a9082dd2e1}\,x + \mathrm{0x49fb9a1f732c329}$
$\widehat{\mathcal{H}} : y^2 + xy = x^5 + x^2 + \mathrm{0xdeb1c6db60e71721}\,x + \mathrm{0xdca12071e07a681}$
$u_{\widehat{g}} = x^2 + \mathrm{0x4003d8b67eb7d6f}\,x + \mathrm{0xa8ee05ac09be989}$
$v_{\widehat{g}} = \mathrm{0x23820d5e5fa3048}\,x + \mathrm{0x99650d58d879df2}$
$\mathrm{ord}(\widehat{g}) = (3)(17)(263)(40609)(30294782659877)(53702210072963)$
$u_{\widehat{h}} = x^2 + \mathrm{0x8703af391365c41}\,x + \mathrm{0x3b7ccd02439f5b8}$
$v_{\widehat{h}} = \mathrm{0x169f56a9082dd2e1}\,x + \mathrm{0xd7d2db5f15cb99}$
Curve specification $m = 103$, $p(x) = x^{103} + x^9 + 1$
$u_g = x^2 + \mathrm{0xeee2d5c07a6bd93a0c59833ba4}\,x + \mathrm{0xa48824b71e13215936f3cfa563}$
$v_g = \mathrm{0xc7224fb356bd2cd32e4a5c14f3}\,x + \mathrm{0xfdf1b8f10539754f7b3b50e2c4}$
$\mathrm{ord}(g) = 10852877190495703277390509258459145399489273609233701107 69$
$k = 479837498327498323543656758279 57$
$u_{\widetilde{g}} = x^2 + \mathrm{0xeee2d5c07a6bdd3a0c59833ba4}\,x + \mathrm{0xa48824b71e13215936f3cfa563}$
$v_{\widetilde{g}} = \mathrm{0xc7224fb356bd2cd32e4a5c14f3}\,x + \mathrm{0xfdf1b8f10539754f7b3b50e2c4}$
$u_{\widetilde{h}} = x^2 + \mathrm{0xeb74574c92bcf7117d5bca8dd2}\,x + \mathrm{0x76b4d8428e57f0cb9a875cee82}$
$v_{\widetilde{h}} = \mathrm{0xacdb9fa0ed3f5dbcd7739723c2}\,x + \mathrm{0x11d81fb7039db7fa36ba893783}$
$\widehat{\mathcal{H}} : y^2 + xy = x^5 + x^2 + \mathrm{0xffe19155edbbbbc589c2452b27}\,x$
$\quad + \mathrm{0x2d2f25e94392ada846ececf413}$
$u_{\widehat{g}} = x^2 + \mathrm{0xeee2d5c07a6bdd3a0c59833ba4}\,x + \mathrm{0xa48824b71e13215936f3cfa563}$
$v_{\widehat{g}} = \mathrm{0xc7224fb356bd2cd32e4a5c14f3}\,x + \mathrm{0x76c96ef71d21ada89f9364f757}$
$\mathrm{ord}(\widehat{g}) = (2)(3)(23)(499)(52345739)(102687017779)(2416263581169375187)$
$\quad (38329842543370836539)$
$u_{\widehat{h}} = x^2 + \mathrm{0xeb74574c92bcf7117d5bca8dd2}\,x + \mathrm{0x76b4d8428e57f0cb9a875cee82}$
$v_{\widehat{h}} = \mathrm{0xacdb9fa0ed3f5dbcd7739723c2}\,x + \mathrm{0x9ae0c9b11b856f1dd212bd221}$
Curve specification $m = 113$, $p(x) = x^{113} + x^9 + 1$
$u_g = x^2 + \mathrm{0xc2b96348cc58e038b71178a9a38b}\,x + \mathrm{0x3b358cf39d80854ad0b4d8ed5f43}$,
$v_g = \mathrm{0x812bd9b8364583ca9abe1ddac461}\,x + \mathrm{0xa6d4259ef3709c31246fdf8cce661}$
$\mathrm{ord}(g) = 539198933343012787158232976738412307606428027150190435497641933 68381$
$k = 479837498327498354365675827957$
$u_{\widetilde{g}} = x^2 + \mathrm{0xe2b96348cc58e038b71178a9a38b}\,x + \mathrm{0x3b358cf39d80854ad0b4d8ed5f43}$,
$v_{\widetilde{g}} = \mathrm{0x812bd9b8364583ca9abe1ddac461}\,x + \mathrm{0xa6d4259ef3709c31246fdf8cce661}$
$u_{\widetilde{h}} = x^2 + \mathrm{0x9618ec3ab49dde5afec0ff40ee1d}\,x + \mathrm{0xc89eb90e270f5072a870244ee4761}$
$v_{\widetilde{h}} = \mathrm{0xd58145b4f23e3be0150195e47759}\,x + \mathrm{0xc9145b2904fba6e0f911e34bf2181}$
$\widehat{\mathcal{H}} : y^2 + xy = x^5 + x^2 + \mathrm{0x125242763d3b9b9d2bd6ad9c49ca}\,x$
$\quad + \mathrm{0xe95c3e0ba8e66dd0c807ef61c0911}$
$u_{\widehat{g}} = x^2 + \mathrm{0xe2b96348cc58e038b71178a9a38b}\,x + \mathrm{0x3b358cf39d80854ad0b4d8ed5f43}$
$v_{\widehat{g}} = \mathrm{0x812bd9b8364583ca9abe1ddac461}\,x + \mathrm{0x476adaa20236340dc6ea942b28611}$
$\mathrm{ord}(\widehat{g}) = (2)(5)(503)(12046651)(183064547)(5637681901967)(24099893265761)$
$\quad (71552493695623998215629)$
$u_{\widehat{h}} = x^2 + \mathrm{0x9618ec3ab49dde5afec0ff40ee1d}\,x + \mathrm{0xc89eb90e270f5072a870244ee4761}$
$v_{\widehat{h}} = \mathrm{0xd58145b4f23e3be0150195e47759}\,x + \mathrm{0x28aaa415f5bd0edc1b94a8ec141f1}$

Algorithm 3 Attack algorithm by injecting a fault x_{P_1}

Input: Hyperelliptic curve $\mathcal{H}$, $g \in J_{\mathcal{H}}(\mathbb{F}_q)$, $g = \langle P_1 \rangle + \langle P_2 \rangle - 2\langle P_\infty \rangle$,
$P_i = (x_{P_i}, y_{P_i})$, $i = 1, 2$, w a parameter to be chosen later.
Output: Scalar k partially with a probability.
1. Inject a fault in $P_1 = (x_{P_1}, y_{P_1})$ for obtaining $\widetilde{P}_1 = (x_{\widetilde{P}_1}, y_{P_1})$.
2. Let $\alpha =: y_{P_1} - y_{P_2}$, for any $y_{\widehat{P}_1} \in \mathbb{F}_q$, $y_{\widehat{P}_2} =: y_{\widehat{P}_1} - \alpha$.
  3.1 Given $y_{\widehat{P}_i}$ solve the equation set (4) get $\widehat{f}_1, \widehat{f}_0$.
  3.2 Define $\widehat{\mathcal{H}} : y^2 + h(x)y = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + \widehat{f}_1x + \widehat{f}_0$.
  3.3 Let $\widehat{P}_i = (x_{\widehat{P}_i}, y_{\widehat{P}_i})$, $\widehat{g} =: \langle \widehat{P}_1 \rangle + \langle \widehat{P}_2 \rangle - 2\langle \infty \rangle$.
  3.4 Obtain $n = \mathrm{ord}(\widehat{g})$ in $J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$.
3. If all the prime factors of n are smaller than w, then
  3.1 Compute $\widetilde{h} = [k]\widetilde{g}$ carried in $J_{\mathcal{H}}(\mathbb{F}_q)$ by F_{2a}
  3.2 Decompose $u_{\widetilde{h}}$, get the roots $x_{\widetilde{Q}_i}, i = 1, 2$.
  3.3 Compute $y_{\widehat{Q}_i} = v_g(x_{\widetilde{Q}_i}), i = 1, 2$.
  3.4 Let $\widehat{Q}_i = (x_{\widetilde{Q}_i}, y_{\widehat{Q}_i})$, $\widehat{h} =: \langle \widehat{Q}_1 \rangle + \langle \widehat{Q}_2 \rangle - 2\langle \infty \rangle$.
  3.5 Utilize Algorithm 1 on $J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$ with $(\widehat{g}, \widehat{h}, n)$ to obtain $k \bmod n$.
4. Return ($k \bmod n$)
In fact, in step 3.2, $\widetilde{h}$ is in $J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$ and $u_{\widetilde{h}}$ may have one or two roots in $\mathbb{F}_q$.

order of $\widehat{g}$ can be efficiently computed. The exhaustive search space depends on the order of $\widehat{g}$, and the order of g.

Let $\mathcal{H}$ be hyperelliptic curve of genus 2 defined over $\mathbb{F}_q$, we have $\sharp J_{\mathcal{H}}(\mathbb{F}_q) \in [(\sqrt{q}-1)^4, (\sqrt{q}+1)^4]$, where $\sharp$ denotes cardinality. In Algorithm 2 and Algorithm 3, we can find a hyperelliptic curve $\widehat{\mathcal{H}}$ of genus 2 defined over $\mathbb{F}_q$ and a divisor $\widehat{g} \in J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$. Without loss of generality, we assume $\mathrm{ord}(\widehat{g}) \in [(\sqrt{q}-1)^4, (\sqrt{q}+1)^4]$.

For $n \in \mathbb{N}$, let $S_1(n)$ denote the largest prime divisor of n. For random integers n, Knuth et al. [27] showed that $\mathrm{Prob}[S_1(n) \leqslant \omega] \approx \rho(\log n/\log \omega)$, where $\rho(u)$ is the Dickman-de Bruijn function satisfying $u\rho'(u) + \rho(u-1) = 0$.

A fault is called a valid fault if the resulting divisor which we get by injecting the fault in the input divisor satisfies Theorem 2 or Theorem 3. Let t be the number of locations where we can inject a valid fault. From Algorithm 2 and Algorithm 3, we can obtain a Jacobian group of a hyperelliptic curve over $\mathbb{F}_q$ whose cardinality is an ω smooth integer with probability at least $1 - (1 - \rho(\log n/\log \omega))^t$.

Given a hyperelliptic curve $\mathcal{H}$ defined over $\mathbb{F}_{2^m}$, $m \in \{61, 103, 113\}$, Tables 3 and 4 give the probability of running Algorithm 2 and Algorithm 3 to get a invalid hyperelliptic curve. The cardinality of the Jacobian group of the invalid hyperellipti curve is an ω smooth integer.

7.2 Experimental results

This subsection reports our experimental results of these fault attacks on three Koblitz curves. The Koblitz curve is defined by $y^2 + xy = x^5 + x^2 + 1$ over $\mathbb{F}_{2^m}$, where $m \in \{61, 103, 113\}$. We test all the results after inserting a flip fault in u_1, u_0 and x_i for $i = 1, 2$. Taking $m = 113$, for example, find that an invalid curve requires 28.379 s with a total of 64.47 MB memory usage(on Intel(R) Core(TM) 2 Duo CPU) including the factorization of the cardinality.

Figure 1 shows the size in bits of the biggest prime factor of the cardinality of the Jacobian group of all the feasible invalid curves. Owing to space constraints, we only give the result of attack on Koblitz hyperelliptic curve over $\mathbb{F}_{2^{113}}$. From the algorithm described above, we can inject 56 faults in u_1, 59 faults in u_0, and 201 faults in x_i. The security level in bits of Koblitz hyperelliptic curve over $\mathbb{F}_{2^{113}}$ is 102. There are 107 invalid curves whose security level in bits is less than 50, and there are 18 invalid curves whose security level in bits is less than 30.

In Table 5, we show the best result of attacking three Koblitz curves. The biggest prime factor of the weakest invalid curves has 42, 41, 36 bits respectively for $m = 61$; 58, 53, 57 for $m = 103$; 68, 69, 39

Table 2 Inserting a flip fault in x_i

Curve specification $m = 61$, $p(x) = x^{61} + x^5 + x^2 + x + 1$

$$g = \langle 0x8900a8b93a076f6, 0x3923bf8285950e7\rangle + \langle 0xd903700f44b0b99, 0x4a5825cb088983f1\rangle - 2\langle\infty\rangle$$

$$\mathrm{ord}(g) = 2658455988447243530986550320280662477$$

$$\widehat{\mathcal{H}} : y^2 + xy = x^5 + x^2 + \mathrm{0xb7cc1cd225fd9781}x + \mathrm{0x7d37ef99a3923b3}$$

$$\widehat{g} = \langle 0xc900a8b93a076f6, 0x3923bf8285950e7\rangle + \langle 0xd903700f44b0b99, 0x4a5825cb088983f1\rangle - 2\langle\infty\rangle$$

$$\mathrm{ord}(\widehat{g}) = (2)(23)(47)(599)(261409249)(9975575507)(314882152177)$$

$$k = 4347983749832345749 83$$

$$\widehat{h} = \langle 0xbd710d522c5dbee, 0x22b7376f98697a\rangle + \langle 0x73ac9b38ae2430c1, 0x22feef1f0a4819b1\rangle - 2\langle\infty\rangle$$

Curve specification $m = 103$, $p(x) = x^{103} + x^9 + 1$

$$g = \langle 0x2a3279a1aa8cf29c3f8acae6b3, 0x755b7ec0c057b9d804eb133b54\rangle + \langle 0xc4d0ac61d0e72ba633d349dd17, 0x6716f3b50a6cb699e8993f9a01\rangle - 2\langle\infty\rangle$$

$$\mathrm{ord}(g) = (10852877190495703277390509258459145399489273609233701107 69)$$

$$\widehat{\mathcal{H}} : y^2 + xy = x^5 + x^2 + \mathrm{0xf52d451fde9ff75f80365df8c4}x + \mathrm{0x3315f129ec8226ce9dcf16e071}$$

$$\widehat{g} = \langle \mathrm{0x2a3279a1aa8cf29c3f8acae6b3}, \mathrm{0x755b7ec0c057b9d804eb133b54}\rangle + \langle \mathrm{0xccd0ac61d0e72ba633d349dd17}, \mathrm{0x6716f3b50a6cb699e8993f9a01}\rangle - 2\langle\infty\rangle$$

$$\mathrm{ord}(\widehat{\mathrm{g}}) = (2)(5)(29)(31)(2045987)(694226125567609)(1606257785136088771)(5014184917771227827)$$

$$k = 479837498327498354365675827957$$

$$\widehat{h} = \langle \mathrm{0x459ae03b3260f17b0931b4c853}, \mathrm{0x67e4d19032f3d9f96c7b29ae75}\rangle + \langle \mathrm{0xc078869148a2840228a89204}, \mathrm{0xaa6d6a0fd301d60bf693f6775}\rangle - 2\langle\infty\rangle.$$

Curve specification $m = 113$, $p(x) = x^{113} + x^9 + 1$

$$g = \langle 0x7d58cac12e5122476d1ab89c8c57, 0x0231395d3e67ac81149cc1b5c581\rangle + \langle 0xbfe1a989e209c27fda0bc0352fdc, 0xcb45a10bd42f4e8758b1b459f8641\rangle - 2\langle\infty\rangle$$

$$\mathrm{ord}(g) = 53919893334301278715823297673841230760642802715019043549764193368381$$

$$\widehat{\mathcal{H}} : y^2 + xy = x^5 + x^2 + \mathrm{0xf8dd487b294ad77a55fe40c3912c}x + \mathrm{0xdaedbdfa7d1d3824ba2d964f4c9d1}$$

$$\widehat{g} = \langle \mathrm{0x7d58cac12e5122476d1ab89c8e57}, \mathrm{0x0231395d3e67ac81149cc1b5c581}\rangle + \langle \mathrm{0xbfe1a989e209c27fda0bc0352fdc}, \mathrm{0xcb45a10bd42f4e8758b1b459f8641}\rangle - 2\langle\infty\rangle$$

$$\mathrm{ord}(\widehat{g}) = (5)(23)(83)(2928268957)(5143307119)(15240965639)(59409661109)(63353145481)(6538557223013)$$

$$k = 479837498327498354365675827957$$

$$\widehat{h} = \langle \mathrm{0x88ad369b0b05288e7c22a7424fc4}, \mathrm{0x4483c387455dd631df98408504c8}\rangle + \langle \mathrm{0xaf358d917992d2162418747b0aa31}, \mathrm{0x049769825db3fb0c809e08ebdc711}\rangle - 2\langle\infty\rangle$$

Table 3 Probability of attack model 1

q	$\sharp(J_{\mathcal{H}}(\mathbb{F}_q))$	t	ω	Probability
2^{61}	2^{122}	30	2^{60}	0.99998
2^{103}	2^{206}	51	2^{69}	0.9212
2^{113}	2^{226}	56	2^{75}	0.9386

Table 4 Probability of attack model 2

q	$\sharp(J_{\mathcal{H}}(\mathbb{F}_q))$	t	ω	Probability
2^{61}	2^{122}	60	2^{60}	0.99999
2^{103}	2^{206}	102	2^{69}	0.9937
2^{113}	2^{226}	112	2^{75}	0.96227

for $m = 113$. It is feasible to solve discrete logarithm problem of these weakest invalid curves by using Silver-Pohlig-Hellman algorithm.

At present, bit size of the security level in practical cryptsystem is 80. The security level in bits of Koblitz hyperelliptic curve over $\mathbb{F}_{2^{103}}$ and $\mathbb{F}_{2^{113}}$ are 87 and 102, respectively. Hence, the Jacobian group Koblitz hyperelliptic curve over $\mathbb{F}_{2^{103}}$ and $\mathbb{F}_{2^{113}}$ can be applied to design cryptosysytem. In Table 5, the security level in bits of the invalid curves are 27 and 20 respectively, i.e. we can solve discrete logarithm problem of these weakest invalid curves with one second by utilizing Silver-Pohlig-Hellman algorithm.

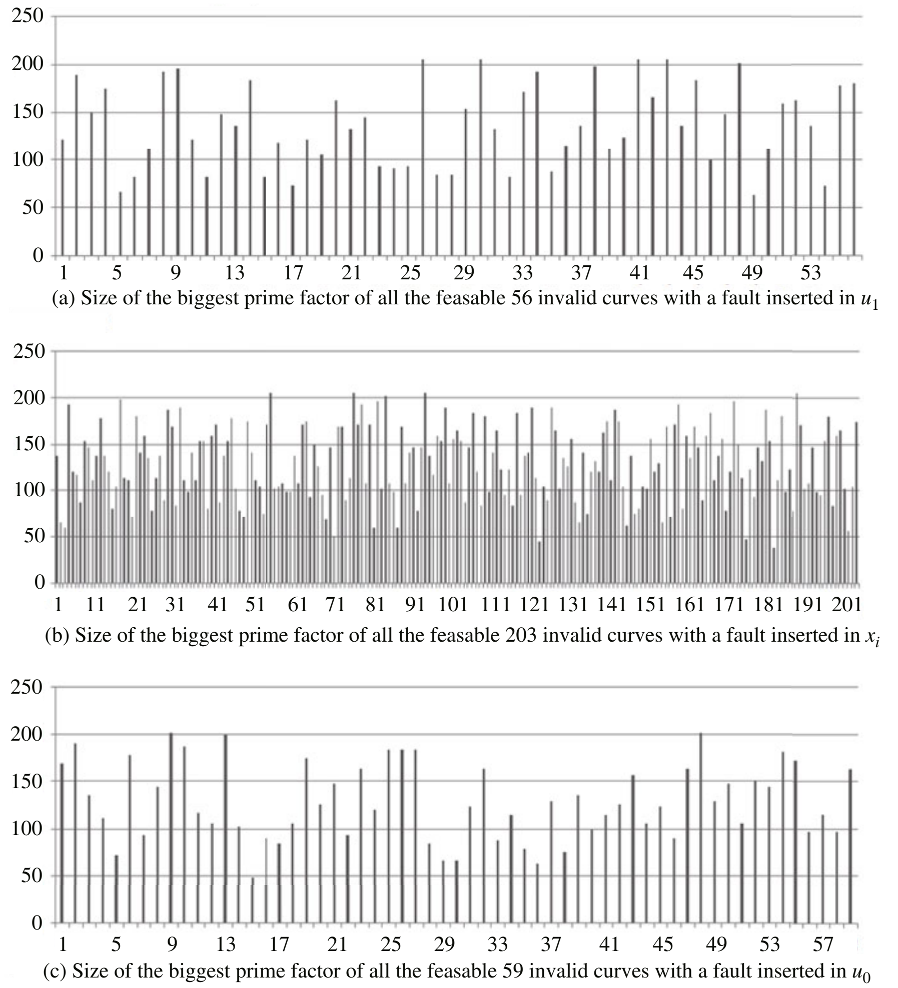

Figure 1 Attack Koblitz curve with $m = 113$. The thick black horizontal line denotes the size of the biggest prime factor of $J_{\mathcal{H}(2^{113})}$ in bits. The vertical lines denote the size of the biggest prime factor of all the feasible invalid curves's cardinality.

Table 5 Size of each prime factor of Koblitz curve and the weakest curve

m	Attack model	Curves	Size of each prime factor of $\sharp(J(\mathbb{F}_q))$
61		Koblitz curve 61	2, 101
	Insert fault in u_1	The Weakest curve	2, 5, 9, 15, 42, 42
	Insert fault in u_0	The Weakest curve	2, 7, 8, 14, 17, 27, 41
	Insert fault in x_i	The Weakest curve	2, 5, 6, 9, 27, 30, 36
103		Koblitz curve 103	2, 15, 174
	Insert fault in u_1	The Weakest curve	2, 2, 5, 9, 24, 36, 57, 58
	Insert fault in u_0	The Weakest curve	2, 14, 29, 39, 51, 53
	Insert fault in x_i	The Weakest curve	2, 3, 5, 5, 21, 45, 57
113		Koblitz curve 113	2, 204
	Insert fault in u_1	The Weakest curve	2, 3, 9, 24, 27, 39, 41, 68
	Insert fault in u_0	The Weakest curve	3, 9, 23, 26, 39, 41, 69
	Insert fault in x_i	The Weakest curve	3, 5, 6, 30, 30, 32, 33, 33, 39

Therefore, we can get the discrete logarithm of the original curves efficiently.

8 Conclusion

In this paper, we have presented invalid-curve attacks according to the representation of divisors in Jacobian group of a hyperelliptic curve that applies to the hyperelliptic curve scalar multiplication (HECSM) algorithm on the genus 2 hyperelliptic curve over binary field. These attacks exploit the fact that the parameters of the hyperelliptic curve equation f_0, f_1 are not used in the group formula for these particular algorithm. By injecting a one bit fault in the input divisor, we may find a hyperelliptic curve $\widehat{\mathcal{H}}$ with the same parameters as the original hyperelliptic curve $\mathcal{H}$ except for parameters f_0, f_1, and the cardinality of the Jacobian group $J_{\widehat{H}}(\mathbb{F}_{2^m})$ is an ω smooth integer. By taking Koblitz as a target curve $\mathcal{H}$, we have shown some weaker Jacobian groups of the resulting hyperelliptic curve $\widehat{\mathcal{H}}$. Finally, we have obtained theoretical probability of getting a hyperelliptic curve whose Jacobian group is weak.

Acknowledgements

This work was supported by National Basic Research Program of China (973) (Grant No. 2013CB834205), National Natural Science Foundation of China (Grant Nos. 61272035, 61133013), and Independent Innovation Foundation of Shandong University (Grant No. 2012JC020).

References

1 Avanzi R, Cohen H, Docke C, et al. Handbook of elliptic and hyperelliptic curve cryptography. Boca Raton: Chapman & Hall/CRC, 2005. 316–320
2 Boneh D, DeMillo R A, Lipton R J. On the importance of eliminating errors in cryptographic computations. J Cryptol, 2001, 14: 101–119
3 Rivest R L, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Commun ACM, 1978, 21: 120–126
4 Rabin M O. Digitalized signatures and public-key functions as intractable as factorization. Massachusetts Institute of Technology Laboratory for Computer Science Technical Report 212. 1979
5 Biehl I, Meyer B, Müller V. Differential fault attacks on elliptic curve cryptosystems. In: International Cryptology Conference, Santa Barbara, 2000. 131–146
6 Koblitz N. Elliptic curve cryptosystems. Math Comput, 1987, 48: 203–209
7 Miller V. Use of elliptic curves in cryptography. In: International Cryptology Conference, Santa Barbara, 1985. 417–426
8 Wang M Q, Zhan T. Analysis of the fault attack ECDLP over prime field. J Appl Math, 2011, 2011: 1–11
9 Ciet M, Joye M. Elliptic curve cryptosystems in the presence of permanent and transient faults. Designs Codes Cryptogr, 2005, 36: 33–43
10 Karabina K, Ustaoglu B. Invalid-curve attack on (hyper)elliptic curve cryptosystems. Adv Math Commun, 2010, 4: 307–321
11 Dominguez-Oviedo A, Hasan M A, Ansari B. Fault-based attack on Montgomerys ladder algorithm. J Cryptol, 2011, 24: 346–374
12 Wang M Q, Xue H Y, Zhan T. Fault attacks on hyperelliptic curve discrete logarithm problem over finite fields. China Commun, 2012, 9: 150–161
13 Blöer J, Otto M, Seifert J-P. Sign change attacks on elliptic curve cryptosystems. In: International Workshop on Fault Diagnosis and Tolerance in Cryptography, Yokohama, 2006. 36–42
14 Frey G, Ruck H. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. Math Comput, 1994, 62: 865–874
15 Pontarelli S, Cardarilli G, Re M, et al. Error detection in addition chain based ECC point multiplication. In: IEEE International On-Line Testing Symposium, Sesimbra Lisbon, 2009. 192–194
16 Stern R, Joshi N, Wu K, et al. Register transfer level concurrent error detection in elliptic curve crypto implementations. In: Workshop on Fault Diagnosis and Tolerance in Cryptography, Washington, 2007. 112–119
17 Dominguez-Oviedo A, Hasan M A. Error detection and fault tolerance in ECSM using input randomization. IEEE Trans Dependable Secur Comput, 2009, 6: 175–187
18 Anderson R, Kuhn M. Low cost attacks on tamper resistant devices. In: International Workshop on Security Protocols, Paris, 1997. 125–136
19 Skiribogatov S, Anderson R. Optical fault induction attacks. In: International Workshop on Cryptographic Hardware and Embedded Systems, Redwood Shores, 2002. 2–12

20 Kim C H, Quisquater J-J. Fault attacks for CRT based RSA: new attacks, new results, and new countermeasures. In: Proceedings of the 1st IFTP TC6/WG8.8/WG11.2 International Conference on Information Security Theory and Practices: Smart Cards, Mobile and Ubiquitous Computing Systems, Heraklion, 2007. 215–228

21 Günther C, Lange T, Stein A. Speeding up the arithmetic on Koblitz curves of genus two. In: International Workshop on Selected Areas in Cryptography, Waterloo, 2000. 106–117

22 Pohlig S, Hellman M. An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. IEEE Trans Inf Theory, 1978, 24: 106–110

23 Pelzl J, Wollings T, Paar C. High performance arithmetic for hyperelliptic curve cryptosystem of genus two. Cryptology ePrint Archive, Report 2003/212, 2003. http://eprint.iacr.org/

24 Lange T, Stevens M. Efficient doubling on genus two curves over binary fields. In: International Conference on Selected Areas in Cryptography, Waterloo, 2004. 170–181

25 Pollard J M. Monte Carlo methods for index computation (mod p). Math Comput, 1978, 32: 918–924

26 Gaudry P, Harley R. Counting points on hyperelliptic curves over finite fields. In: International Symposium of Algorithmic Number Theory, Leiden, 2000. 313–332

27 Knuth D E, Trabb-Pardo L. Analysis of a simple factorization algorithm. Theor Comput Sci, 1976, 3: 321–348

Appendix A Formulae F_{2a}

Here, we show the explicit formulae on $\mathcal{H}$, and we denote them by F_{2a} in this paper (see Tables A1–A3).

Appendix B Proofs of Lemma 4 and Lemma 5

Proof of Lemma 4. We divide the proof of this lemma into two cases: $\deg(u_1) = 2$ and $\deg(u_1) = 1$.

Case 1. Assume $\deg(u_1) = 2$, and define $r = (u_{20} - u_{10})(u_{11}^2 - u_{11}u_{21} + u_{20} - u_{10}) + (u_{11} - u_{21})^2$, $s'_0 = [(u_{11} - u_{21})(u_{11} + 1) - u_{10} + u_{20}](v_{10} - v_{20} + v_{11} - v_{21})$, $s'_1 = (u_{11}^2 - u_{11}u_{21} + u_{20} - u_{10})(v_{11} - v_{21}) + (u_{11} - u_{21})(v_{10} - v_{20}) - u_{11}(u_{11} - u_{21})(v_{11} - v_{21})$. By the definition, r is independent of v_{10}, v_{20}, and s'_0, s'_1 can be linearly represented by $v_{10} - v_{20}$ with coefficient independent of v_{10}, v_{20}.

By a rather complex computation, we have: If $s'_1 \neq 0$, then $u_{31} = \frac{r^2}{s_1'^2} + u_{11} - u_{21}$, $u_{30} = u_{21}\frac{s'_0}{s'_1}h_1\frac{r}{s'_1} + (u_{11} - u_{21} - f_4)\frac{s_0'^2}{s_1'^2} + (\frac{s'_0}{s'_1} - u_{11})(\frac{s'_0}{s'_1} - u_{11} - u_{21}) - u_{10}$, $v_{31} = [u_{31}(u_{21} + \frac{s'_0}{s'_1}) - u'_0 - u_{21}\frac{s'_0}{s'_1} + u_{20}]\frac{s'_0}{r} - v_{21} - h_1$, $v_{30} - v_{20} = [u_{30}(u_{21} + \frac{s'_0}{s'_1}) - u_{20}\frac{s'_0}{s'_1}]\frac{s'_0}{r} - h_0$. If $s'_1 = 0$, then $u_{31} = 1$, $u_{30} = u_{11}\frac{s_0'^2}{r^2} + f_4 + u_{11} + u_{21}$, $v_{31} = 0$, $v_{30} - v_{20} = u_{30}[(u_{21} + u_{30})\frac{s'_0}{r} + h_1 + v_{21}] - u_{20}\frac{s'_0}{r} - h_0$. By the above formula, it is not difficult to see that u_{31}, u_{30}, v_{31}, and $v_{30} - v_{20}$ are rational functions of r, s'_0, s'_1 with coefficient independent of v_{10}, v_{20}.

Case 2. Assume $\deg(u_1) = 1$, and define $s = \frac{v_{10} - v_{20} - v_{21}u_{10}}{u_{20} - u_{21}u_{10} + u_{10}^2}$. A similar computation as in case 1 shows that $u_{31} = f_4 - u_{21} - s^2 - u_{10}$, $u_{30} = f_3 - (f_4 - u_{21})u_{21} - u_{20} - s(su_{21} + h_1) - u_{10}u_{31}$, $v_{31} = s(u_{31} - u_{21}) - v_{21} - h_1$, $v_{30} - v_{20} = s(u_{30} - u_{20}) - h_0$. By the definition, s is a linear representation of $v_{10} - v_{20}$ with coefficient independent of v_{10}, v_{20}. The above formula shows that u_{31}, u_{30}, v_{31} and $v_{30} - v_{10}$ are rational functions of s with coefficient independent of v_{10}, v_{20}. This completes the proof of the lemma.

Proof of Lemma 5. According to formulae for doubling over binary fields in case $\deg(u) = 2$, we define $s'_0 = (u_{11}^2v_{11} + f_4u_{11}^2 + f_2 - v_{11}^2 - h_1v_{11})(h_0 - u_{11}v_{11}) - u_{10}h_1(f_3 + u_{11}^2)$, $s'_1 = (h_0 - h_1 - u_{11}v_{11})(f_3 + u_{11}^2 + u_{11}^2v_{11} + f_4u_{11}^2 + f_2 - v_{11}^2 - h_1v_{11}) - (u_{11}^2v_{11} + f_4u_{11}^2 + f_2 - v_{11}^2 - h_1v_{11})(h_0 - u_{11}v_{11}) + h_1(f_3 + u_{11}^2)(1 + u_{11})$.

If $s'_0 \neq 0$, then $u_{21} = \frac{r^2}{s_1'^2}$, $u_{20} = \frac{s_0'^2}{s_1'^2} + \frac{r}{s'_1}h_1 - \frac{r^2}{s_1'^2}f_4$, $v_{21} = [u_{21}(u_{11} + \frac{s'_0}{s'_1} - u_{21}) - u_{11}\frac{s'_0}{s'_1}]\frac{s'_1}{r} - v_{11} - h_1$, $v_{20} = [u_{20}(u_{11} + \frac{s'_0}{s'_1} - u_{21}) - u_{10}\frac{s'_0}{s'_1}]\frac{s'_1}{r} - v_{10} - h_0$. If $s'_0 = 0$, then $u_{21} = 1$, $u_{20} = f_4 - \frac{s_0'^2}{r^2}$, $v_{21} = 0$, $v_{20} = u_{20}[\frac{s'_0}{r}(u_{20} + u_{11}) + h_1 + v_{11}] - u_0\frac{s'_0}{r} - v_{10} - h_0$.

By the definition, s'_0, s'_1 is independent of v_{10}, u_{21}, u_{20}, v_{21} and $v_{21} - v_{10}$ are rational functions of s'_0, s'_1 with coefficient independent of v_{10}. So u_{21}, u_{20}, v_{21} and $v_{21} - v_{10}$ are independent of v_{10}.

Proof of Theorem 6. Let $y_{P_1} - y_{P_2} = \alpha$. For any $y_{\widehat{P}_1} \in \mathbb{F}_q$, define $y_{\widehat{P}_2} = y_{\widehat{P}_1} - \alpha$. Consider the following liner equation set:

$$\begin{cases} f_1x_{\widetilde{P}_1} + f_0 = y_{\widehat{P}_1}^2 + h(x_{\widetilde{P}_1})y_{\widehat{P}_1} - x_{\widetilde{P}_1}^5 - f_4x_{\widetilde{P}_1}^4 - f_3x_{\widetilde{P}_1}^3 - f_2x_{\widetilde{P}_1}^2, \\ f_1x_{P_2} + f_0 = y_{\widehat{P}_2}^2 + h(x_{P_2})y_{\widehat{P}_2} - x_{P_2}^5 - f_4x_{P_2}^4 - f_3x_{P_2}^3 - f_2x_{P_2}^2. \end{cases} \tag{B1}$$

By assumption $x_{\widetilde{P}_1} \neq x_{P_2}$, the rank of the coefficient matrix $\begin{pmatrix} x_{\widetilde{P}_1} & 1 \\ x_{P_2} & 1 \end{pmatrix}$ is 2. Therefore, there is a unique solution of $\widehat{f}_1$ and $\widehat{f}_0$ in the above equations set.

Table A1 Formulae for addition over finite fields in case $\deg(u_1) = 2,\ \deg(u_2) = 2$

Addition	$(\deg(u_1) = 2,\ \deg(u_2) = 2,\ h_2 = 0)$
Input	Two divisor classes $[u_1, v_1]$,$[u_2, v_2]$ with $u_i = x^2 + u_{i1}x + u_{i0}$ and $v_i = v_{i1}x + v_{i0}$.
Output	The divisor classes $[u', v'] = [u_1, v_1] + [u_2, v_2]$
1.	Compute $r = Res(u_1, u_2)$; $z_1 = u_{11} - u_{21};\ z_2 = u_{20} - u_{10};\ z_3 = u_{11}z_1 + z_2;\ r = z_2z_3 + z_1^2u_{10}$
2.	Compute almost inverse of u_2 modulo u_1; $inv_1 = z_1;\ inv_0 = z_3$
3.	Compute $s' = rs = ((v_1 - v_2)inv) \bmod u_1$; $w_0 = v_{10} - v_{20}$; $w_1 = v_{11} - v_{21}$; $w_2 = inv_0w_0$; $w_3 = inv_1w_1$
	$s'_1 = (inv_0 + inv_1)(w_0 + w_1) - w_2 - w_3(1 + u_{11})$; $s'_0 = w_2 - u_{10}w_3$. If $s'_1 = 0$ see below
4.	Compute $s'' = s + s'_0/s'_1$ and s_1; $w_1 = (rs'_1)^{-1};\ w_2 = rw_1;\ w_3 = s'^2_1w_1$; $w_4 = rw_2;\ w_5 = w_4^2$; and $s''_0 = s'_0w_2$
5.	Compute $l' = s''u_2 = x^3 + l'_2x^2 + l'_1x + l'_0$; $l'_2 = u_{21} + s''_0;\ l'_1 = u_{21}s''_0 + u_{20};\ l'_0 = u_{20}s''_0$
6.	Compute $u' = (s(l + h + 2v_2) - t)/u_1 = x^2 + u'_1x + u'_0$; $u'_0 = (s''_0 - u_{11})(s''_0 - z_1) - u_{10}$
	$u'_0 = u'_0 + l'_1 + h_1w_4 + (z_1 - f_4)w_5$; $u'_1 = z_1 + w_5$
7.	compute $v' = (-h - (l + v_2)) \bmod u' = v'_1x + v'_0$; $w_1 = l'_2 - u'_1;\ w_2 = u'_1w_1 - u'_0 - l'_1;\ v'_1 = w_2w_3 - v_{21} - h_1$;
	$w_2 = u'_0w_1 - l'_0;\ v'_0 = w_2w_3 - v_{20} - h_0$
8.	Return $[u', v']$
In case $s'_1 = 0$, replace 4–6 with the followling.	
$4'$.	Compute s; $inv = 1/r$; $s_0 = s'_0inv$
$5'$.	Compute $u' = (t - s(l + h + 2v_2))/u_1 = x + u'_0$; $u'_0 = f_4 - u_{21} - u_{11}s_0^2$
$6'$.	Compute $v' = (-h - (l + v_2)) \bmod u' = v'_0$; $w_1 = s_0(u_{21} + u'_0) + h_1 + v_{21};\ w_2 = u_{20}s_0 + v_{20} + h_0$;
	$v'_0 = u'_0w_1 - w_2$

Table A2 Formulae for addition over finite fields in case $\deg(u_1) = 1,\ \deg(u_2) = 2$

Addition	$(\deg(u_1) = 1,\ \deg(u_2) = 2,\ h_2 = 0)$
Input	Two divisor classes $[u_1, v_1]$,$[u_2, v_2]$ with $u_1 = x + u_{10}$ and $v_1 = v_{10}$; $u_2 = x^2 + u_{21}x + u_{20}$ and
	$v_2 = v_{21}x + v_{20}$.
Output	The divisor classes $[u', v'] = [u_1, v_1] + [u_2, v_2]$
1.	Compute $r = u_2 \bmod u_1$; $r = u_{20} - (u_{21} - u_{10})u_{10}$
2.	Compute almost inverse of u_2 modulo u_1; $\mathrm{inv} = 1/r$
3.	Compute $s' = rs = ((v_1 - v_2)inv) \bmod u_1$; $s_0 = \mathrm{inv}(v_{10} - v_{20} - v_{21}u_{10})$
4.	Compute $l = su_2 = s_0x^2 + l_1x + l_0$; $l_1 = s_0u_21; l_0 = s_0u_{20}$
5.	Compute $t = (f - v_2h - v_2^2)/u_2 = x^3 + t_2x^2 + t_1x + t_0$; $t_2 = f_4 - u_{21}; t_1 = f_3 - (f_4 - u_{21})u_{21} - u_{20}$
6.	Compute $u' = (t - s(l + h + 2v_2))/u_1 = x^2 + u'_1x + v'_0$; $u'_1 = t_2 - s_0^2 - u_{10}$; $u'_0 = t_1 - s_0(l_1 + h_1) - u_{10}u'_1$
7.	Compute $v' = (-h - (l + v_2)) \bmod u' = v'_1x + v'_0$; $v'_1 = s_0u'_1 - v_{21} - h_1 - l_1$; $v'_0 = s_0u'_0 - v_{20} - h_0 - l_0$
8.	Return $[u', v']$

Let $\widehat{\mathcal{H}}$ be a hyperelliptic curve over finite field $\mathbb{F}_q$ represented by the following Weierstrass equation: $\widehat{\mathcal{H}}$: $y^2+h(x)y = x^5+f_4x^4+f_3x^3+f_2x^2+\widehat{f}_1x+\widehat{f}_0$, Define $\widehat{P}_1 =: (x_{\widehat{P}_1},\ y_{\widehat{P}_1}), \widehat{P}_2 =: (x_{P_2}, y_{\widehat{P}_2}), \widehat{g} =: \langle\widehat{P}_1\rangle+\langle\widehat{P}_2\rangle-2\langle\infty\rangle$. Obviously $\widehat{P}_1, \widehat{P}_2 \in \widehat{\mathcal{H}}(\mathbb{F}_q)$, $\widehat{g} \in J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$. By the definition of divisor $\widehat{g}$ and α, we can find that $u_{\widehat{g}} = u_{\widetilde{g}}$ and $v_{\widetilde{g}1} = v_{\widehat{g}1}$, where $[u_{\widehat{g}}, v_{\widehat{g}}]$ is the Mumford representation of $\widehat{g}$.

By Theorem 1, $u_{\widehat{h}}$ is independent of $\widehat{f}_1, \widehat{f}_0$ and $v_{\widehat{g}0}$, so we have $u_{k\widetilde{g}} = u_{k\widehat{g}}$.

Assume that $k\widehat{g}$ is a reduced divisor. Then there exist points $\widehat{Q}_1, \widehat{Q}_2 \in \widehat{H}(\mathbb{F}_q)$ such that $k\widehat{g}$ can be uniquely represented by $k\widehat{g} = \langle\widehat{Q}_1\rangle+\langle\widehat{Q}_2\rangle-2\langle\widehat{Q}_\infty\rangle$. Putting $\widehat{Q}_i = (x_{\widehat{Q}_i}, y_{\widehat{Q}_i})$, we have $u_{k\widehat{g}} = (x-x_{\widehat{Q}_1})(x-x_{\widehat{Q}_2})$. By the fact that $u_{k\widetilde{g}} = u_{k\widehat{g}}$, $x_{\widehat{Q}_i}$ can be obtained by $u_{\widetilde{h}}$. Since $\widehat{Q}_1, \widehat{Q}_2 \in \widehat{\mathcal{H}}(\mathbb{F}_q)$, we can determine the $y_{\widehat{Q}_i}$ by the equations $\widehat{\mathcal{H}}(x_{\widehat{Q}_i}, y) = 0$, for $i = 1, 2$.

Therefore, we can find a divisor $\widehat{h} \in J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$ such that $\widehat{h} = k\widehat{g}$. This completes the proof of the theorem.

Table A3 Formulae for doubling over finite fields in case $\deg(u) = 2$

Doubling	$(\deg(u) = 2,\ h_2 = 0)$
Input	A divisor classes $[u, v]$ with
	$u = x^2 + u_1x + u_0$ and $v = v_1x + v_0$.
Output	The divisor classes $[u', v'] = [2][u, v]$
1.	Compute $\tilde{v} = (h + 2v) \bmod u = \tilde{v}_1x + \tilde{v}_0$; $\tilde{v}_1 = h_1$; $\tilde{v}_0 = h_0$
2.	Compute $r = Res(\tilde{v}, u)$; $w_0 = v_1^2; w_1 = u_1^2$; $w_2 = \tilde{v}_1^2$; $w_3 = u_1v_1$; $r = u_0w_2 + h_0(h_0 - w_3)$
3.	Compute almost inverse of $\mathrm{inv}' = r\ \mathrm{inv}$; $\mathrm{inv}'_1 = -\tilde{v}_1$; $\mathrm{inv}'_0 = \tilde{v}_0 - w_3$;
4.	Compute $t' = ((f - hv - v^2)/u) \bmod u = t'_1x + t'_0$; $t'_1 = f_3 + w_1$; $t'_0 = u_1(u_1v_1 + f_4u_1) + f_2 - w_0 - h_1v_1$
5.	Compute $s' = (t'\mathrm{inv}') \bmod u$; $w_0 = t'_0\mathrm{inv}'_0$; $w_1 = t'_1\mathrm{inv}'_1$;
	$s'_0 = w_0 - u_0w_1$; $s'_1 = (\mathrm{inv}'_0 + \mathrm{inv}'_1)(t'_0 + t'_1) - w_0 - w_1(1 + u_1)$. If $s'_0 = 0$ see below
6.	Compute $s'' = x + s_0/s_1$ and s_1; $w_1 = 1/(rs'_1)$; $w_2 = rw_1$; $w_3 = s_1'^2w_1$; $w_4 = rw_2$; $w_5 = w_4^2$
	and $s''_0 = s'_0w_2$
7.	$l' = s''u = x^3 + l'_2x^2 + l'_1x + l'_0$; $l'_2 = u_1 + s''_0$; $l'_1 = u_1s''_0 + u_0$; $l'_0 = u_0s''_0$
8.	Compute $u' = s^2 + (h + 2v)s/u + (v^2 + hv - f)/u^2$; $u'_0 = s_0''^2 + w_4h_1 - w_5f_4$; $u'_1 = -w_5$
9.	compute $v' = (-h - (l + v)) \bmod u' = v'_1x + v'_0$; $w_1 = l'_2 - u'_1$; $w_2 = u'_1w_1 - u'_0 - l'_1$;
	$v'_1 = w_2w_3 - v_1 - h_1$; $w_2 = u'_0w_1 - l'_0$; $v'_0 = w_2w_3 - v_0 - h_0$
10.	**Return** $[u', v']$
In case $s'_1 = 0$, replace 6–8 with the following.	
$6'$.	Compute s; $w_1 = 1/r$; $s_0 = s'_0w_1; w_2 = u_0s_0 + v_0 + h_0$
$7'$.	Compute $u' = (f - hv - v^2)/u^2 - (h + 2v)s/u - s^2$; $u'_0 = f_4 - s_0^2$
$8'$.	Compute $v' = (-h - (su + v)) \bmod u'$; $w_1 = s_0(u_1 + u'_0) + h_1 + v_1$; $v'_0 = u'_0w_1 - w_2$

Algorithm C1 Attack algorithm based on Model 1

Input: Hyperelliptic curve $\mathcal{H}$, the Mumford representations $[u_g, v_g]$ of a divisor $g \in J_{\mathcal{H}}(\mathbb{F}_q)$, w a parameter.

Output: Scalar k partially with a probability.

1. Inject a fault in u_g to obtain $\widetilde{u}_g = x^2 + u_{g1}x + \widetilde{u}_{g0}$.
2. Solve $\widetilde{u}_g$ to get $\widetilde{x}_1, \widetilde{x}_2$, if $\widetilde{x}_1, \widetilde{x}_2 \in \mathbb{F}_q$ goto step 3, otherwise goto step 1.
3. Solve u_g to get x_{P_1}, x_{P_2}, and obtain y_{P_i} by $y_{P_i} = v_g(x_{P_i})$, $i = 1, 2$.
4. Let $\alpha =: y_{P_1} - y_{P_2}$, for any $y_{\widehat{P}_1} \in \mathbb{F}_q$, $y_{\widehat{P}_2} =: y_{\widehat{P}_1} - \alpha$.
   4.1 Given $y_{\widehat{P}_i}$, solve equation set (2) to get $\widehat{f}_1, \widehat{f}_0$.
   4.2 Define $\widehat{\mathcal{H}} : y^2 + h(x)y = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + \widehat{f}_1x + \widehat{f}_0$.
   4.3 Let $\widehat{P}_i = (\widetilde{x}_i, y_{\widehat{P}_i})$, $\widehat{g} =: \langle\widehat{P}_1\rangle + \langle\widehat{P}_2\rangle - 2\langle\widehat{P}_\infty\rangle$.
   4.4 Obtain $n = ord(\widehat{g})$ in $J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$.
5. If all the prime factors of n are smaller than w, then
   5.1. Compute $\widetilde{h} = k[\widetilde{u}_g, v_g]$ carried out in $J_{\mathcal{H}}(\mathbb{F}_q)$ by F_{2a}
   5.2 Decompose $u_{\widetilde{h}}$, get the roots $x_{\widehat{Q}_i}(*), i = 1, 2$.
   5.3 Compute $y_{\widehat{Q}_i} = v_g(x_{\widehat{Q}_i}), i = 1, 2$.
   5.4 Let $\widehat{Q}_i = (x_{\widehat{Q}_i}, y_{\widehat{Q}_i})$, $\widehat{h} =: \langle\widehat{Q}_1\rangle + \langle\widehat{Q}_2\rangle - 2\langle\widehat{Q}_\infty\rangle$.
   5.5 Utilize Algorithm 1 on $J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$ with $(\widehat{g}, \widehat{h}, n)$ to obtain $k \bmod n$.
7. Return $(k \bmod n)$

* Actually, in step 5.2, $\widetilde{h}$ is in $J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$ and $u_{\widetilde{h}}$ has two roots in $\mathbb{F}_q$.

Table C1 Insert a flip fault in u_{g0}

Curve specification $m = 61$, $p(x) = x^{61} + x^5 + x^2 + x + 1$
$u_g = x^2 + \mathrm{0x5003d8b67eb7d6f}\,x + \mathrm{0xa8ee05ac09be989}$
$v_g = \mathrm{0x23820d5e5fa3048}\,x + \mathrm{0x074c4c18be9e74b}$
order(g) = 2658455988447243530986550320280662477
$k = 4347983749832345749 83$
$u_{\widetilde{g}} = x^2 + \mathrm{0x5003d8b67eb7d6f}\,x + \mathrm{0xa8ec05ac09be989}$
$v_{\widetilde{g}} = \mathrm{0x23820d5e5fa3048}\,x + \mathrm{0x074c4c18be9e74b}$
$u_{\widetilde{h}} = x^2 + \mathrm{0x6c814b6f0e25c161}\,x + \mathrm{0x78ffd288a0ef6e1}$
$v_{\widetilde{h}} = \mathrm{0xde20ef500589d0f}\,x + \mathrm{0x113c48bab37b6c2}$
$\widehat{\mathcal{H}} : y^2 + xy = x^5 + x^2 + \mathrm{0x21d8c2e7ea623b31}\,x + \mathrm{0x8573d4a349885611}$
$u_{\widehat{g}} = x^2 + \mathrm{0x5003d8b67eb7d6f}\,x + \mathrm{0xa8ec05ac09be989}$
$v_{\widehat{g}} = \mathrm{0x23820d5e5fa3048}\,x + \mathrm{0x41c0925d59d2b671}$
ord($\widehat{\mathrm{g}}$) = (3)(97)(151)(24593)(143827)(390271069)(43826950115759)
$u_{\widehat{h}} = x^2 + \mathrm{0x6c814b6f0e25c161}\,x + \mathrm{0x78ffd288a0ef6e1}$
$v_{\widehat{h}} = \mathrm{0xde20ef500589d0f}\,x + \mathrm{0x57b096ff5437aee1}$
Curve specification $m = 103$, $p(x) = x^{103} + x^9 + 1$
$u_g = x^2 + \mathrm{0xeee2d5c07a6bd93a0c59833ba4}\,x + \mathrm{0xa48824b71e13215936f3cfa563}$
$v_g = \mathrm{0xc7224fb356bd2cd32e4a5c14f3}\,x + \mathrm{0xfdf1b8f10539754f7b3b50e2c4}$
order(g) = 10852877190495703277390509258459145399489273609233701107 69
$k = 479837498327498354365675827957$
$u_{\widetilde{g}} = x^2 + \mathrm{0xeee2d5c07a6bd93a0c59833ba4}\,x + \mathrm{0xa48824b71e17215936f3cfa563}$
$v_{\widetilde{g}} = \mathrm{0xc7224fb356bd2cd32e4a5c14f3}\,x + \mathrm{0xfdf1b8f10539754f7b3b50e2c4}$
$u_{\widetilde{h}} = x^2 + \mathrm{0x9a68f0815dac0c2fa0970ff2f6}\,x + \mathrm{0x3d71cd1bbaba4b3feec04fca4}$
$v_{\widetilde{h}} = \mathrm{0x86194e0eb0df241dcd760da6c5}\,x + \mathrm{0xb6dceb387e5ce3c14b19a23124}$
$\widehat{\mathcal{H}} : y^2 + xy = x^5 + x^2 + \mathrm{0x8cebb70059930116e9beff11c1}\,x$ $+\mathrm{0x899772507fed8b3d86a781fa03}$
$u_{\widehat{g}} = x^2 + \mathrm{0xeee2d5c07a6bd93a0c59833ba4}\,x + \mathrm{0xa48824b71e17215936f3cfa563}$
$v_{\widehat{g}} = \mathrm{0xc7224fb356bd2cd32e4a5c14f3}\,x + \mathrm{0x55aa3c1832342b28b1d5603f57}$
ord($\widehat{\mathrm{g}}$) = (2)(59021)(1112923871)(8925786237751)(31532716137894221) (556287399183096149)
$u_{\widehat{h}} = x^2 + \mathrm{0x9a68f0815dac0c2fa0970ff2f6}\,x + \mathrm{0x3d71cd1bbaba4b3feec04fca4}$
$v_{\widehat{h}} = \mathrm{0x86194e0eb0df241dcd760da6c5}\,x + \mathrm{0x1e876fd14951bda681f792ecb7}$
Curve specification $m = 113$, $p(x) = x^{113} + x^9 + 1$
$u_g = x^2 + \mathrm{0xc2b96348cc58e038b71178a9a38b}\,x + \mathrm{0x3b358cf39d80854ad0b4d8ed5f43}$
$v_g = \mathrm{0xa6d4259ef3709c31246fdf8cce661}\,x + \mathrm{0x812bd9b8364583ca9abe1ddac461}$
order(g) = 539198933343012787158232976738412307606428027150190435497641933683 81
$k = 479837498327498354365675827957$
$u_{\widetilde{g}} = x^2 + \mathrm{0xc2b96348cc58e038b71178a9a38b}\,x + \mathrm{0x3b358cf19d80854ad0b4d8ed5f43}$
$v_{\widetilde{g}} = \mathrm{0xa6d4259ef3709c31246fdf8cce661}\,x + \mathrm{0x812bd9b8364583ca9abe1ddac461}$
$u_{\widetilde{h}} = x^2 + \mathrm{0x503da1588c8eab09118a2d42c5fd1}\,x + \mathrm{0xa15efb97f8482faeeff99f5fa342}$
$v_{\widetilde{h}} = \mathrm{0xf71c9cf9d27203907823b259afee1}\,x + \mathrm{0xe63f7117b897820c334314f738471}$
$\widehat{\mathcal{H}} : y^2 + xy = x^5 + x^2 + \mathrm{0x2d608dd43ab5fd4c9abba4b1ae95}\,x$ $+\mathrm{0xd19a342cd5eed8b4c588d6a999f}$
$u_{\widehat{g}} = x^2 + \mathrm{0xc2b96348cc58e038b71178a9a38b}\,x + \mathrm{0x3b358cf19d80854ad0b4d8ed5f43}$
$v_{\widehat{g}} = \mathrm{0x812bd9b8364583ca9abe1ddac461}\,x + \mathrm{0x7d5858cfe10a2c2eb7d341d909971}$
ord($\widehat{\mathrm{g}}$) = (5)(503)(12046651)(183064547)(5637681901967)(24099893265761) (715524936956239982156 29)
$u_{\widehat{h}} = x^2 + \mathrm{0x503da1588c8eab09118a2d42c5fd1}\,x + \mathrm{0xa15efb97f8482faeeff99f5fa342}$
$v_{\widehat{h}} = \mathrm{0xf71c9cf9d27203907823b259afee1}\,x + \mathrm{0x3db30c46aaed3213a0ff8aa2ffb61}$

Appendix C Attack algorithm by injecting a fault in u_{g0}

By fault Model 1, we can get $[\widetilde{u}_g, v_g]$ by injecting a fault in u_{g0}, where $\widetilde{u}_g = x^2 + u_{g1}x + \widetilde{u}_{g0},\ v_g = v_{g1}x + v_{g0}$.

Let $[u_{\widetilde{h}}, v_{\widetilde{h}}]$ be the Mumford representation of divisor $\widetilde{h} = k[\widetilde{u}_g, v_g]$. The computation should be carried out in $J_{\mathcal{H}}(\mathbb{F}_q)$ by applying F_{2a}.

The following result provides an attack method on Model 1.

Theorem C1. Let $\mathcal{H}$ be a hyperelliptic curve of genus 2 defined over a finite field $\mathbb{F}_q$ of characteristic 2 of form (1), and let $[u_g, v_g]$ be the Mumford representation of a divisor $g \in J_{\mathcal{H}}(\mathbb{F}_q)$. Let $[\widetilde{u}_g, v_g], [u_{\widetilde{h}}, v_{\widetilde{h}}]$ be defined as above. Then there exists a hyperelliptic curves $\widehat{\mathcal{H}}$ defined over $\mathbb{F}_q$ and divisors $\widehat{g}, \widehat{h} \in J_{\widehat{\mathcal{H}}}(\mathbb{F}_q)$ satisfying $u_{\widetilde{h}} = u_{\widehat{h}}$ and $\widehat{h} = k\widehat{g}$. Moreover $u_{\widehat{g}} = x^2 + u_{g1}x + \widetilde{u}_{g0},\ v_{\widehat{g}} = v_{g1}x + v_{\widehat{g}0}$.