Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms

R. Gallant[1], R. Lambert[1], and S. Vanstone[1,2]

[1] Certicom Research, Canada
{rgallant,rlambert,svanstone}@certicom.com
[2] University of Waterloo, Canada

Abstract. The fundamental operation in elliptic curve cryptographic schemes is that of point multiplication of an elliptic curve point by an integer. This paper describes a new method for accelerating this operation on classes of elliptic curves that have efficiently-computable endomorphisms. One advantage of the new method is that it is applicable to a larger class of curves than previous such methods.

1 Introduction

Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$. The dominant cost operation in elliptic curve cryptographic schemes is *point multiplication*, namely computing kQ where Q is an elliptic curve point and k is an integer. This operation is the additive analogue of the exponentiation operation α^k in a general (multiplicative-written) finite group. The basic technique for exponentiation is the repeated square-and-multiply algorithm. Numerous methods for speeding up exponentiation and point multiplication have been discussed in the literature; for a survey, see [9, 10, 14]. These methods can be categorized as follows:

1. Generic methods which can be applied to speed up exponentiation in any finite abelian group, including:
   (a) Comb techniques (e.g. [13]) which precompute tables which depend on Q. Such techniques are applicable when the base point Q is fixed and known a priori, for example in ECDSA signature generation.
   (b) Addition chains which are useful when k is fixed, for example in RSA decryption.
   (c) Windowing techniques which are useful when the base point Q is not known a priori, for example in Diffie-Hellman key agreement.
   (d) Simultaneous multiple exponentiation techniques for computing expressions $k_1Q_1 + k_2Q_2 + \cdots + k_tQ_t$, for example in ECDSA signature verification.

2. Exponent recoding techniques which replace the binary representation of k with a representation which has fewer non-zero terms (e.g, [8, 17]).
3. Methods which are particular to elliptic curve point multiplication such as:
   (a) Selection of an underlying finite field which enables faster field arithmetic. For example, selection of a prime field $\mathbb{F}_p$ where p is a Mersenne prime or a Mersenne-like prime [28], or an optimal field extension [1].
   (b) Selection of a representation of the underlying finite field which enables faster field arithmetic. For example, selection of an irreducible trinomial as the reduction polynomial for binary extension fields.
   (c) Selection of a point representation which enables faster elliptic curve arithmetic [4].
   (d) Selection of an elliptic curve with special properties, for example Koblitz curves [11].

Koblitz curves are elliptic curves defined over $\mathbb{F}_2$, and were first proposed for cryptographic use in [11]. The primary advantage of Koblitz curves is that the Frobenius endomorphism can be exploited to devise fast point multiplication algorithms that do not use any point doublings [27, 29]. These techniques can be generalized to use arbitrary endomorphisms but are generally not efficient.

The contribution of this paper is a new technique for speeding up point multiplication of elliptic curves having an efficiently-computable endomorphism. While the technique is not as efficient as the methods of Solinas [27, 29] for Koblitz curves, they are useful for speeding up point multiplication on a larger class of eliptic curves, for example certain curves over prime fields. Such elliptic curves over prime fields have been included in the WAP WTLS (Wireless Transport Layer Security) standard [30].

The remainder of this paper is organized as follows. §2 defines an endomorphism and reviews how the Frobenius endomorphism can be used to speed up point multiplication in Koblitz curves. Our new work for speeding up point multiplication in elliptic curves which have efficiently-computable endomorphisms is described in §3 and §4. The security of the new method is considered in §5. Finally, we draw our conclusions and discuss avenues for future work in §6.

2 Endomorphisms

Let E be an elliptic curve defined over the finite field $\mathbb{F}_q$. The point at infinity is denoted by $\mathcal{O}$. For any $n \geq 1$, the group of $\mathbb{F}_{q^n}$-rational points on E is denoted by $E(\mathbb{F}_{q^n})$.

An *endomorphism* of E is a rational map $\phi : E \to E$ satisfying $\phi(\mathcal{O}) = \mathcal{O}$ [24]. If the rational map is defined over $\mathbb{F}_q$, then the endomorphism ϕ is also said to be defined over $\mathbb{F}_q$. In this case, ϕ is a group homomorphism of $E(\mathbb{F}_q)$, and also of $E(\mathbb{F}_{q^n})$ for any $n \geq 1$.

Example 1. Let E be an elliptic curve defined over $\mathbb{F}_q$. For each $m \in \mathbb{Z}$ the *multiplication by* m map $[m] : E \to E$ defined by $P \mapsto mP$ is an endomorphism defined over $\mathbb{F}_q$. A special case is the *negation* map defined by $P \mapsto -P$.

Example 2. Let E be an elliptic curve defined over $\mathbb{F}_q$. Then the q^{th} power map $\phi : E \to E$ defined by $(x, y) \mapsto (x^q, y^q)$ and $\mathcal{O} \mapsto \mathcal{O}$ is an endomorphism defined over $\mathbb{F}_q$, called the *Frobenius* endomorphism. If q is a power of 2 and $P \in E(\mathbb{F}_{q^n})$, then $\phi(P)$ can be efficiently computed since squaring is substantially faster than multiplication in a characteristic two finite field.

Example 3 (§7.2.3 of [3]). Let $p \equiv 1 \pmod 4$ be a prime, and consider the elliptic curve

$$E_1 \; : \; y^2 = x^3 + ax \tag{1}$$

defined over $\mathbb{F}_p$. Let $\alpha \in \mathbb{F}_p$ be an element of order 4. Then the map $\phi : E_1 \to E_1$ defined by $(x, y) \mapsto (-x, \alpha y)$ and $\mathcal{O} \mapsto \mathcal{O}$ is an endomorphism defined over $\mathbb{F}_p$. If $P \in E(\mathbb{F}_p)$ is a point of prime order n, then ϕ acts on $\langle P \rangle$ as a multiplication map $[\lambda]$, i.e., $\phi(Q) = \lambda Q$ for all $Q \in \langle P \rangle$, where λ is an integer satisfying $\lambda^2 \equiv -1 \pmod n$. Note that $\phi(Q)$ can be computed using only one multiplication in $\mathbb{F}_p$.

Example 4 (§7.2.3 of [3]). Let $p \equiv 1 \pmod 3$ be a prime, and consider the elliptic curve

$$E_2 \; : \; y^2 = x^3 + b \tag{2}$$

defined over $\mathbb{F}_p$. Let $\beta \in \mathbb{F}_p$ be an element of order 3. Then the map $\phi : E_2 \to E_2$ defined by $(x, y) \mapsto (\beta x, y)$ and $\mathcal{O} \mapsto \mathcal{O}$ is an endomorphism defined over $\mathbb{F}_p$. If $P \in E(\mathbb{F}_p)$ is a point of prime order n, then ϕ acts on $\langle P \rangle$ as a multiplication map $[\lambda]$, where λ is an integer satisfying $\lambda^2 + \lambda \equiv -1 \pmod n$. Note that $\phi(Q)$ can be computed using only one multiplication in $\mathbb{F}_p$.

Example 5 (§7.2.3 of [3]). Let $p > 3$ be a prime such that -7 is a perfect square in $\mathbb{F}_p$, and let $\omega = (1 + \sqrt{-7})/2$, and let $a = (\omega - 3)/4$. Consider the elliptic curve

$$E_3 \;:\; y^2 = x^3 - \frac{3}{4}x^2 - 2x - 1 \tag{3}$$

defined over $\mathbb{F}_p$. Then the map $\phi : E_3 \to E_3$ defined by

$$(x, y) \;\mapsto\; \left(\omega^{-2}\frac{x^2 - \omega}{x - a},\; \omega^{-3}y\frac{x^2 - 2ax + \omega}{(x - a)^2}\right)$$

and $\mathcal{O} \mapsto \mathcal{O}$ is an endomorphism defined over $\mathbb{F}_p$. Computing the endomorphism is a little harder than doubling a point.

Example 6 (§14B of [5]). Let $p > 3$ be a prime such that -2 is a perfect square in $\mathbb{F}_p$, and consider the elliptic curve

$$E_4 \;:\; y^2 = 4x^3 - 30x - 28 \tag{4}$$

defined over $\mathbb{F}_p$. Then the map $\phi : E_4 \to E_4$ defined by

$$(x, y) \;\mapsto\; \left(-\frac{2x^2 + 4x + 9}{4(x + 2)},\; -\frac{2x^2 + 8x - 1}{4\sqrt{-2}(x + 2)^2}y\right)$$

and $\mathcal{O} \mapsto \mathcal{O}$ is an endomorphism defined over $\mathbb{F}_p$. Computing the endomorphism is a little harder than doubling a point.

The existing methods [11, 12, 18, 26, 29] for point multiplication which exploit efficiently-computable endomorphism all use the Frobenius endomorphism. Let E be an elliptic curve defined over a small field $\mathbb{F}_q$, and let ϕ be the Frobenius endomorphism. To compute kP, where $P \in E(\mathbb{F}_{q^n})$, these methods first compute $k' = k \bmod (\phi^n - 1)$ in the ring $\mathbb{Z}[\phi]$. Then, one computes a ϕ-adic expansion $k' = \sum_{i=0}^{t} c_i\phi^i$, where the c_i are elements of a small set, e.g., $\{-q/2, \ldots, q/2\}$, and $t \approx n$. Finally, kP can be efficiently computed as follows:

$$kP \;=\; k'P \;=\; \sum_{i=0}^{t} c_i\phi^i(P). \tag{5}$$

The expression (5) can be evaluated using traditional windowing techniques. Observe that the (slow) point doublings in traditional repeated add-and-double algorithms have been replaced by (fast) evaluations of the Frobenius map.

The methods based on Frobenius map expansions can in principle be extended to an arbitrary endomorphism ψ. However, these techniques will no longer be efficient if computing ψ is more expensive than a point doubling. Furthermore, one may not have $\psi^m - 1 = 0$, so the ψ-adic expansion of k may be significantly longer than the binary expansion of k. Finally, the existing techniques do not apply when $\mathrm{Norm}(\psi) = 1$ (as is the case in Examples 3 and 4) since these techniques require a division operation by ψ which yield a remainder r having norm less than $\mathrm{Norm}(\psi)$.

In the next section, we present a new method which exploits efficiently-computable endomorphisms such as the ones in Examples 3, 4, 5 and 6 to speed up point multiplication.

3 Using Efficient Endomorphisms

Let E be an elliptic curve defined over $\mathbb{F}_q$. Let ϕ be an efficiently-computable endomorphism defined over $\mathbb{F}_q$. To be a little more precise, "efficient" here means a few operations in the underlying field $\mathbb{F}_q$. Let $P \in E(\mathbb{F}_q)$ be a point of prime order n. The map ϕ acts on $\langle P \rangle$ as a multiplication map $[\lambda]$, where λ is a root of the characteristic polynomial of ϕ modulo n.

The problem we consider is that of computing kP for k selected uniformly at random from the interval $[1, n-1]$. Suppose that we can efficiently write $k = k_1 + k_2\lambda \bmod n$, where $k_1, k_2 \in [0, \lfloor\sqrt{n}\rfloor]$ (see §4). Then we have

$$\begin{aligned} kP &= (k_1 + k_2\lambda)P \\ &= k_1P + k_2(\lambda P) \\ &= k_1P + k_2\phi(P). \end{aligned} \tag{6}$$

Then (6) can be computed using windowed simultaneous multiple exponentiation which we review below. In the following, $(k_{t-1}, \ldots, k_1, k_0)_2$ denotes the binary representation of the integer k, and w is the window width.

Algorithm 1. Simultaneous multiple point multiplication

INPUT: w, $k = (k_{t-1}, \ldots, k_1, k_0)_2$, $l = (l_{t-1}, \ldots, l_1, l_0)_2$, P, Q.
OUTPUT: $kP + lQ$.

1. Compute $iP + jQ$ for all $i, j \in [0, 2^w - 1]$.
2. Write $k = (k^{d-1}, \ldots, k^1, k^0)$ and $l = (l^{d-1}, \ldots, l^1, l^0)$ where each k^i and l^i is a bitstring of length w, and $d = \lceil t/w \rceil$.
3. $R \leftarrow \mathcal{O}$.
4. For i from $d-1$ downto 0 do
   4.1 $R \leftarrow 2^w R$.
   4.2 $R \leftarrow R + (k^i P + l^i Q)$.
5. Return(R).

Analysis. Since the bitlengths of k_1 and k_2 in (6) are half the bitlength of k, we might expect to obtain a significant speedup because we have eliminated a significant number of point doublings at the expense of a few point additions. A precise analysis is complicated due to the large number of point multiplication techniques available. Nevertheless the following provides some indication of the relative benefits of our method.

Assume that k is a randomly selected t-bit integer. When $t = 160$, the best general method for computing kP is the signed sliding window method which costs approximately 157 point doubles and 34 point additions using windows of size 4 [16]. The optimum version of Algorithm 1 for computing $kP = k_1 P + k_2 \phi(P)$ costs approximately 79 point doubles and 38 point additions using windows of size 3 [16] plus 1 evaluation of the map ϕ. If the cost of a point doubling is 8 field multiplications and the cost of a point addition is 11 field multiplications (as is the case with Jacobian coordinates [2]), then the ratio of the running times of the new method to the general method is ≈ 0.66. Thus the new method for point multiplication is roughly 50% faster than the best general methods when $t = 160$. This speedup grows as the bitlength of k increases.

Remark. If computing ϕ is cheaper than a point addition, then a few additions can be saved as follows. In the windowed methods [16] for computing $rP + sQ$, we initially compute and store points ϵP and ϵQ for small values of ϵ. If $Q = \phi(P)$, and computing ϕ is cheaper than a point addition, then we can instead compute $\epsilon Q = \epsilon\phi(P) = \phi(\epsilon P)$. For example, in the width-3 windowed method of [16], computing $k_1 P + k_2 \phi(P)$ saves 3 additions at the expense of 3 additional applications of ϕ.

Example 7. An example of an elliptic curve for which our new method is applicable is

$$E\ :\ y^2 = x^3 + 3$$

over the prime field $\mathbb{F}_p$, where

$$p = 1461501637330902918203684832716283019655932313743$$

is a 160-bit prime, and

$$\#E(\mathbb{F}_p) = 1461501637330902918203687013445034429194588307251$$

is prime. This curve is included in the WAP specification of the WTLS protocol [30].

4 Decomposing k

In this section we describe an algorithm which takes as input integers n, λ and $k \in_R [1, n-1]$, and returns integers k_1 and k_2 such that $k \equiv k_1 + k_2\lambda \pmod{n}$. The integers k_1 and k_2 returned are distinguished in that they are both small or, equivalently, the vector $(k_1, k_2) \in \mathbb{Z} \times \mathbb{Z}$ has small Euclidean norm. The term "small" will be made precise below.

Let $G = \mathbb{Z} \times \mathbb{Z}$ and consider the homomorphism $f : G \to \mathbb{Z}_n$ defined by $(i, j) \mapsto (i + \lambda j) \bmod n$. We wish to find a short vector $u \in G$ such that $f(u) = k$; the components of u can then be used as the required k_1 and k_2. Note that it is easy to find a vector $v \in G$ such that $f(v) = k$; $v = (k, 0)$ is such a vector. The problem is in finding a vector that is also short.

Our approach is the following. We first find linearly independent short vectors $v_1, v_2 \in G$ such that $f(v_1) = f(v_2) = 0$. We then find a vector v in the integer lattice generated by v_1 and v_2 that is close to $(k, 0)$. It then follows that $u = (k, 0) - v$ is a short vector with $f(u) = f((k, 0)) - f(v) = k$. Note that both subproblems can be solved using lattice basis reduction algorithms. However, the direct methods presented here are far less cumbersome to implement.

Finding v_1 and v_2. The problem of finding two independent short vectors v_1, v_2 such that $f(v_1) = f(v_2) = 0$ can be solved using the extended Euclidean algorithm. We apply the extended Euclidean algorithm to find the greatest common divisor of n and λ. (This gcd is 1 since n is prime.) The algorithm produces a sequence of equations

$$s_i n + t_i \lambda = r_i, \text{ for } i = 0, 1, 2, \dots, \tag{7}$$

where $s_0 = 1$, $t_0 = 0$, $r_0 = n$, $s_1 = 0$, $t_1 = 1$, $r_1 = \lambda$, and $r_i \geq 0$ for all i. The following properties of the extended Euclidean algorithm can be easily proven by induction.

Lemma 1. *Let s_i, t_i, r_i be the sequence of variables in (7) produced by an application of the extended Euclidean algorithm to positive integers n and λ.*

(i) $r_i > r_{i+1} \geq 0$ *for all* $i \geq 0$.
(ii) $|s_i| < |s_{i+1}|$ *for* $i \geq 1$.
(iii) $|t_i| < |t_{i+1}|$ *for* $i \geq 0$.
(iv) $r_{i-1}|t_i| + r_i|t_{i-1}| = n$ *for all* $i \geq 1$.

Let m be the greatest index for which $r_m \geq \sqrt{n}$. Then $r_m|t_{m+1}| + r_{m+1}|t_m| = n$, and $|t_{m+1}| < \sqrt{n}$. We take $v_1 = (r_{m+1}, -t_{m+1})$. By (7) we have $f(v_1) = 0$. Also, since $|t_{m+1}| < \sqrt{n}$ and $|r_{m+1}| < \sqrt{n}$, we have $||v_1|| \leq 2\sqrt{n}$. We also take v_2 to be the shorter of $(r_m, -t_m)$ and $(r_{m+2}, -t_{m+2})$. Again by (7), we have $f(v_2) = 0$. Heuristically, one expects that v_2 is also short, but we were not able to prove this. Observe that v_1 and v_2 are linearly independent since otherwise if $v_2 = (r_m, -t_m)$ (say), then

$$\frac{r_{m+1}}{r_m} = \frac{-t_{m+1}}{-t_m} = \frac{t_{m+1}}{t_m};$$

but $r_{m+1}/r_m < 1$ by Lemma 1(i) and $|t_{m+1}/t_m| > 1$ by Lemma 1(iii).

Notice that since v_1 and v_2 only depend on n and λ (and not on k), they can be precomputed if n and λ are shared domain parameters.

Finding $\boldsymbol{v}$. A vector v in the integer lattice generated by v_1 and v_2 that is close to $(k, 0)$ can be easily found using elementary linear algebra. By considering $(k, 0)$, v_1 and v_2 as vectors in $\mathbb{Q} \times \mathbb{Q}$, we can write $(k, 0) = \beta_1 v_1 + \beta_2 v_2$, where $\beta_1, \beta_2 \in \mathbb{Q}$. Then round β_1, β_2 to the nearest integers: $b_1 = \lfloor \beta_1 \rceil$, $b_2 = \lfloor \beta_2 \rceil$. Finally, let $v = b_1 v_1 + b_2 v_2$.

The following proves that the vector u is indeed short.

Lemma 2. *The vector $u = (k, 0) - v$, where v is constructed as above, has norm at most max$(||v_1||, ||v_2||)$.*

Proof. We have

$$\begin{aligned} u &= (k,0) - v \\ &= (k,0) - (b_1 v_1 + b_2 v_2) \\ &= (k,0) - [(\beta_1 v_1 + \beta_2 v_2) + (b_1 - \beta_1) v_1 + (b_2 - \beta_2) v_2] \\ &= (\beta_1 - b_1) v_1 + (\beta_2 - b_2) v_2. \end{aligned}$$

Finally, since $|\beta_1 - b_1| \le \frac{1}{2}$ and $|\beta_2 - b_2| \le \frac{1}{2}$, by the Triangle Inequality we have

$$\begin{aligned} ||u|| &\le \tfrac{1}{2}||v_1|| + \tfrac{1}{2}||v_2|| \\ &\le \max(||v_1||, ||v_2||). \end{aligned}$$

□

5 Security Considerations

Elliptic curves having efficiently-computable endomorphisms should be regarded as "special" elliptic curves. As with any "special" instance of a cryptographic scheme, there is always the chance that an attack will be forthcoming that applies to the special instance and significantly weakens the security. Such is the case with low encryption-exponent RSA, and when using small subgroups hidden in a larger group (DSA).

When selecting an elliptic curve E over $\mathbb{F}_q$ for cryptographic use, one must ensure that the order $\#E(\mathbb{F}_q)$ of the elliptic curve is divisible be a large prime number n (say $n \ge 2^{160}$) in order to prevent the Pohlig-Hellman [20] and Pollard's rho [21] attacks. In addition, one must ensure that $\#E(\mathbb{F}_q) \ne q$ in order to prevent the Semaev-Satoh-Araki-Smart attack [23, 22, 26], and that n does not divide $q^k - 1$ for all $k \ge 20$ in order to prevent the Weil pairing [14] and Tate pairing attacks [6]. Given a curve satisfying these conditions, there is no attack know that significantly reduces the time required to compute elliptic curve discrete logarithms. Many such curves having efficient endomorphisms exist and hence appear suitable for cryptographic use. The best line of attack on the elliptic curve discrete logarithm problem on such curves seems to be one based on [7] and [31]. The application of such ideas does not reduce the time to compute a logarithm by more than a small constant factor.

The number of curves for which this technique applies seems to be reasonably large since there are at least 2 candidate curves (see Examples 3, 4, 5 and 6) for each prime p.

6 Conclusions and Further Work

We described a new method for accelerating point multiplication on classes of elliptic curves that have efficiently-computable endomorphisms. The new method for point multiplication is roughly 50% faster than the best general methods. One advantage of the new method is that it is applicable to a larger class of curves than previous such methods. For example, the method is applicable to classes of curves over prime fields and, in particular, is well suited to two curves over prime fields included in the WAP WTLS specification.

One direction in which our method can be generalized is to use higher powers of the endomorphism. For example, one could write $k = k_1 + k_2\lambda + k_3\lambda^2 \bmod n$ for $t/3$-bit integers k_1, k_2, k_3. This could be done by first finding three linearly independent vectors v_1, v_2, v_3 in $\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ of norm roughly $n^{1/3}$, and lying in the kernel of the homomorphism $f : \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ defined by $(x, y, z) \mapsto x + y\lambda + z\lambda^2 \bmod n$. Matters are complicated by the fact that the endomorphism of an elliptic curve satisfies a quadratic equation. For example, for the curves $y^2 = x^3 + b$ over $\mathbb{F}_p$, $\lambda^2 + \lambda + 1 \equiv 0 \pmod{n}$ and so $k_1 + k_2\lambda + k_3\lambda^2 \equiv (k_1 - k_3) + (k_2 - k_3)\lambda \pmod{n}$. Thus simply choosing k_1, k_2, k_3 randomly in $[0, n^{1/3}]$ will result in a k having considerably less entropy than desired.

Acknowledgements

The authors would like to thank Charles Lam, Alfred Menezes, and John Proos for several very helpful comments and suggestions.

References

1. D. Bailey and C. Paar, "Optimal extension fields for fast arithmetic in public-key algorithms", *Advances in Cryptology – Crypto '98*, 1998, 472-485.
2. D. Chudnovsky and G. Chudnovsky, "Sequences of numbers generated by addition in formal groups and new primality and factoring tests", *Advances in Applied Mathematics*, **7** (1987), 385-434.
3. H. Cohen, *A Course in Computational Algebraic Number Theory*, Springer-Verlag, 3rd printing, 1996.
4. H. Cohen, A. Miyaji and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates", *Advances in Cryptology–Asiacrypt '98*, 1998, 51-65.
5. D. Cox, *Primes of the Form $x^2 + ny^2$. Fermat, Class Field Theory and Complex Multiplication*, Wiley, 1989.
6. G. Frey and H. Rück, "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves", *Mathematics of Computation*, **62** (1994), 865-874.

7. R. Gallant, R. Lambert and S. Vanstone, "Improving the parallelized Pollard lambda search on binary anomalous curves", to appear in *Mathematics of Computation.*
8. D. Gollmann, Y. Han and C. Mitchell, "Redundant integer representations and fast exponentiation", *Designs, Codes and Cryptography*, **7** (1996), 135-151.
9. D. Gordon, "A survey of fast exponentiation methods", *Journal of Algorithms*, **27** (1998), 129-146.
10. D. Hankerson, J. Hernandez and A. Menezes, "Software implementation of elliptic curve cryptography over binary fields", *Proceedings of CHES 2000*, to appear.
11. N. Koblitz, "CM-curves with good cryptographic properties", *Advances in Cryptology – Crypto '91*, 1992, 279-287.
12. N. Koblitz, "An elliptic curve implementation of the finite field digital signature algorithm", *Advances in Cryptology – Crypto '98*, 1998, 327-337.
13. C. Lim and P. Lee, "More flexible exponentiation with precomputation", *Advances in Cryptology – Crypto '94*, 1994, 95-107.
14. A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field", *IEEE Transactions on Information Theory*, **39** (1993), 1639-1646.
15. A. Menezes, P. van Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.
16. A. Miyaji, T. Ono and H. Cohen, "Efficient elliptic curve exponentiation", *Proceedings of ICICS '97*, 1997, 282-290.
17. F. Morain and J. Olivos, "Speeding up the computations on an elliptic curve using addition-subtraction chains", *Informatique Théorique et Applications*, **24** (1990), 531-544.
18. V. Müller, "Fast multiplication in elliptic curves over small fields of characteristic two", *Journal of Cryptology*, **1** (1998), 219-234.
19. P. van Oorschot and M. Wiener, "Parallel collision search with cryptanalytic applications", *Journal of Cryptology*, **12** (1999), 1-28.
20. S. Pohlig and M. Hellman, "An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance", *IEEE Transactions on Information Theory*, **24** (1978), 106-110.
21. J. Pollard, "Monte Carlo methods for index computation mod p", *Mathematics of Computation*, **32** (1978), 918-924.
22. T. Satoh and K. Araki, "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves", *Commentarii Mathematici Universitatis Sancti Pauli*, **47** (1998), 81-92.
23. I. Semaev, "Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p", *Mathematics of Computation*, **67** (1998), 353-356.
24. J. Silverman, *The Arithmetic of Elliptic Curves*, Springer-Verlag, 1986.
25. N. Smart, "Elliptic curve cryptosystems over small fields of odd characteristic", *Journal of Cryptology*, **12** (1999), 141-151.
26. N. Smart, "The discrete logarithm problem on elliptic curves of trace one", *Journal of Cryptology*, **12** (1999), 193-196.
27. J. Solinas, "An improved algorithm for arithmetic on a family of elliptic curves", *Advances in Cryptology – Crypto '97*, 1997, 357-371.
28. J. Solinas, "Generalized Mersenne numbers", Technical Report CORR 99-39, Dept. of C&O, University of Waterloo, 1999.

29. J. Solinas, "Efficient arithmetic on Koblitz curves", *Designs, Codes and Cryptography*, **19** (2000), 195-249.
30. WAP WTLS, *Wireless Application Protocol Wireless Transport Layer Security Specification*, Wireless Application Protocol Forum, February 1999. Drafts available at http://www.wapforum.org
31. M. Wiener and R. Zuccherato, "Faster attacks on elliptic curve cryptosystems", *Selected Areas in Cryptography*, 1999, 190-200.