Securing pairing-based cryptography on smartcards

Mustapha Hedabou

Department of Computer Science,
ENSA de Safi,
Route Sidi Bouzid BP 63, Safi, Morocco
Fax: +212-524-66-80-12
E-mail: mhedabou@gmail.com

Abstract: Research efforts have produced many algorithms for fast pairing computation and several efficient hardware implementations. However, side channel attacks (SCAs) are a serious threat on hardware implementations. Previous works in this area have provided some costly countermeasures against side channel attacks. In this paper, we propose a new countermeasure based on the isomorphism classes in order to secure pairing computation on elliptic curves defined over $\mathbb{K}$ with $Char(\mathbb{K}) \neq 2, 3$. The proposed technique requires only eight additional field multiplications which is negligible compared to the cost of the other known countermeasures.

Keywords: pairing computation; countermeasures; isomorphism classes; side channel attacks; SCAs.

Reference to this paper should be made as follows: Hedabou, M. (2012) 'Securing pairing-based cryptography on smartcards', *Int. J. Information and Computer Security*, Vol. 5, No. 1, pp.68–76.

Biographical notes: Mustapha Hedabou received his MSc in Mathematics from the University of Paul Sabatier, Toulouse, France. In 2006, he received his PhD in Computer Science from INSA de Toulouse, France. He is currently an Associate Professor at ENSA de Safi, University of Marrakech in Morroco. His area interest covers information security, public key cryptography based on elliptic curves and identity-based cryptography.

1 Introduction

The first identity-based scheme was proposed by Boneh and Franklin (2003). Their scheme uses the bilinear functions defined on elliptic curve such as Weil pairing and Tate pairing. A lot of work has focused on the improvement of Tate and Weil pairing computation. All these algorithms are based on Miller's (2004) algorithm. Recently, new methods like Eta and Ate pairing (Duursma and Lee, 2003; Barreto et al., 2007; Granger et al., 2007) have been proposed.

Software and hardware implementations of pairings have also been presented. Indeed, pairings have implemented on FPGAs (Shu et al., 2006) and smart cards (Scott et al., 2006). However, side channel attacks (SCAs) (Kocher et al., 1999) are a serious threat on such cryptographic applications. To our knowledge, only few techniques for securing pairing-based cryptosystems against SCAs are known. These countermeasures include the randomisation of the private data and the point blinding in the bilinear pairing which are proposed by Page and Vercauteren (2004) and Scott (2005). The use of the projective coordinates have also been proposed by Kim et al. (2006) for securing the Eta pairing. In spite of their efficiency, complexity analysis shows that the cost of all these countermeasures is very high.

In this paper, we propose a new countermeasure for securing pairing-based cryptosystems against SCAs on elliptic curves defined over $\mathbb{K}$ with $Char(\mathbb{K}) \neq 2, 3$. The proposed countermeasure uses the isomorphism classes of an elliptic curve to randomise the secret information. The cost of the proposed countermeasure is only eight field multiplications.

2 Pairing computation

Let $E(\mathbb{F}_q)$ be an elliptic curve defined over a field $\mathbb{F}_q$. Let l be a positive integer, co-prime to q, such that $E(\mathbb{F}_q)$ contains a point of order l. In cryptographic implementations, l is usually taken to be a large prime such that $l \mid \#E(\mathbb{F}_q)$. Let k be the smallest integer satisfying $l \mid q^k - 1$. This value k, is the embedding degree of the curve with respect to l. The Tate pairing is defined in terms of rational functions over points of an elliptic curve evaluated in a divisor. Let $P \in E(\mathbb{F}_q)[l]$, then $l(P) - l(O)$ is a principal divisor. So there is a rational function $f_p \in \mathbb{F}_{q^k}(E)$ with $div(f_P) = l(P) - l(O)$. Let $Q \in E(\mathbb{F}_{q^k})[l]$ be a point with coordinates in $\mathbb{F}_{q^k}$; then we construct a divisor D_Q such that $D_Q \sim (Q) - (O)$. D_Q should be chosen so that its support is disjoint from the support of the divisor of f_p. Now let μ_l be the subgroup of l^{th} roots of unity in $\mathbb{F}^*_{q^k}$. The Tate pairing is defined as follow:

$$e_l : E(\mathbb{F}_q)[l] \times E(\mathbb{F}_{q^k})[l] \longrightarrow \mu_l$$

$$(P, Q) \longrightarrow f_p(D_Q)^{\frac{q^k-1}{l}}$$

Miller's (2004) algorithm provides a way to compute the Tate pairing as well as Weil pairing. Let $f()$ be the function that calculates the line functions required by Miller's algorithm, and returns a value in $\mathbb{F}_{q^k}$. This function in turn requires function $A.add(B)$ which adds the elliptic curve points $A = A + B$ using standard methods, and returns the slope of the line joining A and B.

In this paper, we will use the BKLS algorithm (Barreto et al., 2002) for the the computation of the Tate pairing over the elliptic curve $E(\mathbb{F}_q)$. The embedding degree is 2 and $q = 3 \mod 4$. The point Q is handled as a point on the twisted curve $E'(\mathbb{F}_q)$. We denote by $\overline{m}$ the conjugate of an element m and by n_i a bit from the binary representation of an integer n. The computation of the function $f()$ is done as follows.

Algorithm 1 Function $f()$

Input: an elliptic curve $E(\mathbb{F}_q) : y^2 = x^3 + ax + b$, points A, B in $E(\mathbb{F}_q)$ and a point Q
Output: $f(A, B, Q)$
1. let $A = (x_i, y_i)$, $Q = (x_q, y_q)$
2. $\lambda_i = A.add(B)$
3. return $y_C - \lambda_i(x_Q + x_C) - iy_Q$,

where (x_C, y_C) are the coordinates of the sum of A and B, i.e., $(x_C, y_C) = A + B$. Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two points neither being the point at infinity, the slope $\lambda_i = P_1.add(P_2)$ is defined by:

$$\lambda_i = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} \text{ if } P_1 \neq P_2 \\ \\ \frac{3{x_1}^2 + a}{2y_1} \text{ if } P_1 = P_2 \end{cases}$$

Algorithm 2 Computation of the Tate pairing

Input: an elliptic curve points $E(\mathbb{F}_q) : y^2 = x^3 + ax + b$, a point P of prime order l and a point Q on the twisted curve $E'(\mathbb{F}_q)$
Output: $e_l(P, Q)$
1. $m = 1$
2. $A = P$
3. $n = l - 1$.
4. for $i \leftarrow \lfloor log(l) \rfloor - 1$ down to 0 do
5. $m \leftarrow m^2.f(A, A, Q)$
6. if $n_i = 1$ then $m \leftarrow m \cdot f(A, P, Q)$
7. end for.
8. $m \leftarrow \frac{\overline{m}}{m}$
9. return $m^{\frac{p+1}{l}}$

3 SCAs and previous countermeasures

For many years, researches related to side channel attacks (Kocher et al., 1999) are focused on scalar multiplication on elliptic curves. Page and Vercauteren (2004) proposed an attack on Tate pairing computation based on Duursma and Lee (2003) algorithm over supersingular elliptic curves in characteristic three. Scott (2005) showed that the SCAs can also be efficient on Tate, Ate and Eta pairing. Kim et al. (2006) claimed that the Eta pairing computation on supersingular curves of characteristic 2 may succumb to SPA, DPA and timing attacks. However, all of these attacks are theoretical. Recently, Whelan et al. (2008) carried out successfully a CPA attack on Eta pairing computation and presented a practical analysis that validates this attack.

When a secret data influences the execution path and the sequence being executed, then SCAs can be a significant threat. In pairing-based cryptography, the secret data may affect the behaviour of the underlaying finite field operations. The power analysis of field operations can then reveal some secret information that may be exploited by side channel attacks. Indeed, during the pairing computation a field operation like $y.r$,

where y is a secret data and r is a known value, is performed. Since a SCA attack can be carried out against field multiplication a SCA attack can also be carried out successfully against pairing computation. However, for some kind of pairing, like Eta pairing in characteristic two, a more complicated operation is computed. In this case, the multiplication $a(b+r)$ where a and b are unknown is performed. Kim et al. (2006) claimed that the power analysis is harder but is still possible.

In Algorithm 2, the leakage of information occurs during the computation of the function $f()$ which involves the coordinates of the secret point. If we assume that Q is secret and P is public, a field operation like $a(b+c)$ where a and c are known and b is secret is performed. According to Scott (2005), this is the most vulnerable path. The estimated power consumption of a portion of the computation of $f()$ is calculated given hypothesis for the value of the secret coordinates x_Q and y_Q and for n known values x_P and y_P. On the other hand, if P is secret and Q is public, which is the most difficult path (Scott, 2005), a field operation like $a(b+c)$ where a and c are secret and b is known is performed. A power analysis attack can also be carried out in this case (Kim et al., 2006).

As mentioned above, for thwarting SCA on pairing-based cryptosystems many countermeasures have been proposed. Page and Vercauteren (2004) used the bilinearity to randomise the private data, i.e., $e_l(P,Q) = e_l(sP, tQ)^{\frac{1}{st}}$ where s and t are random integers. Furthermore, the exponentiation to the power $\frac{1}{st}$ can be removed by selecting s and t satisfying $st = 1 \mod l$, where l is the order of the underlying elliptic curve. They also presented a method for blinding the input point by computing $e_l(P,Q) = e_l(P, Q+R).e_l(P,R)^{-1}$. Scott (2005) suggested to randomise each operation involving the secret data by multiplying all intermediate variables by a random element of $\mathbb{F}_q$. For example, in the case of Tate pairing the function $f()$ computes $r \cdot y_C - \lambda_i(r \cdot x_Q + r \cdot x_C) - r \cdot iy_Q$ for a new random integer $r \in \mathbb{F}_q$ at each step. Finally, Kim et al. (2006) proposed to use the projective coordinates $(r.x_Q, r.y_Q, r)$ instead of the affine coordinates (x_Q, y_Q). In this case, the function $f()$ computes $r \cdot y_C - \lambda_i(x_Q + r \cdot x_C) - iy_Q$, where r is a random integer.

4 Proposed countermeasure

In this section, we propose a new countermeasure for securing the pairing computation on elliptic curves E defined over finite $\mathbb{K}$ with $Char(\mathbb{K}) \neq 2, 3$. The main idea of the proposed countermeasure is to compute the pairing over a random elliptic curve that belongs to the isomorphism classes of E. The output result by $f()$ will be multiplied by a random integer that belongs to $\mathbb{F}_q$. This mere change that occurs on the pairing computation process is a field multiplication that will be removed by the final exponentiation. Thus, the proposed countermeasure does not change the result of the pairing computation. In this paper, our focus will concern the Tate pairing. The extension to the Ate pairing and other pairing variants is obvious. We recall that the use of the isomorphism classes for thwarting SCAs was first proposed by Joye and Tymen (2001) for elliptic curves cryptography.

4.1 Securing the Tate pairing computation

First, we recall briefly the isomorphism classes of an elliptic curve. For more details, the reader can refer to Menezes (1993). Let E and E' be two elliptic curves defined over a finite field $\mathbb{K}$, we say that they are isomorphic if E and E' are isomorphic as algebraic varieties. The following theorem gives the isomorphism classes of an elliptic curve.

Theorem 1: Let $\mathbb{K}$ be a finite field with $Char(\mathbb{K}) \neq 2, 3$. The elliptic curves E and E' defined over $\mathbb{K}$, given respectively by $y^2 = x^3 + ax + b$ and $y^2 = x^3 + a'x + b'$ are isomorphic if and only if there exists $u \in \mathbb{K}$ such that $u^4a = a'$ and $u^6b = b'$. Furthermore, the isomorphism is defined by

$$\varphi : E(\mathbb{K}) \longrightarrow E'(\mathbb{K})$$

$$(x, y) \longrightarrow (u^2x, u^3y)$$

Let $E(\mathbb{F}_q)$ be an elliptic curve, P a point in $E(\mathbb{F}_q)[l]$ with a prime order and Q a point on the twisted curve $E'(\mathbb{F}_q)$. Instead of computing $e_l(P, Q)$ we propose to compute $\overline{e_l}(\varphi(P), \varphi(Q))$, where φ is an elliptic curve isomorphism. The new pairing $\overline{e_l}$ is defined as follows:

$$\overline{e_l} : \varphi(E(\mathbb{F}_q)[l]) \times \varphi(E'(\mathbb{F}_q)[l]) \longrightarrow \mu_l$$

$$(P', Q') \longrightarrow f_{P'}(D_{Q'})^{\frac{q^k-1}{l}}$$

To run the computation of the modified Tate pairing we perform the adding and doubling point operations on an elliptic curve of the isomorphism classes of $E(\mathbb{F}_q)$. Since the only parameter of the elliptic curve equation involved in the adding and doubling formulaes is a the computation of the secure Tate pairing is done as follows:

Algorithm 3 Secure computation of the Tate pairing

Input: an elliptic curve points $E(\mathbb{F}_q) : y^2 = x^3 + ax + b$, a point P of prime order l and a point Q on the twisted curve $E'(\mathbb{F}_q)$
Output: $\overline{e_l}(\varphi(P), \varphi(Q))$
1. Generate a random element $r \in \mathbb{F}_q$
2. $a \leftarrow r^4a$
3. $P \leftarrow \varphi(P) = (r^2x_P, r^3y_P)$
4. $Q \leftarrow \varphi(Q) = (r^2x_Q, r^3y_Q)$
5. $A \leftarrow P$
6. $m = 1$
7. $n = l - 1$
8. for $i \leftarrow \lfloor log(l) \rfloor - 1$ down to 0 do
9. $m \leftarrow m^2.f(A, A, Q)$
10. if $n_i = 1$ then $m \leftarrow m \cdot f(A, P, Q)$
11. end for
12. $m \leftarrow \frac{\overline{m}}{m}$
13. return $m^{\frac{p+1}{l}}$

Correctness of Algorithm 3: In the following, we prove that Algorithm 3 outputs the correct value of the Tate pairing.

Theorem 2: We have

$$\overline{e_l}(\varphi(P), \varphi(Q)) = e_l(P, Q)$$

To prove this theorem we will first prove the following lemma.

Lemma 1: If we set $\varphi : (x, y) \to (r^2x, r^3y)$ for some $r \in \mathbb{F}^*{}_q$, then we have

$$f(\varphi(A), \varphi(B), \varphi(Q)) = r^3 f(A, B, Q)$$

Proof: We denote $\varphi(x_R, y_R) = (x_{\varphi(R)}, y_{\varphi(R)})$ for every point R, which means that $x_{\varphi(R)} = r^2 x_R$ and $y_{\varphi(R)} = r^3 y_R$.

Since we have $C = A + B$ and φ is an isomorphism then we get $\varphi(C) = \varphi(A + B) = \varphi(A) + \varphi(B)$ which implies:

$$f(\varphi(A), \varphi(B), \varphi(Q)) = y_{\varphi(C)} - \lambda_{\varphi(C)}(x_{\varphi(Q)} + x_{\varphi(C)}) - iy_{\varphi(Q)} \quad (*)$$

where $\lambda_{\varphi(C)}$ is the slop of the line joining $\varphi(A)$ and $\varphi(B)$. From adding and doubling formulas, we have:

$$\lambda_{\varphi(C)} = \begin{cases} \frac{y_{\varphi(B)} - y_{\varphi(A)}}{x_{\varphi(B)} - x_{\varphi(A)}} \text{ if } A \neq B \\ \\ \frac{3x^2_{\varphi(A)} + a'}{2y_{\varphi(A)}} \text{ if } A = B \end{cases}$$

Since $a' = r^4 a$, $x_{\varphi(R)} = r^2 x_R$ and $y_{\varphi(R)} = r^3 y_R$ for every point R we have

$$\lambda_{\varphi(C)} = \begin{cases} r\frac{y_B - y_A}{x_B - x_A} \text{ if } A \neq B \\ \\ r\frac{3x_A^2 + a}{2y_A} \text{ if } A = B \end{cases}$$

Thus $\lambda_{\varphi(C)} = r\lambda_C$. From the equation $(*)$ we have

$$f(\varphi(A), \varphi(B), \varphi(Q)) = r^3 y_C - r\lambda_C(r^2 x_Q + r^2 x_C) - ir^3 y_Q = r^3 f(A, B, Q)$$

Which completes the proof of the lemma.

Proof of the theorem: Since φ is an isomorphism it is obvious to see that $\varphi(P) \in \varphi(E(\mathbb{F}_q))[l]$ and $\varphi(Q) \in \varphi(E'(\mathbb{F}_q))[l]$.

To prove the theorem, we will show that at the end of each loop we get $m' = \beta m$ where m is the output result by Algorithm 2 at the end of each loop, and $\beta \in \mathbb{F}^*{}_q$. This will be done by induction.

At the beginning, Algorithm 3 computes $2\varphi(P) = \varphi(2P)$ and $f(\varphi(2P), \varphi(2P), \varphi(Q))$. By the lemma 1, $f(\varphi(2P), \varphi(2P), \varphi(Q))$ is equal to $r^3 f(P, P, Q)$, for some $r \in \mathbb{F}^*{}_q$. If the corresponding bit is zero then $m' = r^3 \cdot f(P, P, Q) = r^3 m$ and thus we are done. If it is different from zero, the proposed algorithm computes $2\varphi(P) + \varphi(P) = \varphi(3P)$ and thus $m' = m' \cdot$

$f(\varphi(3P), \varphi(P), \varphi(Q))$ which is equal to $m'r^3 \cdot f(3P, P, Q)$ tanks to Lemma 1. Consequently, $m' = r^6 m$ which proves the induction hypothesis for the first loop.

Now let suppose that the induction hypothesis holds for each $i < n$. At the n^{th} loop, Algorithm 2 computes $2kP$ and $f(2kP, 2kP, Q)$ if the corresponding bit is zero, for some integer k. Thus $m \leftarrow m \cdot f(2kP, 2kP, Q)$. The proposed algorithm performs $2\varphi(kp) = \varphi(2kp)$ and $f(\varphi(2kP), \varphi(2kP), \varphi(Q)) = r^3 \cdot f(2kP, 2kP, Q)$ which implies that $m' \leftarrow m' \cdot f(\varphi(2kP), \varphi(2kP), \varphi(Q)) = r^3 m' \cdot f(2kP, 2kP, Q)$. By the induction hypothesis we have $m' = \beta m$ for some integer β. Thus $m' = \beta' m$ for some $\beta' = r^3\beta$ that belong to $\mathbb{F}_q$, which complete the proof.

If the corresponding bit is one, Algorithms 2 and 3 perform respectively the additional computation $m \cdot f((2k+1)P, P, Q)$ and $m' \cdot f(\varphi((2k+1)P), \varphi(P), \varphi(Q))$. Since $m' = \beta m$ and $f(\varphi((2k+1)P), \varphi(P), \varphi(Q)) = r^3 \cdot f((2k+1)P, P, Q)$ for some β and r in $\mathbb{F^*}_q$, then the induction hypothesis holds which completes the proof of the theorem.

4.2 Security and efficiency

This section discusses the security and efficiency of the proposed countermeasure. The weakness of Algorithm 2 lies in the computation of $f() : y_C - \lambda_i(x_Q + x_C) - iy_Q$, which

involves the coordinates of the secret point. With the proposed countermeasure, all used variables at each loop are randomised. Indeed, the function $f()$ is computed as follows:

$$\begin{aligned} &y_{\varphi(C)} - \lambda_{\varphi(C)}(x_{\varphi(Q)} + x_{\varphi(C)}) - iy_{\varphi(Q)} \\ &= r^3 \cdot y_C - r \cdot \lambda_C(r^2 \cdot x_Q + r^2 \cdot x_C) - ir^3 \cdot y_Q \end{aligned}$$

Furthermore, the integers used to randomise all intermediate values are changed at each loop. Hence, the proposed countermeasure randomises the behaviour of the Tate pairing computation and thus makes it secure against SCAs. In fact, the proposed countermeasure acts like the Scott's one but the overhead computation is done only once.

To estimate the cost of the proposed countermeasure we need to evaluate the field operations required to make Algorithm 2 secure against SCAs. The only more field operations required by the proposed countermeasure are the computations of $\varphi(P) = (r^2 x_P, r^3 y_P), \varphi(Q) = (r^2 x_Q, r^3 y_Q)$ and $a' = r^4 a$ which cost exactly six field multiplications (M) and two field squaring (S). To our knowledge, in field $\mathbb{K}$ with $Char(\mathbb{K}) \neq 2, 3$ the cost of a squaring is not less expensive to that of a multiplication. Thus, we can conclude that the cost of our countermeasure is eight field multiplications.

Now we estimate the computational cost of the techniques proposed by Scott (2005) and Kim et al. (2006). Theses techniques are the most efficient known countermeasures for securing the pairing-based computation against side channel attack. The countermeasure proposed by Scott randomises all the intermediate variables in Algorithm 2 by an integer r. With this technique, Algorithm 1 computes $ry_C - \lambda_i(rx_Q + rx_C) - iry_Q$ instead of $y_C - \lambda_i(x_Q + x_C) - iy_Q$. This costs four field multiplications. At each loop the function f is used once by Algorithm 2 if the corresponding bit is zero and twice otherwise. Thus, the cost of the countermeasure proposed by Scott is approximately $4log(l)M + 4l'M = 6log(l)M$,

where l' is the number of non-zero bits in the binary representation of l. On average, the number of non-zero bits in the binary representation of l is approximately $log(l)/2$, then the cost of the countermeasure proposed by Scott is approximately $4log(l)M + 4(\frac{log(l)}{2})M = 6log(l)M$.

When the mixed coordinates are used, as suggested in Kim et al. (2006), the function $f()$ computes $ry_C - \lambda_i(x_Q + rx_C) - iy_Q$ instead of $y_C - \lambda_i(x_Q + x_C) - iy_Q$. In the same way, it is easy to see that the cost of the countermeasure proposed by Kim et al. is approximately $log(l)M + \frac{log(l)}{2}M = \frac{3log(l)}{2}M$.

Table 1 gives a comparison of overheads for countermeasures against SCAs.

Table 1 Efficiency study

Countermeasure	*Additional cost*
Scott countermeasure	$6log(l)M$
Kim et al. countermeasure	$\frac{3log(l)}{2}M$
Proposed countermeasure	$8M$

5 Conclusions

In this paper, we have proposed a new countermeasure for securing pairing computation on elliptic curves defined over field $\mathbb{F}_q$ ($Char(\mathbb{F}_q) \neq 2, 3$). The proposed countermeasure is based on elliptic curve isomorphism classes. The cost of the proposed countermeasure is only eight field multiplications. Thus, The only field operations required by the proposed countermeasure are the computations of $\varphi(P) = (r^2x_P, r^3y_P)$, $\varphi(Q) = (r^2x_Q, r^3y_Q)$ and $a' = r^4a$ which cost exactly eight field multiplications.

References

Barreto, P.S.L.M., Kim, H.Y., Lynn, B. and Scott, M. (2002) 'Efficient algorithms for pairing-based cryptosystems', in *Advances in Cryptology*, LNCS 2442, pp.354–368, Springer-Verlag.

Barreto, P.S.L.M., Galbraith, S.D., O'Eigeartaigh, C. and Scott, M. (2007) 'Efficient pairing computation on supersingular abelian varieties', in *Designs, Codes and Cryptography*, Vol. 42, No. 3, pp.239–271.

Boneh, D. and Franklin, M. (2003) 'Identity based encryption from the Weil pairing', in *Journal of Computing*, Vol. 32, No. 3, pp.586–615.

Duursma, I.M. and Lee, H-S. (2003) 'Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$', in *ASIACRYPT*, LNCS 2894, pp.111–123, Springer-Verlag.

Granger, R., Hess, F., Oyono, R., Theriault, N. and Vercauteren, F. (2007) 'Ate pairing on hyperelliptic curves', in *EUROCRYPT*, LNCS 4515, pp.430–447, Springer-Verlag.

Joye, M. and Tymen, C. (2001) 'Protections against differential analysis for elliptic curve cryptography: an algebraic approach', in *CHES 2000*, Vol. 2162, LNCS, pp.377–390, Springer-Verlag.

Kim, T-H., Takagi, T., Han, D-G., Kim, H.W. and Lim, J. (2006) 'Side channel attacks and countermeasures on pairing based cryptosystems over binary fields', in *Cryptology and Network Security (CANS 2006)*, LNCS 4301, pp.168–181, Springer-Verlag.

Kocher, P., Jaffe, J. and Jun, B. (1999) 'Differential power analysis', in M. Wiener (Ed.): *Advances in Cryptology-CRYPTO '99*, Vol. 1666, pp.388–397, LNCS.

Menezes, A.J. (1993) *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, Massachusetts. USA.

Miller, V.S. (2004) 'The Weil pairing, and its efficient calculation', in *Journal of Cryptology*, Vol. 17, No. 4, pp.235–261.

Page, D. and Vercauteren, F. (2004) 'Fault and side-channel attacks on pairing based cryptography', in Cryptology ePrint Archive, Report 2004/283.

Scott, M. (2005) 'Computing the Tate pairing', in *CT-RSA 2005*, LNCS 3376, pp.293–304, Springer-Verlag.

Scott, M., Costigan, N. and Abdulwahab, W. (2006) 'Implementing cryptographic pairings on smartcards', in *CHES 2006*, LNCS 4249, pp.134–147, Springer-Verlag.

Shu, C., Kwon, S. and Gaj, K. (2006) 'FPGA accelerated Tate pairing based cryptosystems over binary fields', in Cryptology ePrint Archive, Report 2006/179.

Whelan, C., Page, D., Vercautern, F., Scott, M. and Marnane, W. (2008) 'Implementation attacks and countermeasures', in M. Joye and G. Neven (Eds.): *Identity-Based Cryptography*, Vol. 1, Cryptology and Information Security Series.