A Correlation Power Analysis Attack against Tate Pairing on FPGA*

Weibo Pan and William P. Marnane

Dept. of Electrical and Electronic Engineering
University College Cork, Cork, Ireland
{weibop,liam}@rennes.ucc.ie

Abstract. Pairings on elliptic curves are deeply researched and used in applications such as identity based schemes. Recently there have been several hardware implementations of the Tate Pairing. Along with the algorithms, their security has to be considered. This paper presents a correlation power analysis (CPA) attack against a Tate pairing implementation. Real power traces are taken from the FPGA implementation. The experimental result shows a successful attack.

Keywords: Tate pairing, CPA, FPGA.

1 Introduction

Pairing based cryptography is a new type of public-key cryptographic scheme based on Elliptic Curve Cryptography (ECC). ECC is efficient because it achieves the same security level as 1024 bit key RSA cryptography, with only a 163 bit key [8]. Pairings have the properties of bilinearity and non-degeneracy which is of interest for many applications. Cryptographic schemes based on the bilinear pairings have been developed to exploit Miller's algorithm [9]. Among the popular pairings, the Tate pairing has proved to be the most efficient in all fields for frequently used key sizes [10]. There have been many algorithms implementing the Tate pairing [12,15]. Barreto et al. [12] developed a fast algorithm of the Tate pairing on supersingular elliptic curve over finite fields of characteristic two ($GF(2)$). Shu et al. [7] developed a fast hardware implementation of the algorithm, while our work [6] discussed different hardware designs for implementing the algorithm on a Xilinx FPGA.

In implementating a cryptosystem, security as well as efficiency is a factor that has to be considered. An attacker can recover the secret information by monitoring the side channel informations such as power consumption[1]. Thus along with the pairing algorithms, side channel analysis (SCA) has become popular. The implementations of cryptosystems might be insecure against SCAs if not implemented carefully. Whelan et al. [4] and Kim et al. [5] investigated the possibility of SCA, including simple, differential and correlation power analysis

* This material is based upon works supported by the Science Foundation Ireland under Grant No. [SFI/ 08/RFP/ENE1643].

A. Koch et al. (Eds.): ARC 2011, LNCS 6578, pp. 340–349, 2011.

(SPA, DPA and CPA) against practical pairing algorithm. In 2005, Page and Vercauteren [14] presented the first side channel analysis of Duursma-Lee's algorithm [15] for characteristic three. This paper presents an implementation of a CPA attack on an FPGA implementation of the Tate pairing [6] using the pairing algorithm developed by Shu et al. [7].

2 Tate Pairing Algorithm

2.1 Tate Pairing Over $GF(2^m)$

Let E be an elliptic curve over a finite field of characteristic two: $E(GF(2^m))$: $Y^2+Y = X^3+X+g$, where $g \in \{0,1\}$. A point on this elliptic curve is represented as a pair of elements $(x,y) \in GF(2^m)$ which satisfy the curve equation.

The Tate pairing in cryptosystems, is generally represented by $e_l(P;Q)$, where P and Q are points of order l on curve $E(GF(2^m))$, m=163 in this work. It evaluates to a point over the extended field $GF(2^{4m})$. A closed formula of the Tate pairing[7] implemented in [6] is given in Algorithm 1.

Algorithm 1. Algorithm for computing Tate pairing

Input: $P = (\alpha, \beta), Q = (x, y)$ **Output**: $C = e_l(P;Q)$

1: $C \leftarrow 1$
2: $u \leftarrow x^2 + y^2 + g + \frac{m-1}{2}, v \leftarrow x^2 + 1, \alpha \leftarrow \alpha^4, \beta \leftarrow \beta^4, \gamma \leftarrow \alpha v$
3: **for** $i = 0$ to $m - 1$
4: $\quad A(t) \leftarrow \gamma + u + \beta + (\alpha + v + 1)t + (\alpha + v)t^2$
5: $\quad C \leftarrow C^2 * A(t)$
6: $\quad u \leftarrow u + v, v \leftarrow v + 1, \alpha \leftarrow \alpha^4, \beta \leftarrow \beta^4, \gamma \leftarrow \alpha v$
7: **endfor**
8: $C(x) = C(x)^{2^{2m}-1}$
9: **return** $C(x)$

2.2 Design of Tate Pairing Components Over $GF(2^m)$

As shown in Algorithm 1, there are additions, squarings, multiplications and division in the algorithm, among which the division appears only once in the final exponentiation in step 8.

Addition is the most basic operation in the algorithm, performed as per $c(x) = a(x)+b(x)$. It adds two elements on $GF(2^m)$ through an XOR chain, taking only one clock cycle. Thus addition and subtraction are equivalent in $GF(2^m)$, and it is noted that there exists $a + a = 0$ in $GF(2^m)$.

Squaring in $GF(2^m)$ is represented by $c(x) = a^2(x) \bmod f(x)$ where $f(x)$ is the irreducible polynomial of $GF(2^m)$. A bit-parallel squaring architecture introduced in [17] is applied in this design. Rather than inputing two same elements into a multiplier, the specific squarer simply interleaves the input with zeroes, followed by a $2m - 1$ to m bit reduction block. It takes only one clock cycle to compute a squaring.

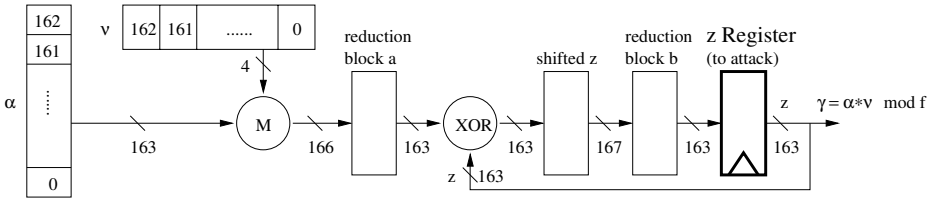

Fig. 1. $GF(2^m)$ Digit Serial Multiplier applied in Tate pairing

Division in $GF(2^m)$ is represented by $c(x) = a(x)/b(x) \bmod f(x)$. A field division using an architecture based on the Extended Euclidean Algorithm introduced in [18] is applied in the design of this paper. It takes $2m$ clock cycles to compute a division. Such time consuming operation appears only once in the final exponentiation of the Tate pairing in step 8 of Algorithm 1.

Multiplication in $GF(2^m)$ is represented as per $c(x) = (a(x) * b(x)) \bmod f(x) = (\sum_{i=0}^{m-1} a_i x^i * \sum_{j=0}^{m-1} b_j x^j) \bmod f(x)$. There are 7 multiplications in step 4-5 in Algorithm 1. These multiplications take most of the operation time of the Tate pairing algorithm. The architecture of Digit-Serial Multiplication (DSM) applied in the Tate pairing design introduced by Hankerson et al. [16] is shown as in Fig.1.

The DSM takes elements α and v as input, and outputs the products modulo the fixed irreducible polynomial $f(x)$. It deals with d bits of the input element in every iteration and takes $n = \frac{m}{d}$ clock cycles to finish the operation. A larger digit size d makes the multiplier larger while reduces the calculation time in the same time. The trade-offs between area and computation time was discussed in [6].In this paper, we pick digit size $d = 4$ as an example.

2.3 Tate Pairing Architecture Over $GF(2^m)$

There are 7 multipliers in step 4-6. A designer can schedule these 7 multiplications by putting in different number of multipliers [6]. A simplest way is to use only 1 multiplier, and have all the 7 multiplications operated in serial. The architecture of this design is shown in Fig.2. While putting in more multipliers, the multiplications can be operated in parallel. As there are 7 multiplications in each iteration of the *for* loop of Algorithm 1, applying 7 multipliers pipelined in the design and have all the multiplications operated in parallel is the fastest design, with maximum area as shown in Fig.3. In this paper, we present the minimum area design using only 1 multiplier and the maximum area design using 7 multipliers. The attack and correlation result will be presented in section 4.

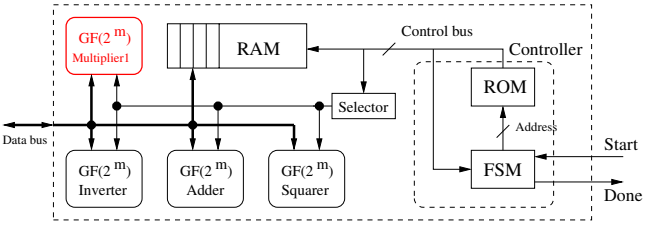

Fig. 2. Tate pairing implementation architecture over $GF(2^m)$, 1 multiplier

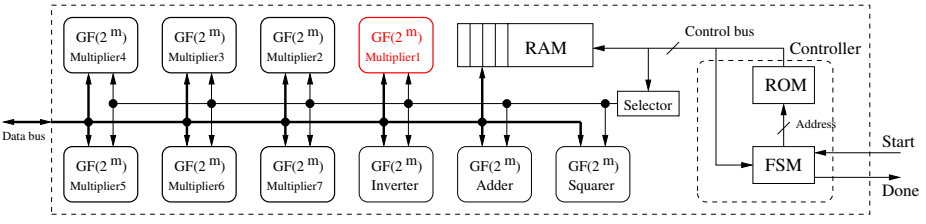

Fig. 3. Tate pairing implementation architecture over $GF(2^m)$, 7 multipliers

2.4 Tate Pairing in Identity Based Encryption

Pairings are used in Identity Based Encryption (IBE)[11] as a public key scheme. In IBE the Tate pairing $e_l(P;Q)$ is used to generate cypher text C with two inputs: a public key and a secret text. Since Tate pairing has the property that $C = e_l(P;Q) = e_l(Q;P)$, either P or Q can be input as the secret text. This means that the attacker can choose which input to be the secret text[4]. In the design of this paper, we assume that P is public and Q is the secret text to attack. Due to point compression, the Elliptic Curve $E(GF(2^m))$ restricts the two coordinates of a point $Q(x,y)$, knowing one coordinate leads to knowing the other. Thus we only need to focus on the x coordinate of the secret input $Q(x,y)$. The detail of how the coordinate x is revealed is introduced in Section 4.

3 CPA Model and Attack

Based on Hamming Distance (HD) model, CPA reveals the relationship between hardware power consumption and the intermediate values of the operations.

3.1 Hamming Distance Model

HD model is based on Hamming Weight (HW)[1] model. In a hardware implementation of a cryptosystem, an m-bit binary data word D is represented as $D = \Sigma_{j=0}^{m-1} d_j 2^j$, where $d_j \in \{0,1\}$. Its HW is the number of elements that are equal to 1, i.e. $H(D) = \Sigma_{j=0}^{m-1} d_j$. This is the HW model based upon which many power analyses attacks on software implementations are built. Since H is an

integer between 0 and m, if the data words D are independent and uniformly distributed, D has an average HW $\mu_H = m/2$ and a variance $\sigma_H^2 = m/4$.

The HD model [2] assumes that the side channel information leaked from a system depends on the number of bits switching from one state to the other and is more appropriate for hardware implementations. The basic HD model is:

$$W = aH(D \oplus R) + noise, \tag{1}$$

where *noise* encloses switching and electrical noise, D is the current state and R is next state, a is a scalar gain between W the power consumption and H the HW of $(D \oplus R)$. $H(D \oplus R)$ here represents the number of bits switched between register states D and R. This is called the HD between D and R. In this model, R is usually targeted by the attacker.

3.2 Correlation Power Analysis

The basic principle of CPA [2] is that there exists a relationship between HD of two register states and the measurable power consumption. The correlation factor between HD and consumed power, is used to tell whether the HD model fits the real power consumption or not. It is the covariance between the two variables H and W normalized by the product of their standard deviations. Assuming the noise is of Gaussian distribution, with the HD model, we have:

$$\rho_{WH} = \frac{cov(W, H)}{\sigma_W \sigma_H} \tag{2}$$

This relationship shows that ρ_{WH} helps determine the next state R. Assuming the variance of noise tends to 0, if the hypothetical value of R is correct, the value ρ_{WH} tends to ± 1 at the correlated point. In experiments, if an attacker predicts the correct secret value at the target clock cycle, there will be a high correlation value at the related point, otherwise the correlation values tend to 0.

4 CPA against Tate Pairing and Result Analysis

In this paper, the design in [6] is implemented on a SASEBO-GII board [13] which was designed for side channel attacks. The algorithm is running at speed of 24 MHz. A number $N = 1000$ of power traces of the operation are taken, measured using a 1Ω resistor at the V_{CC} side. By putting in different number of multipliers, there are several different schedules of designing the architecture of the Tate pairing algorithm. Here we implement two different design architectures. The simple design contains only 1 multiplier in the architecture and the most complicated design has 7 multipliers in parallel in the architecture.

As mentioned in Section 2, the x coordinate of secret input Q tells the secret. To reveal x, we pick the multiplication $\gamma \leftarrow \alpha v$ in step 6 of Algorithm 1. It involves the coordinates α and v, where v relates closely with secret coordinate value x. Once the secret value v is known, the coordinate values x can be achieve by XOR "1" $\in GF(2^m)$ followed by a square root operation [3], and thus the secret text Q.

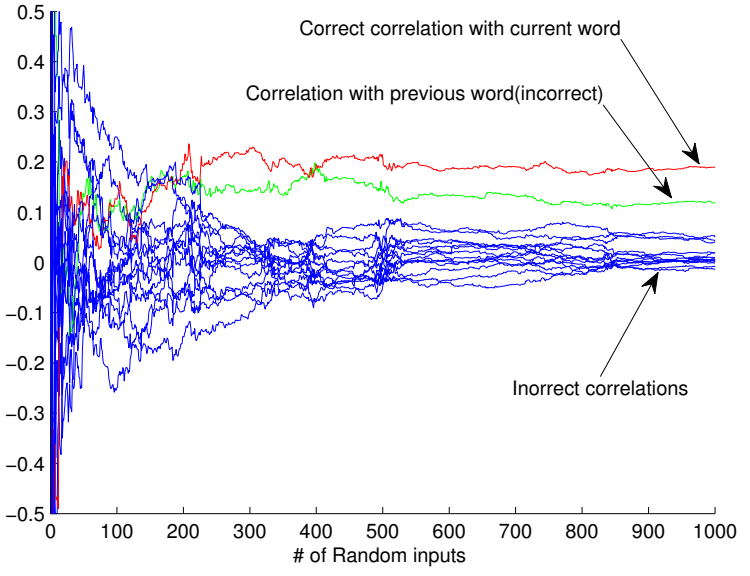

Fig. 4. CPA against Tate pairing at target time point, 1 mult

4.1 CPA against Single Multiplier Design

In the architecture of Fig.2 only one digit serial multiplier [16] is used. All multiplications in the Pairing algorithm are operated in serial using the only multiplier. The structure of the multiplier is shown in Fig.1. With two inputs α and v, the multiplier calculates the product of α and d=4 bits of v, called a 4-bit word of v, in every iteration. The targeted register "z Register" stores the product of α and a 4-bit word of v and is updated in every iteration. The multiplication $\gamma \leftarrow \alpha v$ finishes in $n = \lceil \frac{m}{d} \rceil = \lceil \frac{163}{4} \rceil = 41$ clock cycles.

Since v contains the secret and α is known by the attacker, we input $N = 1000$ random plaintexts as α, and collect the power traces of the operations. The most significant bits (MSBs) are first dealt with in the multiplication. The multiplier deals with 4 bits of input v in most iterations except for the first iteration. In the first iteration, since the targeted field size is $m = 163$, the multiplier deals with one bit '0' as the most significant bit and the first 3 bits of v as the least significant bits.

For each clock cycle of the attack, we do the following steps:

1. for each of the $N = 1000$ public inputs, generate 2^4 hypothetical values of the 4 bits of input v by traversing all possible values from "0000" to "1111".
2. generate hypothetical values of the target register "z Register" in last and the current clock cycle.
3. calculate the Hamming Distance of the value in "z Register" between last and the current clock cycle.
4. calculate the correlations between the hypothetical values and the measured power traces.

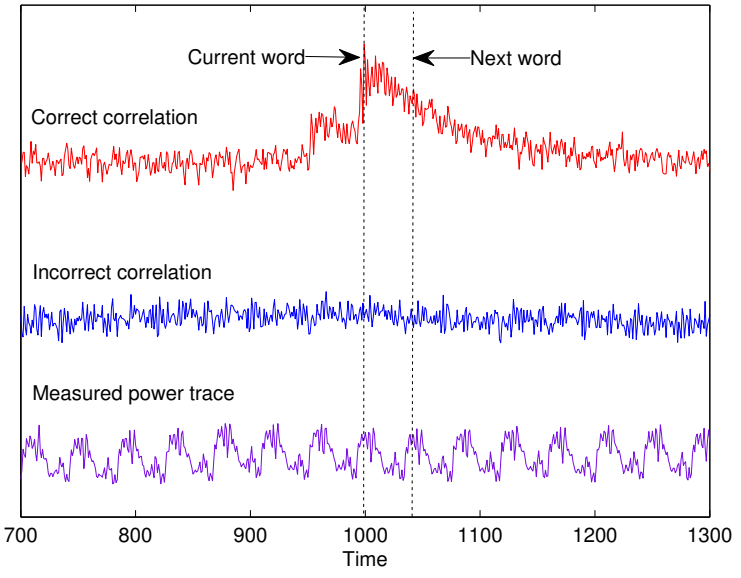

Fig. 5. CPA against Tate pairing power traces, 1 mult

Fig.4 shows the correlation attack against the target multiplication in the Tate pairing algorithm. Assume the previous 4-bit words of v "(0)101 0111 1000" are correctly guessed. The hypothetical values of the next 4-bit word is between "0000" - "1111". Fig.4 shows the correlation of all 2^4 predictions at the target time point. It is obvious that a correlation value comes much higher than others, which indicates the correct prediction "1001" of current 4-bit word of v. The second highest correlation value, which is about half the highest, indicates the effect remained from the previous 4-bit word of v (1000).

For better analysis, we pick traces of all time points rather than the target time point. Fig.5 shows the correct correlation and an incorrect correlation, compared with the real measured power trace sample. In the correct correlation trace, there is a peak at the target time point. While the incorrect correlation trace corresponds to noise. The correct correlation peak doesn't drop instantly after the target time point, but decays in the next 2 clock cycles. Thus every correct prediction of the input affects the next clock cycle's prediction, with a correlation value of about half its peak value. By doing the same attack shown in Fig.5 $n = \lceil \frac{m}{d} \rceil = \lceil \frac{163}{4} \rceil = 41$ times, the secret text v is determined, and thus the secret input $Q(x, y)$. The architecture of the complicated design with 7 multipliers is shown in

4.2 CPA against Maximum Area Design of Tate Pairing

The architecture of the 7 multiplier design is shown as in Fig.3. In this design, all the 7 multipliers are parallel and pipelined in each iterations of the *for* loop of Algorithm 1. What's different from the single multiplier design is that all the

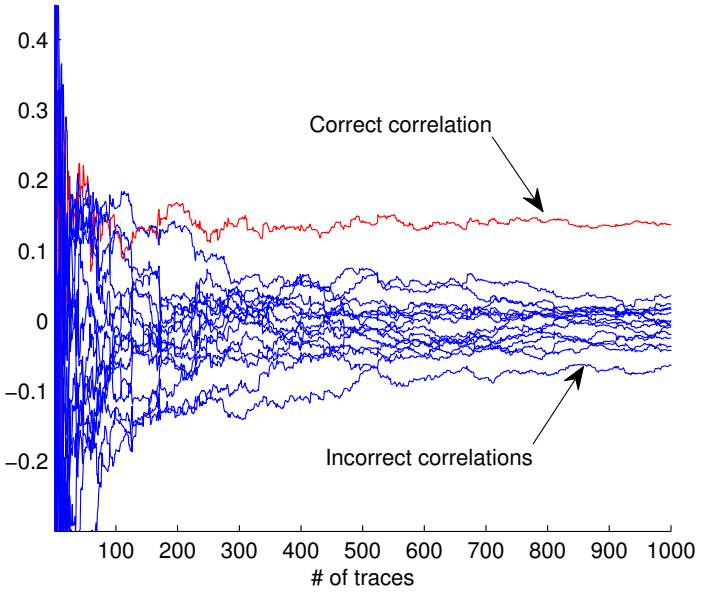

Fig. 6. CPA against Tate pairing at target time point, 7 mults

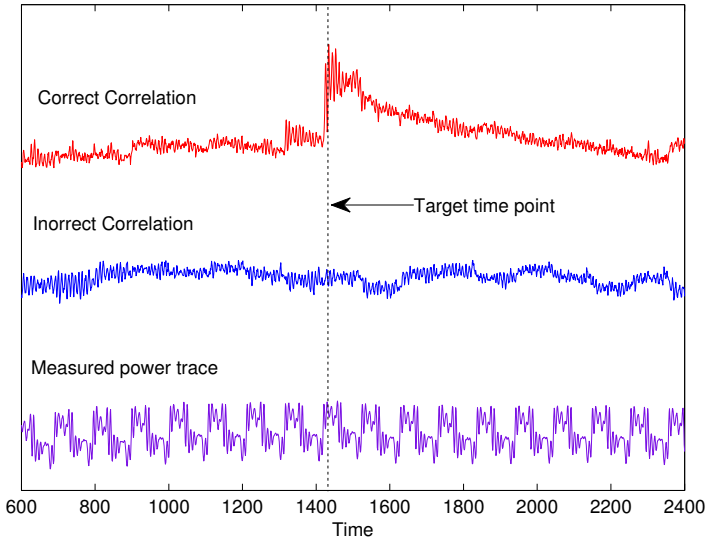

Fig. 7. CPA against Tate pairing power traces, 7 mults

7 multiplications in each iteration of the *for* loop are operated in the same time. So the FPGA consumes more power in each clock cycle, and thus more noise. However the effect of such noise will be removed in the power analysis because its correlation value tends to 0 after taking N=1000 power trace samples.

For the multiple multipliers design, we do the same as what we did to the single multiplier design. Fig.6 shows the correlation attack against the target multiplication in the Tate pairing algorithm. As can be seen from Fig.6, at the target time point, it is obvious that the correct correlation value is higher than the incorrect ones. The target time point is different from the single multiplier design because all 7 multiplications are operated in the same time. From Fig.7 we see that the correlation peak is still sharp which means the noise introduced by putting in 6 more multipliers doesn't affect the correlation result.

5 Conclusion

In this paper we presented the first correlation power analysis attack against an FPGA implementation of the Tate pairing. The multiplication over $GF(2^m)$ in the pre-computation step was targeted. We performed this attack by correlating the predicted value (i.e. the hypothetical value of an intermediate value in the multiplication generated by partial input) with the real measured power traces. A single multiplier and a maximum area design with 7 multipliers of the Tate pairing have been targeted. The result shows a peak in the trace of the correct hypothesis value at the corresponding period. Hence the design of the pairing algorithms should be carefully considered.

References

1. Kocher, P.C., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
2. Brier, E., Clavier, C., Olivier, F.: Correlation Power Analysis with a Leakage Model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 135–152. Springer, Heidelberg (2004)
3. Fong, K., Hankerson, D., López, J., Menezes, A.: Field inversion and point halving revisited. IEEE Transactions on Computers 2004 53, 1047–1059 (2004)
4. Whelan, C., Scott, M.: Side channel analysis of practical pairing implementations: Which path is more secure? In: Nguyên, P.Q. (ed.) VIETCRYPT 2006. LNCS, vol. 4341, pp. 99–114. Springer, Heidelberg (2006)
5. Kim, T.H., Takagi, T., Han, D.-G., Kim, H.W., Lim, J.: Side Channel Attacks and Countermeasures on Pairing Based Cryptosystems over Binary Fields. In: Pointcheval, D., Mu, Y., Chen, K. (eds.) CANS 2006. LNCS, vol. 4301, pp. 168–181. Springer, Heidelberg (2006)
6. Pan, W., Marnane, W.: A Reconfigurable Implementation of the Tate Pairing Computation over *GF* (2^m). In: Sirisuk, P., Morgan, F., El-Ghazawi, T., Amano, H. (eds.) ARC 2010. LNCS, vol. 5992, pp. 80–91. Springer, Heidelberg (2010)
7. Shu, C., Kwon, S., Gaj, K.: FPGA Accelerated Tate Pairing Based Cryptosystems over Binary Fields. In: Proceedings of the IEEE International Conference on Field Programmable Technology 2006, pp. 173–180. IEEE, Los Alamitos (2006)
8. Gupta, V., Gupta, S., Chang, S.: Performance analysis of elliptic curve cryptography for SSL. In: Proceedings of the 1st ACM Workshop on Wireless Security, pp. 87–94. ACM Press, New York (2002)

9. Miller, V.S.: Short Programs for functions on Curves. unpublished manuscript (1986)
10. Granger, R., Page, D., Smart, N.P.: High Security Pairing-Based Cryptography Revisited. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 480–494. Springer, Heidelberg (2006)
11. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
12. Barreto, P.S.L.M., Kim, H.Y., Lynn, B., Scott, M.: Efficient algorithms for pairing-based cryptosystems. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 354–369. Springer, Heidelberg (2002)
13. Research Center for Information Security, National Institute of Advanced Industrial Science and Technoloty. Side-channel Attack Standard Evaluation Board SASEBO-GII Specification. Version 1.0 (2009)
14. Page, D., Vercauteren, F.: Fault and Side-Channel Attacks on Pairing Based Cryptography. IEEE Transactions on Computers 55(9), 1075–1080 (2006)
15. Duursma, I.M., Lee, H.-S.: Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 111–123. Springer, Heidelberg (2003)
16. Hankerson, D., Menezes, A., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer, Heidelberg (2004)
17. Mastrovito, E.D.: VLSI Architectures for Computation in Galois Fields. PhD thesis, Dept. Electrical Engineering, Linkoping University, Linkoping, Sweden (1991)
18. Shantz, S.C.: From Euclids GCD to Montgomery Multiplication to the Great Divide. Tech. Rep. SMLI TR-2001-95, Sun Microsystems, pp. 1–10 (2001)