Weil Pairing vs. Tate Pairing in IBE systems

Ezra Brown, Eric Errthum, David Fu

October 10, 2003

1. Introduction

Although Boneh and Franklin use the Weil pairing on elliptic curves to create Identity-Based Encryption (IBE) systems [2], they also mention that the Tate pairing could be used in place of the Weil pairing to speed up calculations. Here we first present a simplified IBE system that utilizes some basic properties of the Weil pairing. Next we define the Tate pairing and show how to compute both the Weil and the Tate pairings. Lastly, we present and prove a relationship between the two pairings which is often informally summarized as "The Weil pairing is just two Tate pairings."

2. A Simple IBE Key Establishment

In [2] the Weil pairing on an elliptic curve is used for identity-based key establishment and encryption methods. Here we give a simplified version of the key establishment in [2] for the purpose of motivating a comparison between the relative performances of the Tate and Weil pairings.

2.1. Elliptic Curve Credentials

To start, an elliptic curve, E, is chosen over some field $\mathbb{F}_q$ such that $E(\mathbb{F}_q)$ satisfies the following two properties: (I) the size of $E(\mathbb{F}_q)$, denoted $\#E(\mathbb{F}_q)$, is a small multiple of a large prime r, and (II) there exists a small positive integer d such that $r \mid q^d - 1$. The second property guarantees that the complete r-torsion, $E[r]$, lies in $E(\mathbb{F}_{q^d})$ [1]. Ideally, d should be chosen so that the square-root attack on the discrete-log problem in the order-r subgroup of $E(\mathbb{F}_q)$ and the factor-base attack on the discrete-log problem in the multiplicative group of the field $\mathbb{F}_{q^d}$ are balanced in difficulty.

We will denote by $\mathcal{R}$ the order-r subgroup of $E(\mathbb{F}_q)$. Since $E[r] \subset E(\mathbb{F}_{q^d})$, there exists a point $H \in E(\mathbb{F}_{q^d})$ such that H is r-torsion but $H \notin \mathcal{R}$.

2.2. The Weil Pairing on Elliptic Curves

If A and B are r-torsion points on some elliptic curve $E(\mathbb{F}_{q^d})$, let us denote the r-Weil pairing of A and B by $e_r(A, B)$. Then e_r is a map

$$e_r(\cdot, \cdot) : E[r] \times E[r] \to \mathbb{F}_{q^d}^*.$$

A sound definition of the Weil pairing and an exposition of its properties can be found in [9]. For our purposes, the pivotal property of the Weil pairing is its bilinearity. That is, for E an elliptic curve with $A_0, A_1, B_0, B_1 \in E[r]$:

$$\begin{aligned} e_r(A_0 + A_1, B_0) &= e_r(A_0, B_0) \cdot e_r(A_1, B_0), \\ e_r(A_0, B_0 + B_1) &= e_r(A_0, B_0) \cdot e_r(A_0, B_1). \end{aligned}$$

In particular, for any integer k we have

$$e_r(kA_0, B_0) = e_r(A_0, kB_0) = e_r(A_0, B_0)^k.$$

Another property of the Weil pairing is that the Weil pairing of a point with itself is trivial: $e_r(A_0, A_0) = 1$. This property is not especially important in our example, but it will be used later for comparison to the Tate pairing.

2.3. Identity-Based Key Establishment Set-Up

The identity-based encryption scheme presented here is run by a trusted central authority. It is a master-key system in which the public keys are determined from a user's ID. The central authority chooses a public point H as in §2.1 and a random integer s modulo r which will be the master-key and kept secret. From these, the central authority computes the universal public key

$$U := sH.$$

Upon enrollment in the system, the user is given a long-term key pair, $V, C \in \mathcal{R}$ by the central authority. The public key C is computed from the user's ID in a publicly known way and the private key V is

$$V := sC.$$

2.4. Key Agreement for Alice and Bob

Suppose that Alice and Bob are both users of this IBE system and that Alice wishes to form a key agreement with Bob. Since they are members of the system, Alice and Bob have long-term key pairs (V_A, C_A) and (V_B, C_B), respectively. To begin key establishment, Alice chooses a pseudo-random integer z_A and computes

$$W_A := z_A H.$$

The pair (z_A, W_A) is her one-time key pair, and she transmits W_A to Bob, keeping z_A secret. Now, Alice computes the secret shared value, SSV, as

$$SSV_A = e_r(z_A C_B, U), \tag{1}$$

and Bob computes it as

$$SSV_B = e_r(V_B, W_A). \tag{2}$$

Equations (1) and (2) are equal by the bilinearity of the Weil pairing:

$$\begin{aligned} SSV_B = e_r(V_B, W_A) &= e_r(sC_B, z_A H) \\ &= e_r(z_A C_B, sH) \\ &= e_r(z_A C_B, U) = SSV_A. \end{aligned}$$

Since this IBE scheme hinges on the computation of the Weil pairing, it would be beneficial to find a similar bilinear mapping that can be computed more efficiently. The Tate pairing is a choice candidate.

3. The Tate Pairing

The Tate pairing (also sometimes referred to as the Tate-Lichtenbaum pairing) was first introduced into the world of cryptography by Frey and Ruck [4] as an extension of the work done by Menezes, Okamoto, and Vanstone [7]. It was later mentioned in [2] as a means of speeding up the encryption and decryption schemes. Shortly thereafter, Galbraith released a paper [5] in which he outlined the basics of the Tate pairing and its computation. He also gave some experimental timings and an explanation as to why the Tate pairing is faster to compute than the Weil pairing.

3.1. Definition and Properties

Before proceeding to the Tate pairing, we set some notation. For a point $P \in E$, we let (P) denote the divisor of P. Thus, $(P) + (Q)$ denotes a sum in the divisor group and is a divisor of degree 2, whereas $(P + Q)$ denotes a divisor of degree 1, the point being the elliptic curve sum of the points P and Q. For a rational function g on E and a divisor $D = \sum_{P \in E} n_P P$, we let $g(D)$ denote the value $\prod_{P \in E} g(P)^{n_P}$. We say that two divisors D, D' are *disjoint* if their supports are disjoint; i.e., $n_P \neq 0$ in D implies that $n_P = 0$ in D' and $n_P \neq 0$ in D' implies that $n_P = 0$ in D.

In [5], the Tate pairing is defined as follows: Let $G := E(\mathbb{F}_{q^d})$ where q and d satisfy the requirements in §2.1. Let $\mathcal{O}$ denote the identity of G, commonly known as the "point at infinity." Then the Tate pairing is a mapping

$$< \cdot, \cdot > : E[r] \times G/rG \to \mathbb{F}^*_{q^d}/(\mathbb{F}^*_{q^d})^r. \tag{3}$$

The right-hand-side of (3), $\mathbb{F}_{q^d}^*/(\mathbb{F}_{q^d}^*)^r$, can be thought of as the set of equivalence classes in $\mathbb{F}_{q^d}^*$ where $a \equiv b$ if and only if there exists $c \in \mathbb{F}_{q^d}^*$ such that $a = bc^r$. Galbraith calls this "equivalence modulo rth powers."

For points $P \in E[r]$ and $Q \in G/rG$ we define the Tate pairing to be

$$< P, Q >:= g(D),$$

where g is a rational function with $\mathrm{div}(g) = r((P) - (\mathcal{O}))$, $D \sim (Q) - (\mathcal{O})$ and the supports of D and $\mathrm{div}(g)$ are disjoint.

Since $g(D) \in \mathbb{F}_{q^d}^*$, the value of $< P, Q >$ defines an equivalence class in $\mathbb{F}_{q^d}^*/(\mathbb{F}_{q^d}^*)^r$. However, its definite value is dependent on the choice of D. In most algorithms D has the form $D = (Q + S) - (S)$ for some $S \in G$. In such cases we will denote this dependence by $< P, Q >_S$. To achieve a value in $\mathbb{F}_{q^d}^*$ that is independent of S, $< P, Q >_S$ must be raised to the power $(q^d - 1)/r$ to eliminate rth powers. In fact, some authors (e.g. [3]) define the Tate pairing to be $< P, Q >^{(q^d-1)/r}$ since typically a unique value is needed. For our purposes we will let $t(P, Q)$ denote $< P, Q >^{(q^d-1)/r}$.

The Tate pairing can be shown [4] to be well-defined, non-degenerate, and, most importantly, bilinear, and thus $t(kP, Q) = t(P, kQ) = t(P, Q)^k$. This allows the Tate pairing to be used in place of the Weil pairing in most cryptographic applications. However, one key difference between the Tate pairing and the Weil pairing is that the Tate pairing of a point with itself is not necessarily trivial: $t(P, P) \neq 1$. This fact allowed Frey, Muller and Ruck to use the Tate pairing to reduce the discrete logarithm problem on certain elliptic curves to the discrete logarithm problem over a finite field in cases in which the Weil pairing fails to do so [3].

3.2. Weil and Tate Pairing Computations

In an often cited but never published paper by Miller [8], an algorithm is given for computing a rational function g with $\mathrm{div}(g) = r((P) - (\mathcal{O}))$ for a given $P \in E[r]$. This algorithm uses a "double and add" method to compute $rP = \mathcal{O}$, while along the way constructing a rational function from the lines and verticals that arise. This algorithm can easily be modified [5] to return the Tate pairing, $< P, Q >_S$. The modified algorithm takes as input the point $P \in E[r]$, the point $Q \in G$, and a "good" point $S \in G$. (The limitations on S will be discussed shortly.) The modified algorithm returns the value of g computed at the divisor

$$D = (Q + S) - (S) \sim (Q) - (\mathcal{O}). \tag{4}$$

During the "double and add" method of computing rP, various other multiples of P arise. We will refer to this set of multiples of P as the addition chain of P and denote it by $\mathcal{A}(P) = \{P, a_1P, a_2P, ..., \mathcal{O}\}$. The only restriction on S is that $Q + S, S \notin \mathcal{A}(P)$. Typically, when q is large and the gods are friendly, such an S is easily found.

For $P, Q \in E[r]$ Silverman's alternate definition ([9], Ch.3, Exer.16) is amenable to computing the Weil pairing. Given disjoint divisors, A and B, equivalent to $(P) - (\mathcal{O})$ and $(Q) - (\mathcal{O})$, respectively, the Weil pairing of P and Q is given by

$$e_r(P, Q) = \frac{f_A(B)}{f_B(A)}, \tag{5}$$

where $\mathrm{div}(f_A) = rA$ and $\mathrm{div}(f_B) = rB$.

In [6], Menezes provides a way to compute (5). One first uses Miller's algorithm to produce rational functions f_1 and f_2 such that $\mathrm{div}(f_1) = r((P + T_1) - (\mathcal{O}))$ and $\mathrm{div}(f_2) = r((T_1) - (\mathcal{O}))$ where T_1 is some r-torsion point. Setting $f_A = f_1/f_2$ gives the desired result of $\mathrm{div}(f_A) = r((P + T_1) - (T_1)) \sim r((P) - (\mathcal{O}))$. Similarly, Miller's algorithm can produce a rational function f_B using another r-torsion point T_2 such that $\mathrm{div}(f_B) = r((Q+T_2)-(T_2)) \sim r((Q) - (\mathcal{O}))$. We then let $A = (P + T_1) - (T_1)$ and $B = (Q + T_2) - (T_2)$ and evaluate (5). In the case of the Weil pairing, T_1 and T_2 need only to be chosen so that A and B are disjoint.

Computing the Tate pairing is easier than computing the Weil pairing. The modified version of Miller's algorithm already returns the value $g(D) =< P, Q >_S$ where $\mathrm{div}(g) = r[(P) - (\mathcal{O})]$ and D is as in (4). To obtain our unique value, $t(P, Q)$, we only need to exponentiate in the field $\mathbb{F}_{q^d}$ by $(q^d - 1)/r$.

4. The Weil is Two Tates

As can be seen in the previous section, the methods for computing the two pairings are very similar. This leads one to wonder if there is a nice relation between them. In many texts (e.g. [4], [5]), words to the effect of "The Weil pairing is just two applications of the Tate pairing" are tossed around without care and rarely elaborated on. Through a lot of introspection and a little luck, we were able to find a simple expression relating the pairings and indeed show that the computation of the Weil pairing can be done through two computations of the Tate pairing. In fact, the rest of this section is directed toward showing that for any $P, Q \in E[r]$

$$e_r(P, Q) = \frac{< P, Q >_S}{< Q, P >_{-S}}. \tag{6}$$

4.1. The Proofs

We first state a general lemma whose proof requires a lot of ugly algebra. To provide cleanliness, we introduce some notation. For two points C, D, let $\ell_{C,D}$ denote the line through C and D, and let $M_{C,D}$ denote its slope (provided it is defined). For a point C, let ν_C denote the vertical line at C, i.e., $\nu_C = \ell_{C,\mathcal{O}}$.

Lemma 1 *Let E be an elliptic curve over a field K and suppose that the points P, Q and S on E satisfy one of the following criteria:*

$$\begin{array}{ll} \textit{(i)} & S \notin \{P-Q, P-S, -Q-S, P, -Q, -S\}, \\ \textit{(ii)} & S, P, Q \in E[2], \\ \textit{(iii)} & S = \mathcal{O}. \end{array}$$

Then

$$\frac{\nu_{P-S}}{\ell_{P,-S}}((Q+S)-(S)) = \frac{\nu_{Q+S}}{\ell_{Q,S}}((P-S)-(-S)). \tag{7}$$

We assume the equation for E is in Weierstrass form. Note: The criteria for P, Q, and S are solely to assure that both sides of (7) are defined (numerator and denominator are allowed to simultaneously vanish, in which case we let the quotient be 1). We need to show that

$$\frac{\ell_{P,-S}(S)}{\ell_{P,-S}(Q+S)} \cdot \frac{\nu_{P-S}(Q+S)}{\nu_{P-S}(S)} \;=\; \frac{\ell_{Q,S}(-S)}{\ell_{Q,S}(P-S)} \cdot \frac{\nu_{Q+S}(P-S)}{\nu_{Q+S}(-S)}\,. \tag{8}$$

Proof. If we regard lines as rational functions on E, then for points A, B, C of E such that the following expressions make sense, we have that

$$\begin{aligned} \ell_{A,B}(C) &= y_C - y_A - M_{A,B}(x_C - x_A) \\ &= y_C - y_B - M_{A,B}(x_C - x_B) \\ \nu_A(C) &= x_C - x_A, \end{aligned}$$

where x_A and y_A denote the coordinates of a point A. Furthermore, $x_{-A} = x_A$. Hence,

$$\ell_{P,-S}(S) = y_S - y_{-S} - \left(\frac{y_P - y_{-S}}{x_P - x_{-S}}\right)(x_S - x_{-S}) = y_S - y_{-S};$$

similarly, $\ell_{Q,S}(-S) = y_{-S} - y_S = -\ell_{P,-S}(S)$. From the definition, $\nu_{P-S}(S) = x_S - x_{P-S}$, $\nu_{Q+S}(-S) = x_{-S} - x_{Q+S}$, and $\nu_{Q+S}(P-S) = x_{P-S} - x_{Q+S} = -\nu_{P-S}(Q+S)$. Hence, it suffices to show that

$$\frac{-\ell_{Q,S}(-S)}{\ell_{P,-S}(Q+S)} \cdot \frac{\nu_{P-S}(Q+S)}{x_S - x_{P-S}} = \frac{\ell_{Q,S}(-S)}{\ell_{Q,S}(P-S)} \cdot \frac{-\nu_{P-S}(Q+S)}{x_{-S} - x_{Q+S}}\,. \tag{9}$$

The numerators of (9) are equal, so we need only show that

$$\ell_{P,-S}(Q+S) \cdot (x_S - x_{P-S}) = \ell_{Q,S}(P-S) \cdot (x_{-S} - x_{Q+S}). \tag{10}$$

The addition formulas ([9], p. 58) give

$$\begin{aligned} y_{-S} &= -y_S - a_1 x_S - a_3, \\ \text{and} & \\ y_{Q+S} &= -(y_S + a_1 x_{Q+S} + a_3 + M_{Q,S}(x_{Q+S} - x_S)). \end{aligned}$$

Substituting the above values for y_{-S} and y_{Q+S} into the expression for $\ell_{P,-S}(Q+S)$ and using the fact that $x_{-S} = x_S$, we see that

$$\begin{aligned} \ell_{P,-S}(Q+S) &= y_{Q+S} - y_{-S} - M_{P,-S}(x_{Q+S} - x_{-S}) \\ &= -[(y_S + a_1 x_{Q+S} + a_3 + M_{Q,S}(x_{Q+S} - x_S)] \\ &\quad -[-y_S - a_1 x_S - a_3] \\ &\quad -M_{P,-S}(x_{Q+S} - x_S) \\ &= -a_1(x_{Q+S} - x_S) - M_{Q,S}(x_{Q+S} - x_S) - M_{P,-S}(x_{Q+S} - x_S) \\ &= (x_S - x_{Q+S})(a_1 + M_{P,-S} + M_{Q,S}) \end{aligned}$$

Similarly

$$\ell_{Q,S}(P-S) = (x_S - x_{P-S})(a_1 + M_{P,-S} + M_{Q,S})$$

And now it is clear that both sides of (10) are equal to

$$(x_S - x_{P-S})(x_S - x_{Q+S})(a_1 + M_{P,-S} + M_{Q,S}).$$

■

With that established, we can now move on to the relevant result.

Proposition 1 *Let $P, Q \in E[r]$. Let $S \in E$ be chosen such that $Q+S, S \notin \mathcal{A}(P)$, $P-S, -S \notin \mathcal{A}(Q)$, and that S, P, and Q satisfy the conditions of Lemma 1. Then*

$$e_r(P,Q) = \frac{< P, Q >_S}{< Q, P >_{-S}}$$

Remark: In practical applications S is easily found. For example, if $P \in \mathcal{R}$ and $Q \notin \mathcal{R}$ then one possibility is $S = Q - P$. Fortunately, as the field size increases, it becomes more likely that a randomly chosen S will satisfy the required conditions.

Proof. In the Weil pairing computation, we make a special choice of A and B, namely

$$A = (P-S) - (-S) \qquad \text{and} \qquad B = (Q+S) - (S).$$

Then we have functions f_A, f_B such that

$$\text{div}(f_A) = r(P-S) - r(-S) \qquad \text{and} \qquad \text{div}(f_B) = r(Q+S) - r(S).$$

Let

$$A' = (P) - (\mathcal{O}) \qquad \text{and} \qquad B' = (Q) - (\mathcal{O}),$$

and let $g_{A'}, g_{B'}$ be functions such that

$$\operatorname{div}(g_{A'}) = r(P) - r(\mathcal{O}) \qquad \text{and} \qquad \operatorname{div}(g_{B'}) = r(Q) - r(\mathcal{O}).$$

We want to show that

$$\frac{f_A(B)}{f_B(A)} = \frac{g_{A'}(B)}{g_{B'}(A)}$$

or equivalently,

$$\frac{f_A(B)}{g_{A'}(B)} = \frac{f_B(A)}{g_{B'}(A)}.$$

Consider the function on left-hand-side of the above equation. We have

$$\begin{aligned}
\operatorname{div}\left(\frac{f_A}{g_{A'}}\right) &= r(A) - r(A') \\
&= r((P-S) - (-S)) - r((P) - (\mathcal{O})) \\
&= r((P-S) - (-S) - (P) + (\mathcal{O})) \\
&= r((P-S) + (S-P) - 2(\mathcal{O}) - (S-P) - (-S) - (P) + 3(\mathcal{O})) \\
&= r\left(\operatorname{div}\left(\frac{\nu_{P-S}}{\ell_{P,-S}}\right)\right).
\end{aligned}$$

This implies that there is a nonzero constant $k_A \in K$ such that

$$\frac{f_A}{g_{A'}} = k_A \left(\frac{\nu_{P-S}}{\ell_{P,-S}}\right)^r.$$

Similarly for B, there exists a constant k_B such that

$$\frac{f_B}{g_{B'}} = k_B \left(\frac{\nu_{Q+S}}{\ell_{Q,S}}\right)^r.$$

We are now reduced to showing that

$$k_A \left(\frac{\nu_{P-S}}{\ell_{P,-S}}\right)^r (B) = k_B \left(\frac{\nu_{Q+S}}{\ell_{Q,S}}\right)^r (A);$$

i.e.,

$$k_A \left(\frac{\nu_{P-S}}{\ell_{P,-S}}\right)^r ((Q+S) - (S)) = k_B \left(\frac{\nu_{Q+S}}{\ell_{Q,S}}\right)^r ((P-S) - (-S)).$$

However, the constants k_A, k_B will cancel out in their respective sides (since the divisors are of the form $(A) - (B)$), and thus the result follows from Lemma 1.

■

4.2. The Consequences

Now that we have established that the Weil pairing is, in fact, two applications of the Tate pairing, it is a good time to ask what it all means. First of all, this provides an upper bound on the amount of computation needed to obtain a Weil pairing. Given that we can efficiently perform Miller's algorithm, the Weil pairing is only two applications of the Tate pairing and a division away. Further, Galbraith suggests that this is in fact the most efficient known way to calculate the Weil pairing [5]. Another conclusion one may draw is that the Weil pairing only takes twice the computational time of the Tate pairing. However, this conclusion is not always true. As Galbraith points out, since typically $P \in E(\mathbb{F}_q)$ and $Q \in E(\mathbb{F}_{q^d})$, computing $< Q, P >$ involves significantly more complex field arithmetic than computing $< P, Q >$. This leads to the Weil pairing requiring more than twice the run time of one Tate pairing.

If we return to our IBE key exchange and require that the publicly computed C is an element of $E(\mathbb{F}_q)$, then all of the pairings in the key exchange have as their first argument a point in $E(\mathbb{F}_q)$ and as their second argument a point in $E(\mathbb{F}_{q^d})$. Thus, it should be significantly quicker to use the Tate pairing instead of the Weil pairing.

4.3. Experimental Timing Results

Using $d = 4$, we implemented the Tate pairing on the supersingular elliptic curve $y^2 + y = x^3 + x + 1$ over the fields $GF(2^{4m})$ for $m \in \{139, 163, 175, 199, 235\}$. The following table summarizes our timing results (in seconds) along with a comparison between the timings for a Weil and a Tate pairing. We used a Sun Sparc Ultra 5 with a speed of 270Mhz and 128Mb RAM. However, little effort was put into making the algorithms efficient. In particular, all arithmetic was carried out in $GF(2^{4m})$ when necessary, instead of using the extension method outlined in [5] (this can be coded up at a later date). Thus, it is more prudent to compare the relative computation times instead of the absolute times. The Δ column represents the percentage of time saved by using the Tate pairing in place of the Weil pairing. This relative percentage should not change significantly with quicker algorithms.

m	$t(P,Q)$	$t(Q,P)$	$e_r(P,Q)$	Δ
139	2.76	11.13	10.14	73%
163	3.83	5.03	2.75	-39%
175	4.56	17.12	15.03	70%
199	7.26	30.47	27.37	73%
235	10.19	29.72	22.93	56%

4.4. The Exceptional Case $m = 163$

One may notice from looking at the above table, that for $m = 163$, the Weil pairing is, in fact, faster to compute than the Tate pairing. This seems to be a consequence of the combination of the special form our large prime r with the inefficiency of our arithmetic operations (in particular multiplication) in the large field. For the elliptic curve given by

$y^2 + y = x^3 + x + 1$ over $\mathrm{GF}(2^{652})$, we have

$$r = 2^{163} + 2^{82} + 2^0.$$

Since r is so sparse, the execution of Miller's algorithm is much quicker. Meanwhile, the time to perform the exponentiation is not affected. Thus, the exponentiation, which is required for the Tate pairing but not for the Weil pairing, dominates the computation, allowing the Weil pairing to be computed more quickly.

4.5. Conclusions

The conclusions one can draw are not quite as simple as saying the Weil pairing always requires more than twice the computational time as the Tate pairing. Instead, one may need to take into account the density of the large prime r. However, in general, it can be said that for two points, P and Q, in an r-torsion subgroup of an elliptic curve, with $P \in E(\mathbb{F}_q)$ and $Q \in E(\mathbb{F}_{q^d})$, the Tate pairing, $t(P, Q)$, can be computed in less than half the time it takes to compute the Weil pairing, $e_r(P, Q)$, provided r has reasonable density. At the same time, though, it should be noted that $t(Q, P)$ will often take longer to compute than $e_r(Q, P)$ since, in this case, the small-field property of P aids the Weil pairing, but not the Tate Pairing. For these reasons, anyone who plans on implementing the Tate pairing in place of the Weil pairing needs to consider the rationality of the points and the bit density of their torsion.

References

[1] R. Balasubramanian and N. Koblitz, "The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm", *J. Crypto.*, **11** (1998) 141-145.

[2] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing", *Proc. CRYPTO 2001*, Springer LNCS 2139 (2001) 213-229.

[3] G. Frey, M. Müller, H.-G. Rück, "The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems", *IEEE Trans. Inform. Theory*, **45**, No.5 (1999) 1717-1719.

[4] G. Frey and H.-G. Rück, "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves", *Mathematics of Computation*, **62**, No.206 (1994) 865-874.

[5] S. Galbraith, K. Harrison, and D. Soldera, "Implementing the Tate pairing", *Algorithmic Number Theory 2002*, Springer LNCS 2369 (2002) 324-337.

[6] A. J. Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, Boston, 1993.

[7] A. J. Menezes, T. Okamoto, and S. A. Vanstone, "Reducing elliptic curve logarithms to logarithms in the finite field", *IEEE Trans. Inform. Theory*, **39**, No.5 (1993) 1639-1646.

[8] V. Miller, *Short programs for functions on curves*, unpublished manuscript, 1986.

[9] J. Silverman, *The Arithmetic of Elliptic Curves*, Springer GTM 106, 1986.