Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT®
ELSEVIER Applied Mathematics and Computation 151 (2004) 671–678
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

A self-pairing map and its applications to cryptography ☆

Hyang-Sook Lee

Department of Mathematics, Ewha Womans University, Seoul 120-750, South Korea

Abstract

In this paper, we propose a bilinear pairing map on finitely generated free R-modules with rank two where R is a commutative ring with 1, and apply this pairing to elliptic curves over fields. We also verify the pairing on elliptic curves can be applicable to the current cryptographic schemes.

Keywords: Bilinear pairing map; Free modules; Elliptic curves; Cryptography

1. Introduction

Weil pairing and Tate pairing are well-known functions defined on elliptic curves. The bilinearity and non-degeneracy of these pairings are very useful in cryptography. These pairings were first used as cryptanalytic tools to reduce the complexity of the discrete log problem for some weak elliptic curves to the discrete log problem over finite fields [5,9]. This implies the discrete log problem is not intractable for some special types of elliptic curves. These pairings were also used to construct a one round tripartite Diffie–Hellman protocol [6]. The identity based encryption scheme was also constructed based on the Weil pairing [3]. There are a lot of useful results regarding the Weil pairing and Tate pairing in pure mathematics as well as cryptography. The usual application of the bilinear pairing is of the type $G_1 \times G_1 \rightarrow G_2$ where G_1

☆ The author was supported by grant no. R06-2002-012-0100-0 from the research program of KOSEF.

E-mail address: hsl@ewha.ac.kr (H.-S. Lee).

0096-3003/$ - see front matter
doi:10.1016/S0096-3003(03)00369-2

and G_2 are different groups. However, the motivation of this paper is to find a self-bilinear map $G \times G \to G$ which might be useful in cryptographic schemes where G is a group. The self-bilinear map gives a method assigning a pair of elements to the other in the group G. In this paper, we present a new bilinear pairing $A \times A \to A$ where A is a finitely generated free R-module with rank two and R is a commutative ring with 1. We apply the structure of this pairing to the elliptic curves. We also show that the pairing on elliptic curves is applicable to the current cryptographic schemes. In addition, we expect the pairing to be more useful in cryptography or in pure mathematics.

This paper is organized as follows:

In Section 2, we define a self-bilinear pairing on finitely generated free R-modules with rank 2 where R is a commutative ring with 1 and give its properties. In Section 3, we apply this pairing to the elliptic curves over fields. In Section 4, we verify the pairing on elliptic curves is applicable to the current cryptographic schemes such as an authenticated key agreement protocol and a digital signature algorithm.

2. A self-bilinear pairing on modules

In this section we construct a bilinear pairing on finitely generated free R-modules with rank 2 where R is a commutative ring with 1.

Let R be a commutative ring with 1, A be a finitely generated free R-module with rank 2 and (g,h) be a generating pair for A. We consider elements $a = r_1g + s_1h,\ b = r_2g + s_2h,\ c = r_3g + s_3h$ in A, where $r_1, r_2, r_3, s_1, s_2, s_3 \in R$. For some fixed $\alpha, \beta \in R$, we define a pairing map

$$\mathscr{L}_{\alpha,\beta} : A \times A \to A$$

by $\mathscr{L}_{\alpha,\beta}(a,b) = (r_1s_2 - s_1r_2)(\alpha g + \beta h)$. If $\alpha = \beta = 0$, $\mathscr{L}_{\alpha,\beta}$ is trivial. Thus we exclude the case where both α and β are zero. If $a = \bar{a}$ and $b = \bar{b}$, then we have $r_1 = \bar{r}_1$, $s_1 = \bar{s}_1$, $r_2 = \bar{r}_2$ and $s_2 = \bar{s}_2$ by independency of g and h. This implies $\mathscr{L}_{\alpha,\beta}(a,b) = \mathscr{L}_{\alpha,\beta}(\bar{a},\bar{b})$. Therefore the map is well defined.

Proposition 2.1. *The pairing $\mathscr{L}_{\alpha,\beta}$ has the following properties*:

(i) *Identity*: *For all $a \in A$, $\mathscr{L}_{\alpha,\beta}(a,a) = 0$.*
(ii) *Bilinearity*: *For all $a,b,c \in A$,*

$$\mathscr{L}_{\alpha,\beta}(a+b,c) = \mathscr{L}_{\alpha,\beta}(a,c) + \mathscr{L}_{\alpha,\beta}(b,c),$$
$$\mathscr{L}_{\alpha,\beta}(a,b+c) = \mathscr{L}_{\alpha,\beta}(a,b) + \mathscr{L}_{\alpha,\beta}(a,c).$$

(iii) *Anti-symmetry*: *For all $a,b \in A$, $\mathscr{L}_{\alpha,\beta}(a,b) = -\mathscr{L}_{\alpha,\beta}(b,a)$.*
(iv) *Non-degeneracy*: *If $a \in A$, $\mathscr{L}_{\alpha,\beta}(a,0) = 0$. Moreover, if $\mathscr{L}_{\alpha,\beta}(a,b) = 0$ for all $b \in A$, then $a = 0$.*

Proof

(i) For all $a \in A$, $\mathscr{L}_{\alpha,\beta}(a,a) = (r_1 s_1 - s_1 r_1)(\alpha g + \beta h) = 0$.

(ii) For all $a, b, c \in A$,

$$\begin{aligned}\mathscr{L}_{\alpha,\beta}(a+b,c) &= \mathscr{L}_{\alpha,\beta}((r_1+r_2)g + (s_1+s_2)h, r_3 g + s_3 h)\\ &= ((r_1+r_2)s_3 - (s_1+s_2)r_3)(\alpha g + \beta h)\\ &= (r_1 s_3 - s_1 r_3)(\alpha g + \beta h) + (r_2 s_3 - s_2 r_3)(\alpha g + \beta h)\\ &= \mathscr{L}_{\alpha,\beta}(a,c) + \mathscr{L}_{\alpha,\beta}(b,c),\end{aligned}$$

$$\begin{aligned}\mathscr{L}_{\alpha,\beta}(a,b+c) &= \mathscr{L}_{\alpha,\beta}(r_1 g + s_1 h, (r_2+r_3)g + (s_2+s_3)h)\\ &= (r_1(s_2+s_3) - s_1(r_2+r_3))(\alpha g + \beta h)\\ &= (r_1 s_2 - s_1 r_2)(\alpha g + \beta h) + (r_1 s_3 - s_1 r_3)(\alpha g + \beta h)\\ &= \mathscr{L}_{\alpha,\beta}(a,b) + \mathscr{L}_{\alpha,\beta}(a,c).\end{aligned}$$

(iii) From (ii) we have $\mathscr{L}_{\alpha,\beta}(a+b,a+b) = \mathscr{L}_{\alpha,\beta}(a,a) + \mathscr{L}_{\alpha,\beta}(a,b) + \mathscr{L}_{\alpha,\beta}(b,a) + \mathscr{L}_{\alpha,\beta}(b,b)$. By using (i), we obtain

$$\mathscr{L}_{\alpha,\beta}(a,b) = -\mathscr{L}_{\alpha,\beta}(b,a).$$

(iv) For any $a \in A$, $\mathscr{L}_{\alpha,\beta}(a,0) = (r_1 \cdot 0 - s_1 \cdot 0)(\alpha g + \beta h) = 0$. If $\mathscr{L}_{\alpha,\beta}(a,b) = (r_1 s_2 - s_1 r_2)(\alpha g + \beta h) = 0$ for all $b \in A$, then $\frac{r_1}{s_1} = \frac{r_2}{s_2}$ for all (r_2, s_2). This implies $r_1 = s_1 = 0$. Therefore $a = 0$. □

3. A self-bilinear pairing on elliptic curves

In this section, we apply the pairing on elliptic curve groups over fields.

Let K be a field with characteristic zero or a prime p. Let $E = E(\bar{K})$ be an elliptic curve over $\bar{K}$ where $\bar{K}$ is an algebraic closure of K. We say E is a torsion group if for each point $P \in E$, there is a positive integer l such that $lP = O$. The smallest such integer is called the order of P. An n-torsion point is a point $P \in E(\bar{K})$ satisfying $nP = O$. Let $E(K)[n]$ denote the subgroup of n-torsion points in $E(K)$, where $n \neq 0$. We write $E[n]$ for $E(\bar{K})[n]$. We assume $\mathrm{char}(K) = 0$ or p, where p is relatively prime to n. Then $E[n] \approx Z_n \oplus Z_n$.

Let $\{S, T\}$ be some fixed generating pair for $E[n]$. Then any point can be expressed by these generating points S and T. We consider the points $P = a_1 S + b_1 T$, $Q = a_2 S + b_2 T$ and $R = a_3 S + b_3 T$ in $E[n]$, where a_1, a_2, a_3, b_1, b_2 and b_3 are integers in $[0, n-1]$. For some fixed integers $\alpha, \beta \in [0, n-1]$, we define a function

$$\mathscr{L}^n_{\alpha,\beta} : E[n] \times E[n] \to E[n]$$

by $\mathscr{L}^n_{\alpha,\beta}(P,Q) = (a_1b_2 - b_1a_2)(\alpha S + \beta T)$. We exclude the trivial case where both α and β are zero. If $P = \bar{P}$ and $Q = \bar{Q}$, then $\mathscr{L}^n_{\alpha,\beta}(P,Q) = \mathscr{L}^n_{\alpha,\beta}(\bar{P},\bar{Q})$. Obviously $\mathscr{L}^n_{\alpha,\beta}(P,Q) \in E[n]$.

Proposition 3.1. *The $\mathscr{L}^n_{\alpha,\beta}$-pairing satisfies the followings*:

(i) *Identity*: *For all* $P \in E[n]$, $\mathscr{L}^n_{\alpha,\beta}(P,P) = O$.
(ii) *Bilinearity*: *For all* $P,Q,R \in E[n]$,

$$\mathscr{L}^n_{\alpha,\beta}(P+Q,R) = \mathscr{L}^n_{\alpha,\beta}(P,R) + \mathscr{L}^n_{\alpha,\beta}(Q,R),$$

$$\mathscr{L}^n_{\alpha,\beta}(P,Q+R) = \mathscr{L}^n_{\alpha,\beta}(P,Q) + \mathscr{L}^n_{\alpha,\beta}(P,R).$$

(iii) *Anti-symmetry*: *For all* $P,Q \in E[n]$, $\mathscr{L}^n_{\alpha,\beta}(P,Q) = -\mathscr{L}^n_{\alpha,\beta}(Q,P)$.
(iv) *Non-degeneracy*: *If* $P \in E[n]$, $\mathscr{L}^n_{\alpha,\beta}(P,O) = O$. *Moreover, if* $\mathscr{L}^n_{\alpha,\beta}(P,Q) = O$ *for all* $Q \in E[n]$, *then* $P = O$.
(v) *Compatibility*: *If* $P \in E[nn']$ *and* $Q \in E[n]$, *then* $\mathscr{L}^{nn'}_{\alpha,\beta}(P,Q) = \mathscr{L}^n_{\alpha,\beta}(n'P,Q)$. *Also if* $P \in E[n]$ *and* $Q \in E[nn']$, *then* $\mathscr{L}^{nn'}_{\alpha,\beta}(P,Q) = \mathscr{L}^n_{\alpha,\beta}(P,n'Q)$.

Proof

(i) For all $P \in E[n]$, $\mathscr{L}^n_{\alpha,\beta}(P,P) = (a_1b_1 - b_1a_1)(\alpha S + \beta T) = O$.
(ii) For all $P,Q,R \in E[n]$,

$$\begin{aligned}\mathscr{L}^n_{\alpha,\beta}(P+Q,R) &= \mathscr{L}^n_{\alpha,\beta}((a_1+a_2)S + (b_1+b_2)T, a_3S + b_3T)\\ &= ((a_1+a_2)b_3 - (b_1+b_2)a_3)(\alpha S + \beta T)\\ &= (a_1b_3 - b_1a_3)(\alpha S + \beta T) + (a_2b_3 - b_2a_3)(\alpha S + \beta T)\\ &= \mathscr{L}^n_{\alpha,\beta}(P,R) + \mathscr{L}^n_{\alpha,\beta}(Q,R),\end{aligned}$$

$$\begin{aligned}\mathscr{L}^n_{\alpha,\beta}(P,Q+R) &= \mathscr{L}^n_{\alpha,\beta}(a_1S + b_1T, (a_2+a_3)S + (b_2+b_3)T)\\ &= (a_1(b_2+b_3) - b_1(a_2+a_3))(\alpha S + \beta T)\\ &= (a_1b_2 - b_1a_2)(\alpha S + \beta T) + (a_1b_3 - b_1a_3)(\alpha S + \beta T)\\ &= \mathscr{L}^n_{\alpha,\beta}(P,Q) + \mathscr{L}^n_{\alpha,\beta}(P,R).\end{aligned}$$

(iii) From (ii) we have $\mathscr{L}^n_{\alpha,\beta}(P+Q,P+Q) = \mathscr{L}^n_{\alpha,\beta}(P,P) + \mathscr{L}^n_{\alpha,\beta}(P,Q) + \mathscr{L}^n_{\alpha,\beta}(Q,P) + \mathscr{L}^n_{\alpha,\beta}(Q,Q)$. By using (i), we obtain

$$\mathscr{L}^n_{\alpha,\beta}(P,Q) = -\mathscr{L}^n_{\alpha,\beta}(Q,P).$$

(iv) For any $P \in E[n]$, $\mathscr{L}^n_{\alpha,\beta}(P,O) = \mathscr{L}^n_{\alpha,\beta}(a_1S + b_1T, 0\cdot S + 0\cdot T) = a_1\cdot 0 - b_1\cdot 0)(\alpha S + \beta T) = O$. If $\mathscr{L}^n_{\alpha,\beta}(P,Q) = (a_1b_2 - b_1a_2)(\alpha S + \beta T) = O$ for all $Q \in E[n]$, then $(a_1/b_1) = (a_2/b_2)$ for all (a_2,b_2). This implies $a_1 = b_1 = 0$. Therefore $P = O$.
(v) If (S,T) is a generating pair for $E[nn']$, then $(n'S, n'T)$ is a generating pair for $E[n]$. Let $P = a_1S + b_1T \in E[nn']$. Then $n'P = a_1(n'S) + b_1(n'T) \in E[n]$.

If $Q = a_2S + b_2T \in E[n]$, then $Q = a'_2(n'S) + b'_2(n'T)$ where $a_2 = a'_2n'$, $b_2 = b'_2n'$. Therefore

$$\begin{aligned}\mathscr{L}^{nn'}_{\alpha,\beta}(P,Q) &= (a_1b_2 - b_1a_2)(\alpha S + \beta T) = (a_1b'_2n' - b_1a'_2n')(\alpha S + \beta T)\\ &= (a_1b'_2 - b_1a'_2)(\alpha(n'S) + \beta(n'T)) = \mathscr{L}^{n}_{\alpha,\beta}(n'P,Q).\end{aligned}$$

It is similar to show $\mathscr{L}^{nn'}_{\alpha,\beta}(P,Q) = \mathscr{L}^{n}_{\alpha,\beta}(P,n'Q)$. □

4. Application of $\mathscr{L}$-pairing to cryptography

A *protocol* is a multi-party algorithm, defined by a sequence of steps precisely specifying the actions required by two or more parties in order to achieve a specified objective. *Key establishment* is a process or protocol whereby a shared secret becomes available to two or more parties, for subsequent cryptographic use. A *key agreement protocol* is a key establishment technique in which a shared secret is derived by two (or more) parties as a function of information contributed by, or associated with, each of these, such that no party can predetermine the resulting value. The key agreement protocol is *contributory* if each party equally contributes to the key and guarantees its freshness. *Key authentication* is the property whereby one party is associated that no other party aside from a specially identified second party may gain access to a particular secret key. Key authentication is said to be *implicit* if each party sharing the key is assured that no other party can learn the secret shared key. We refer to [10] for more information.

Let E be an elliptic curve over a finite field F_q. The *elliptic curve discrete log problem* (ECDLP) in $E(F_q)$ is the following: Given $P \in E(F_q)$ with order n, $Q \in \langle P\rangle$, find r such that $Q = rP$. The *Diffie–Hellman problem for elliptic curves* is computing abP for given P, aP, bP. The following is the well-known Diffie–Hellman [4] key exchange protocol for elliptic curves.

Diffie–Hellman Key Exchange Protocol. Alice and Bob fix an elliptic curve $E(F_q)$ and a generator $P \in E(F_q)$. They select random integers a, b and exchange aP, bP, respectively. Then Alice and Bob, both obtain the common secret key abP. We refer to [7,8,11] to understand the elliptic curve cryptosystem.

We consider an elliptic curve E over F_q, where $q = p^r$ and p is a large enough prime for the discrete logarithm in F_q to be hard and nobody knows how to compute discrete logarithm on the elliptic curve. Choose a large prime l such that $E[l] \subseteq E(F_{q^k})$ for some smallest integer k. We choose a generating pair $\{P,Q\}$ in $E[l]$ and integers $\alpha,\beta \in [0, l-1]$ which determine the $\mathscr{L}^{l}_{\alpha,\beta}$-pairing. The parameters $(P,Q,\mathscr{L}^{l}_{\alpha,\beta})$ are publicly known. Let $h : E(F_q) \to Z/l$ and $H : \{0,1\}^n \to Z/l$ be hash functions. Now we apply our $\mathscr{L}^{l}_{\alpha,\beta}$-pairing to the cryptographic schemes such as authenticated key agreement and digital signature algorithm on elliptic curves. Using $\mathscr{L}^{l}_{\alpha,\beta}$-pairing in cryptography is

based on the difficulty of computing $\mathscr{L}^l_{\alpha,\beta}(aP,Q)$, $\mathscr{L}^l_{\alpha,\beta}(P,bQ)$ and $\mathscr{L}^l_{\alpha,\beta}(aP,bQ)$, without knowing the secret values a and b by the construction of the $\mathscr{L}^l_{\alpha,\beta}$-pairing. In fact, computing $\mathscr{L}^l_{\alpha,\beta}$-pairing with only public values is as hard as solving the discrete log problem on elliptic curves. Our first protocol (I) is modified from the protocol in [2], and the second scheme (II) is modified from the well-known elliptic curve digital signature algorithm [1]. Our schemes include only one random secret key per user. This is more efficient than using two random secret keys in those known schemes.

We assume that two communication parties Alice(A) and Bob(B) wish to share a common secret information.

(*I*) *Authenticated* 2*-party key agreement* (*A-ECDH*)

(i) Key generation
- A and B choose random secret integers $a,b \in (1,l-1)$ respectively. They compute aP and bP, and then broadcast these values. Thus the public values of the system are $(P,Q,aP,bP,\mathscr{L}^l_{\alpha,\beta})$.

(ii) Transmission
- A computes $\mathscr{L}^l_{\alpha,\beta}(aP,Q)$ and

  $\mathrm{A} \to \mathrm{B} : \mathscr{L}^l_{\alpha,\beta}(aP,Q)$.

- B computes $\mathscr{L}^l_{\alpha,\beta}(bP,Q)$, $K = baP = (x,y)$ and $h(K)$, and

  $\mathrm{B} \to \mathrm{A} : J = h(K)\mathscr{L}^l_{\alpha,\beta}(bP,Q)$.

(iii) Authenticated common key generation
- When A receives $J = h(K)\mathscr{L}^l_{\alpha,\beta}(bP,Q)$, compute $K = abP$ and $h(K)^{-1} (\mathrm{mod}\, l)$. Then the shared secret key computed by A is $S = ah(K)^{-1}J = a\mathscr{L}^l_{\alpha,\beta}(bP,Q)$. Also S computed by B is $b\mathscr{L}^l_{\alpha,\beta}(aP,Q)$. Therefore A and B obtain the common secret key $S = ab\mathscr{L}^l_{\alpha,\beta} \times (P,Q)$. □

The above protocol A-ECDH using $\mathscr{L}$-pairing provides implicit key authentication by the following theorem.

Theorem 4.1. *A-ECDH protocol is a contributed authenticated key agreement protocol.*

Proof. From the construction of the shared key $S = ab\mathscr{L}^l_{\alpha\beta}(P,Q)$, it is clear that A-ECDH protocol is contributory. Let C be an active adversary able to modify, delay, or inject messages.

Attack on B. Let S_B be the shared key computed by B. It can be expressed as $S_\mathrm{B} = b\mathscr{L}^l_{\alpha,\beta}(c_1P,Q)$ where c_1 is a quantity possibly known to C, i.e. C can substitute the first flow of the protocol with $\mathscr{L}^l_{\alpha,\beta}(c_1P,Q)$. Then computing

$b\mathscr{L}^l_{\alpha,\beta}(c_1P,Q)$ requires C to compute $b\mathscr{L}^l_{\alpha,\beta}(P,Q)$. However, the only expression containing $b\mathscr{L}^l_{\alpha,\beta}(P,Q)$ is $h(K)\mathscr{L}^l_{\alpha,\beta}(bP,Q)$ in the second flow. But, computing $b\mathscr{L}^l_{\alpha,\beta}(P,Q)$ from $h(K)\mathscr{L}^l_{\alpha,\beta}(bP,Q)$ is intractable without knowing $K=(x,y)$.

Attack on A. The key computed by A is $S_A = ah(K)^{-1}\mathscr{L}^l_{\alpha,\beta}(c_2P,Q)$ where c_2 is possibly chosen by C. First we suppose that $c_2 = c_3h(K)$ where c_3 is polynomially independent of $h(K)$ and known to C. Then $S_A = ah(K)^{-1}\mathscr{L}^l_{\alpha,\beta}\times(c_3h(K)P,Q) = a\mathscr{L}^l_{\alpha,\beta}(c_3P,Q)$. However, computing $c_3h(K)\mathscr{L}^l_{\alpha\beta}(P,Q)$ such that c_3 is known to C is intractable without computing $h(K)\mathscr{L}^l_{\alpha\beta}(P,Q)$. Next we suppose c_2 is polynomially independent of $h(K)$. Since S_A is a function of $h(K)^{-1}$, it is not computable by C. □

(*II*) *Digital signature algorithm* (*ECDSA*)

(i) Key generation
- A chooses a random secret integer $a \in (1, l-1)$ and computes the public value $aP = (x_1, y_1)$.

(ii) Signature generation: To sign a message m,
- A computes $\mathscr{L}^l_{\alpha,\beta}(aP,Q) = (x_2,y_2)$.
- A computes $r = x_2 (\mathrm{mod}\, l)$. If $r = 0$, then go to step (i).
- A computes $a^{-1}(\mathrm{mod}\, l)$.
- A computes $H(m)$, where $H(m)$ is the hash value of the message.
- A computes $s = a^{-1}(H(m)+r)(\mathrm{mod}\, l)$. If $s = 0$, then go to step (i).
- The signature for the message m is the pair (s,r).

(iii) Signature verification: To verify Alice's signature (r,s) of the message m,
- B verifies that r and s are integers in the interval $[1, l-1]$.
- B computes $w = s^{-1}(\mathrm{mod}\, l)$ and $H(m)$.
- B computes $w(H(m)+r)P = (\bar{x}_1, \bar{y}_1)$. If $\bar{x}_1 \neq x_1 (\mathrm{mod}\, l)$, reject the signature. Otherwise computes $w(H(m)+r)\mathscr{L}^l_{\alpha,\beta}(P,Q) = (x_0,y_0)$. If this value is O, then reject the signature. Otherwise computes $v = x_0(\mathrm{mod}\, l)$.
- Accept the signature if and only if $v = r$. □

Acknowledgements

This paper was written during visiting University of Illinois at Urbana-Champaign. The author thanks the Professors N. Boston, I. Duursma and A. Stein for their hospitalities.

References

[1] ANSI X9.62 and FIPS 186-2, Elliptic Curve Digital Signature Algorithm, 1998.
[2] G. Atenies, M. Steiner, G. Tsudik. Authenticated group key agreement and friends, in: ACM Conference on Computer and Communications Security, 1998.

[3] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, in: Proceedings of Crypto 2001, Lecture Notes in Computer Science, vol. 2139, pp. 213–229.
[4] W. Diffie, M. Hellman, New direction in cryptography, IEEE Transactions on Information Theory IT-22 (6) (1996) 644–654.
[5] G. Frey, H. Ruck, A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves, Mathematics of Computation 62 (1994) 865–874.
[6] A. Joux, A one round protocol for tripartite Diffie–Hellman, in: Lecture Notes in Computer Science, vol. 1838, ANTS 2000, pp. 385–393.
[7] N. Koblitz, Algebraic aspects of cryptography, Springer, Berlin, 1997.
[8] A. Menezes, Elliptic curve public key cryptosystems, Kluwer Academic Publishers, Dordecht, 1994.
[9] A. Menezes, T. Okamoto, S. Vanston, Reducing elliptic curve logarithms to logarithms in a finite field, IEEE Transaction on Information Theory 39 (1993) 1639–1646.
[10] A. Menezes, T. Okamoto, S. Vanston, Handbook of Applied Cryptography, CRC Press, Boca Raton, 1997.
[11] J. Silverman, The Arithmetic of Elliptic Curves, Springer-Verlag, New York, 1985.