Speeding up the Bilinear Pairings Computation on Curves with Automorphisms

Chang-An Zhao, Fangguo Zhang, and Jiwu Huang

[1] School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, P.R.China
[2] Guangdong Key Laboratory of Information Security Technology, Guangzhou 510275, P.R.China
zhcha@mail2.sysu.edu.cn
isszhfg@mail.sysu.edu.cn
isshjw@mail.sysu.edu.cn

Abstract. In this paper we present a new algorithm for computing the bilinear pairings on a family of non-supersingular elliptic curves with non-trivial automorphisms. We obtain a short iteration loop in Miller's algorithm using non-trivial efficient automorphisms. The proposed algorithm is as efficient as the algorithm in [12].

Keywords: Tate pairing, non-supersingular curves, pairing-based cryptosystems, automorphisms.

A Note

This work is same as [12] from algorithmic angle. We only give a new derivation of Scott's algorithm [12]. Therefore this paper will not be submitted anywhere, it is just another explanation of Scott's algorithm [12].

1 Introduction

The bilinear pairing is an explicit mapping from a set of certain points on an elliptic curve to a multiplicative subgroup of a finite field. It has found many interesting applications in elliptic curve cryptography [11].

Many efficient algorithms for implementing the pairings have been proposed [6]. In particular, the Eta pairing [1] and the Ate pairing [8] are introduced

for their efficient computations recently. Their main ideas are to shorten the main iteration loop in Miller's algorithm [10]. The Eta pairing optimizes Miller's algorithm using some special automorphisms on supersingular curves, and the Ate pairing speeds up the bilinear pairings computation mainly using Frobenius endomorphisms on non-supersingular elliptic curves.

In this paper, we utilize some non-trivial automorphisms on a family of non-supersingular elliptic curves for accelerating the bilinear pairings computation. We obtain a short iteration loop using some efficient automorphisms. The length of the main iteration loop in our algorithm is half the length of the main loop in traditional Miller's algorithm.

The rest of this paper is organized as follows. Section 2 explains the Tate pairing and a family of non-supersingular elliptic curves with non-trivial automorphisms. Section 3 gives the main results and proposes a new algorithm. Section 4 analyzes the efficiency of the proposed algorithm. Section 5 gives the conclusions.

2 Mathematical Preliminaries

2.1 The Tate Pairing

Let $\mathbb{F}_q$ be a finite field with $q = p^m$ elements, where p is a prime. Let E be an elliptic curve defined over $\mathbb{F}_q$, and let $\mathcal{O}$ be the point at infinity. Let r be a prime such that $r|\#E(\mathbb{F}_q)$, and let k be the embedding degree, i.e., the minimal positive integer such that $r|q^k-1$. We also assume that r^2 does not divide q^k-1 and k is greater than 1.

Let $P \in E[r]$ and $Q \in E(\mathbb{F}_{q^k})$, and let D be the divisor which is equivalent to $(Q)-(\mathcal{O})$. For every integer i and point P, let $f_{i,P}$ be a function such that

$$(f_{i,P}) = i(P) - (iP) - (i-1)(\mathcal{O}).$$

Then the Tate pairing is a map

$$e : E[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r,$$

$$e(P,Q) = f_{r,P}(D).$$

By Theorem 1 in [2], one can define the reduced Tate pairing as

$$e(P,Q) = f_{r,P}(Q)^{\frac{q^k-1}{r}}.$$

The above definition is convenient since we often require a unique element of $\mathbb{F}_{q^k}$ in cryptographic applications. Note that $f_{r,P}(Q)^{a(q^k-1)/r} = f_{ar,P}(Q)^{(q^k-1)/r}$ for any integer a .

2.2 Miller's Algorithm

In this subsection, we briefly recall how the Tate pairing can be computed in polynomial time using Miller's algorithm [10].

Let $P \in E[r]$ and $Q \in E(\mathbb{F}_{q^k})$. Let $l_{R,T}$ be the equation of the line through points R and T, and let v_S be the equation of the vertical line through point S. Then for $i, j \in \mathbb{Z}$, we have

$$f_{i+j,P}(Q) = f_{i,P}(Q) f_{j,P}(Q) \frac{l_{iP,jP}(Q)}{v_{(i+j)P}(Q)}.$$

Miller's algorithm is described in Algorithm 1.

Algorithm 1 Miller's algorithm

Input: $r = \sum_{i=0}^{n} l_i 2^i$, where $l_i \in \{0, 1\}$. $P \in E[r]$ and $Q \in E(F_{q^k})$.
Output: $e(P, Q)$
1. $T \leftarrow P$, $f_1 \leftarrow 1$
2. for $i = n-1, n-2, ..., 1, 0$ do
2.1 $f_1 \leftarrow f_1^2 \cdot \frac{l_{T,T}(Q)}{v_{2T}(Q)}$, $T \leftarrow 2T$
2.2 if $l_i = 1$ then
2.3 $f_1 \leftarrow f_1 \cdot \frac{l_{T,P}(Q)}{v_{T+P}(Q)}$, $T \leftarrow T + P$
3. return $f_1^{(q^k-1)/r}$

2.3 A Family of Elliptic Curves with Non-trivial Automorphisms

Let p be a large prime, consider the underlying non-supersingular elliptic curves over $\mathbb{F}_p$

$$E_1 : y^2 = x^3 + B, \text{where } p \equiv 1 \bmod 3$$
$$E_2 : y^2 = x^3 + Ax, \text{where } p \equiv 1 \bmod 4$$

Both them have efficiently-computable endomorphisms which have been applied in fast point multiplication [7] and computations of the Tate pairing [12]. In fact, these endomorphisms are also non-trivial automorphisms which have been applied in speeding up the discrete log computation [5].

We will mainly consider the first curve E_1 for accelerating the computation of the bilinear pairings in this paper. However, the generalization to the second curve E_2 is direct. Note that some suitable curves like E_1 have low embedding degrees and hence can be applied in pairing-based cryptosystems [12, 16].

Suppose that β is an element of order three in $\mathbb{F}_p$. A non-trivial automorphism of the above curve E_1 is defined as

$$\phi : E_1 \to E_1$$

$$(x, y) \to (\beta x, y).$$

Since this automorphism ϕ is also an isogeny, its dual isogeny is

$$\hat{\phi} : E_1 \to E_1$$

$$(x, y) \to (\beta^2 x, y).$$

It is not difficult to show that $\hat{\phi} \circ \phi = [1]$, $\hat{\phi} = \phi^2$ and $\#ker\phi = 1$ (see Silverman [15] pages 84-86). Note that $\hat{\phi}$ is also a non-trivial automorphism on the first curve E_1.

We cite some useful facts from [7]. Let $P \in E_1(\mathbb{F}_p)$ be a point of prime order r, where r^2 does not divide the order of $E_1(\mathbb{F}_p)$. Then ϕ and $\hat{\phi}$ act restrictively on the subgroup $<P>$ as multiplication maps $[\lambda]$ and $[\hat{\lambda}]$ respectively, where λ and $\hat{\lambda}$ are two roots of the equation: $x^2 + x + 1 = 0 \pmod{r}$. Note that $\lambda P = \phi(P)$ can be computed using one multiplication in $\mathbb{F}_p$.

Similarly, let $\alpha \in \mathbb{F}_p$ be an element of order 4. Then the map $\phi_2 : E_2 \to E_2$ defined by $(x, y) \mapsto (-x, \alpha y)$ is an automorphism of E_2. Its dual isogeny $\hat{\phi_2} : E_2 \to E_2$ is defined by $(x, y) \mapsto (-x, -\alpha y)$. Let $P \in E_2(\mathbb{F}_p)$ is a point of prime order r, then ϕ_2 acts restrictively on $<P>$ as a multiplication map $[\lambda]$, where λ is an integer satisfying $\lambda^2 + 1 = 0 \bmod r$. Note that $\lambda P = \phi_2(P)$ can be also computed using one multiplication in $\mathbb{F}_p$.

3 Speeding up the Bilinear Pairings Computation

In this section, we obtain a new bilinear pairing on some non-supersingular curves using non-trivial automorphisms. As a consequence, a novel algorithm for computing the bilinear pairings is proposed.

3.1 Main Result

The main result of this paper is summarized in the following theorem.

Theorem 1. *Let E_1 be a non-supersingular curve over $\mathbb{F}_p$ with automorphisms ϕ and $\hat{\phi}$ defined as above. Let $k > 1$ be its embedding degree. Let $P \in E_1(\mathbb{F}_p)$ be a point of prime order r, where $r^2 \nmid \#E(\mathbb{F}_p)$. Let $[\lambda]$ be the multiplication map of the subgroup $<P>$ defined as above such that $\lambda P = \phi(P)$. Let a be an integer such that $ar = \lambda^2+\lambda+1$. Let $l_{\phi(P),\hat{\phi}(P)}$ be the equation of the line through points $\phi(P)$ and $\hat{\phi}(P)$. Then for $Q \in E_1(\mathbb{F}_{p^k})$, we have*

$$e(P,Q)^a = (f_{\lambda,P}(Q)^{\lambda+1} \cdot f_{\lambda,P}(\hat{\phi}(Q)) \cdot l_{\phi(P),\hat{\phi}(P)}(Q))^{\frac{p^k-1}{r}}.$$

Note that such an integer a must exist since $\lambda^2+\lambda+1 = 0 \pmod{r}$. A second useful remark is that $e(P,Q)^a$ is non-degenerate provided that r does not divide a. The proof of Theorem 1 is split into three short lemmas.

Lemma 1. *Using the notation of Theorem 1, we have*

$$e(P,Q)^a = (f_{\lambda^2+\lambda,P}(Q) \cdot l_{-P,P}(Q))^{\frac{p^k-1}{r}}.$$

Proof. By definition of the reduced Tate pairing, we have

$$e(P,Q)^a = f_{r,P}(Q)^{\frac{a(p^k-1)}{r}} = f_{ar,P}(Q)^{\frac{p^k-1}{r}}.$$

Since $ar = \lambda^2 + \lambda + 1$, we obtain

$$e(P,Q)^a = f_{ar,P}(Q)^{\frac{p^k-1}{r}} = f_{\lambda^2+\lambda+1,P}(Q)^{\frac{p^k-1}{r}}.$$

since $(\lambda^2 + \lambda)P = -P$, we have

$$(f_{\lambda^2+\lambda+1,P}) = (f_{\lambda^2+\lambda,P} \cdot f_{1,P} \cdot l_{-P,P}).$$

Furthermore, $f_{1,P} = 1$ up to a scalar multiple in $\mathbb{F}_p^*$, so we obtain

$$e(P,Q)^a = f_{\lambda^2+\lambda+1,P}(Q)^{\frac{p^k-1}{r}} = (f_{\lambda^2+\lambda,P}(Q) \cdot l_{-P,P}(Q))^{\frac{p^k-1}{r}}$$

which completes the proof. □

Lemma 2. *Using the notation of Theorem 1, we can choose $f_{\lambda^2+\lambda,P} l_{-P,P}$ such that*

$$(f_{\lambda^2+\lambda,P} \cdot l_{-P,P}) = (f_{\lambda,P}^{\lambda+1} \cdot f_{\lambda,\lambda P} \cdot l_{\phi(P),\hat{\phi}(P)}).$$

Proof. We have $(f_{i,P}) = i(P) - (iP) - (i-1)(\mathcal{O})$ and $(\lambda^2+\lambda)P = -P$. Therefore

$$\begin{aligned}(f_{\lambda^2+\lambda,P} \cdot l_{-P,P}) =& (f_{\lambda^2,P} \cdot f_{\lambda,P} \cdot \frac{l_{\lambda^2 P,\lambda P}}{l_{(\lambda^2+\lambda)P,-(\lambda^2+\lambda)P}} \cdot l_{-P,P}) \\ =& (f_{\lambda^2,P} \cdot f_{\lambda,P} \cdot l_{\lambda^2 P,\lambda P}).\end{aligned}$$

Furthermore, since $\lambda P = \phi(P)$ and $\lambda^2 P = \phi^2(P) = \hat{\phi}(P)$, we have

$$l_{\lambda^2 P,\lambda P} = l_{\lambda P,\lambda^2 P} = l_{\phi(P),\hat{\phi}(P)}.$$

Also, (see Lemma 2 in [1])

$$(f_{\lambda^2,P}) = (f^{\lambda}_{\lambda,P} \cdot f_{\lambda,\lambda P}).$$

Hence we have

$$(f_{\lambda^2+\lambda,P} \cdot l_{-P,P}) = (f_{\lambda^2,P} \cdot f_{\lambda,P} \cdot l_{\lambda^2 P,\lambda P}) = (f^{\lambda+1}_{\lambda,P} \cdot f_{\lambda,\lambda P} \cdot l_{\phi(P),\hat{\phi}(P)})$$

which completes the proof . □

Lemma 3. *For $P \in E(\mathbb{F}_p)[r]$ and $Q \in E(\mathbb{F}_{p^k})$, we have $f_{\lambda,\lambda P}(Q) = f_{\lambda,P}(\hat{\phi}(Q))$, with ϕ and $\hat{\phi}$ defined as above.*

Proof. By definition we have $(f_{\lambda,\lambda P}) = \lambda(\lambda P) - (\lambda^2 P) - (\lambda - 1)(\mathcal{O})$. We also have $\phi(P) = \lambda P$ and $\#ker\phi = deg[1] = 1$ (see [15] Chapter III pages 85-86). By properties of the pullback we obtain

$$\begin{aligned}\phi^*(f_{\lambda,\lambda P}) =& \phi^*(\lambda(\lambda P) - (\lambda^2 P) - (\lambda - 1)(\mathcal{O})) \\ =& \lambda(P) - (\lambda P) - (\lambda - 1)(\mathcal{O}) \\ =& (f_{\lambda,P}).\end{aligned}$$

Furthermore, $\phi^*(f_{\lambda,\lambda P}) = (f_{\lambda,\lambda P} \circ \phi)$, hence we can take (up to a scalar multiple in $\mathbb{F}_p^*$)

$$f_{\lambda,\lambda P} \circ \phi = f_{\lambda,P}.$$

Applying $\hat{\phi}$ to the above equality yields

$$f_{\lambda,\lambda P} \circ \phi \circ \hat{\phi} = f_{\lambda,P} \circ \hat{\phi}.$$

Since $\phi \circ \hat{\phi} = [1]$, we have

$$f_{\lambda,\lambda P} = f_{\lambda,P} \circ \hat{\phi}.$$

This completes the proof. □

Proof of Theorem 1: Since $P \in E(\mathbb{F}_p)[r]$, Lemma 3 gives

$$f_{\lambda,\lambda P}(Q) = f_{\lambda,P}(\hat{\phi}(Q)),$$

and substituting the above equality into Lemma 2, we can easily obtain

$$f_{\lambda^2+\lambda,P}(Q) \cdot l_{-P,P}(Q) = f_{\lambda,P}^{\lambda+1}(Q) \cdot f_{\lambda,P}(\hat{\phi}(Q)) \cdot l_{\phi(P),\hat{\phi}(P)}(Q).$$

Substituting the above equation into Lemma 1, we have

$$\begin{aligned} e(P,Q)^a =& (f_{\lambda^2+\lambda,P}(Q) \cdot l_{-P,P}(Q))^{\frac{p^k-1}{r}} \\ =& (f_{\lambda,P}(Q)^{\lambda+1} \cdot f_{\lambda,P}(\hat{\phi}(Q)) \cdot l_{\phi(P),\hat{\phi}(P)}(Q))^{\frac{p^k-1}{r}}. \end{aligned}$$

This completes the whole proof of Theorem 1. □

Similarly, such a bilinear pairing also exists on the second curve E_2 with non-trivial automorphisms ϕ_2 and $\hat{\phi}_2$. Let $P \in E_2(\mathbb{F}_p)$ be a point of large prime order r such that $\lambda^2 + 1 = ar$, where a is an integer. Let k be the embedding degree of E_2. Then for $Q \in E_2(\mathbb{F}_{p^k})$, we also gives the following formula

$$e(P,Q)^a = (f_{\lambda,P}(Q)^{\lambda} \cdot f_{\lambda,P}(\hat{\phi}_2(Q)))^{\frac{p^k-1}{r}}$$

for computing the new bilinear pairing on E_2.

Note that $e(P,Q)^a$ gives a bilinear pairing since $e(P,Q)$ is bilinear. Furthermore, it is non-degenerate provided that $r \nmid a$, which is the practice case since a is much smaller than r. Therefore, we obtain a new non-degenerate, bilinear pairing which is equal to a fixed power of the traditional reduced Tate pairing. In practice, we can make that a is equal to 1 for some elliptic curves with non-trivial automorphisms [12, 16]. In this case, we can keep that the value of the new pairing is equal to a correct value of the Tate pairing.

3.2 A New Algorithm for Computing the Bilinear Pairings

In this section, we will give a new algorithm for computing the pairing $e(P,Q)^a$ by Theorem 1. For simplicity, we only consider non-supersingular elliptic curves with non-trivial automorphisms having embedding degree $k = 2$. However, the new method can also apply to non-supersingular curves with large embedding degrees. It should be pointed out that some suitable curves with large embedding degrees are given in the work of Takashima [16].

Let Q be in the trace-zero subgroup [3, 13] for good efficiency. Note that the denominator can be omitted in Miller's algorithm since the x-coordinates of P, Q and $\hat{\phi}(Q)$ are contained in $\mathbb{F}_p^*$ now. Let $l_{R,T}$ be the equation of the line through points R and T. The proposed algorithm is given in Algorithm 2.

Algorithm 2 Computations of $e(P,Q)^a$ using automorphisms

Input: $\lambda = \sum_{i=0}^{n} l_i 2^i$, where $l_i \in \{0,1\}$. $P \in E(\mathbb{F}_p)[r]$ and $Q \in E(F_{p^k})$.
Output: $e(P,Q)^a$
1. $T \leftarrow P$, $f_1 \leftarrow 1$, $f_2 \leftarrow 1$, $f_3 \leftarrow l_{\phi(P),\hat{\phi}(P)}(Q)$
2. for $i = n-1, n-2, ..., 1, 0$ do
2.1 $f_1 \leftarrow f_1^2 \cdot l_{T,T}(Q)$, $f_2 \leftarrow f_2^2 \cdot l_{T,T}(\hat{\phi}(Q))$, $T \leftarrow 2T$
2.2 if $l_i = 1$ then
2.3 $f_1 \leftarrow f_1 \cdot l_{T,P}(Q)$, $f_2 \leftarrow f_2 \cdot l_{T,P}(\hat{\phi}(Q))$, $T \leftarrow T + P$
3. $f_1 \leftarrow f_1^{\lambda+1}$,
4. return $(f_1 \cdot f_2 \cdot f_3)^{(p-1)(p+1)/r}$

We give some useful remarks on Algorithm 2. The equation of the line $l_{\phi(P),\hat{\phi}(P)}$ is easily obtained since $\phi(P)$ and $\hat{\phi}(P)$ have the same y-coordinate. Computing $\hat{\phi}(Q)$ only requires one multiplication since β^2 and x-coordinate of Q are in $\mathbb{F}_p^*$. Let T be in $E(\mathbb{F}_p)$ with coordinates (x_T, y_T), and let m be the slope of the line $l_{T,T}$. Then the equation of the line $l_{T,T}$ is $(y - y_T) - m(x - x_T) = 0$. Therefore the evaluation of $l_{T,T}(Q)$ and $l_{T,T}(\hat{\phi}(Q))$ only requires two multiplications in $\mathbb{F}_p$. Similarly, computing $l_{T,P}(Q)$ and $l_{T,P}(\hat{\phi}(Q))$ also requires two multiplications in $\mathbb{F}_p$.

4 Efficiency Consideration

Now the performance of the proposed algorithm is considered in this section. We neglect the cost of field additions and subtractions, as well as the cost of multiplication by small constants. The computational cost of one multiplication in $\mathbb{F}_p^*$ is denoted as M.

We first point out that $\hat{\phi}(Q)$ can be precalculated using only $1M$ because β^2 and x-coordinate of Q are in $\mathbb{F}_p^*$. The evaluations of $l(Q)$ and $l(\hat{\phi}(Q))$ cost $2M$ since the equation of the line can be reused. One multiplication and one square in $\mathbb{F}_{p^2}^*$ require $3M$ and $2M$, respectively [13]. We assume that the computational cost of an inverse in $\mathbb{F}_p^*$ is $10M$. We also count one square as one multiplication

in $\mathbb{F}_p^*$. Hence, one point doubling requires $14M$ and one point addition requires $13M$ in $E(\mathbb{F}_p)$ [9].

For good efficiency, we can make that λ has a low Hamming weight. Scott has found such a suitable elliptic curve with $\lambda = 2^{80} + 2^{16}$ using Cocks-Pinch algorithm [12]. We compute the Tate pairing on this curve using the proposed algorithm in the following. Note that $r = \lambda^2 + \lambda + 1$ has 161 bits. Hence a is equal to 1 in this case. Therefore the value of $e(P,Q)^a$ is the same as the value of the traditional Tate pairing. Note that the length of the iteration loop in the new algorithm is half the length of the iteration loop in the previous traditional algorithm. However, the new algorithm requires more multiplications than the traditional algorithm in one iteration loop.

Now we give a detailed efficiency consideration on the proposed algorithm. Let $P = (x_P, y_P) \in E(\mathbb{F}_p)[r]$ and $Q = (x_Q, y_Q) \in E(\mathbb{F}_{p^2})$, where $x_Q \in \mathbb{F}_p$ and $y_Q \in \mathbb{F}_{p^2}$. We first consider the cost of line 1 in Algorithm 2. It is easily checked that $l_{\phi(P),\hat{\phi}(P)}(Q)$ is equal to $y_Q - y_P$ since P, $\phi(P)$ and $\hat{\phi}(P)$ have the same y-coordinate. So line 1 requires no multiplications in Algorithm 2. Now we consider the computational cost of line 2.1 in Algorithm 2. The cost of one point doubling is $14M$. The two line equation evaluations require $2M$ since $\hat{\phi}(Q)$ can be precalculated easily. The remainder in 2.1 requires two squares and two multiplications in $F_{p^2}^*$, which cost $10M$. Therefore line 2.1 requires $26M$. The number of the main loop is 79, so the total cost of line 2.1 is $26 \cdot 79 = 2054M$. It is not difficult to show that the total cost of line 2.3 requires $21M$. Line 3 requires $80 \cdot 2 = 160M$ for this exponentiation. By now we cost $2054 + 21 + 160 = 2235M$. There are two multiplications in $\mathbb{F}_{p^2}$ in line 4 of Algorithm 2, which require $6M$. The exponentiation $(p-1)$ requires five multiplications and one inverse in $\mathbb{F}_p^*$ since the Frobenius map can be used here. Using the Lucas laddering algorithm [14], we can compute the exponentiation $(p+1)/r$ in $(512 - 161) \cdot 2 = 702M$. So the total contribution of line 4 is $6 + 15 + 702 = 723M$. Therefore the total computational cost of the new algorithm is $2235 + 723 = 2958M$.

5 Conclusion

A efficient algorithm has been proposed for computing the bilinear pairing on a family of non-supersingular curves with non-trivial automorphisms. Similar to the Eta pairing and the Ate pairing, the main technique in this paper is to shorten

the main iteration loop. The proposed method is as efficient as the previous methods in [12] on these elliptic curves. It should be pointed out that the new method can also be used in non-supersingular curves having large embedding degrees. It is possible to further optimize the new algorithm and extend it in hyperelliptic curves.

Acknowledgements

We would like to thank Xiao Ma for his helpful comments which improved this paper, and Katsuyuki Takashima for providing us a copy of [16].

References

1. P.S.L.M. Barreto, S. Galbraith, C. ÓhÉigeartaigh, and M. Scott. Efficient Pairing Computation on Supersingular Abelian Varieties. *In Designs, Codes and Cryptography*. Springer-Verlag, 2005. Also availabe from http://eprint.iacr.org/2004/375.
2. P.S.L.M. Barreto, H.Y. Kim, B. Lynn, and M. Scott. Efficient Algorithms for Pairing-based Cryptosystems. *In Advances in Cryptology-Crypto'2002*, volume 2442 of *Lecture Notes in Computer Science*, pages. 354-368. Springer-Verlag, 2002.
3. P.S.L.M. Barreto, H.Y. Lynn, M. Scott. On the Selection of Pairing-Friendly Groups. *10th Annual International Workshop, SAC 2003,*, volume 3006 of *Lecture Notes in Computer Science*, pages. 17-25. Springer-Verlag, 2004.
4. D. Boneh and M. Franklin. Identity-based Encryption from the Weil pairing. *SIAM Journal of Computing*, 32(3): 586-615, 2003.
5. I. Duursma, P. Gaudry, and F. Morain. *Speeding up the Discrete Log Computation on Curves with Automorphisms*, *AsiaCrypt'99*, volume 1716 of *Lecture Notes in Computer Science*, pages. 203-121. Springer-Verlag, 1999.
6. S. Galbraith. Pairings, *Ch. IX of I.F.Blake, G.Seroussi, and N.P.Smart, eds., Advances in Elliptic Curve Cryptography*. Cambridge University Press, 2005.
7. R.P. Gallant, R.J. Lambert and S.A. Vanstone. Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms, *In Advances in Cryptology - Crypto 2001*, volume 2139 of *Lecture Notes in Computer Science*. Springer- Verlag, 2001.
8. F. Hess, N.P. Smart and F. Vercauteren. The Eta Pairing Revisited. *IEEE Transactions on Information Theory*, vol 52, Pages. 4595-4602, Oct. 2006. Also available from http://eprint.iacr.org/2006/110.
9. IEEE Std 1363-2000. Standard Specifications for Public-key Cryptography. IEEE P1363 Working Group, 2000.

10. V.S. Miller. Short Programs for Functions on Curves. Unpublished manuscript, 1986.
11. K.G. Paterson. Cryptography from Pairing. *Ch. X of I.F.Blake, G.Seroussi, and N.P.Smart, eds., Advances in Elliptic Curve Cryptography*. Cambridge University Press, 2005.
12. M. Scott. Faster Pairings Using an Elliptic Curve with an Efficient Endomorphism. *Progress in Cryptology - INDOCRYPT 2005*, volume 3797 of *Lecture Notes in Computer Science*, pages. 258-269. Springer-Verlag, 2005.
13. M. Scott. Computing the Tate Pairing. *In CT-RSA05*, volume 3376 of *Lecture Notes in Computer Science*, pages. 293-304. Springer-Verlag, 2005.
14. M. Scott and P.S.L.M. Barreto. Compressed Pairings. *In Advances in Cryptology-Crypto'2004*, volume 3152 of *Lecture Notes in Computer Science*, pages. 140-156. Springer-Verlag, 2004.
15. J.H. Silverman. *The Arithmetic of Elliptic Curves*. Springer-Verlag, New York, 1986.
16. K. Takashima. Scaling Security of Elliptic Curves with Fast Pairing Using Efficient Endomorphisms. *IEICE Trans. Fundamentals*, vol E90-A, no.1, pages. 152-159. Jan. 2007.