Pairing-based Cryptography

R. Barua and R. Dutta

Chapter 1

Preliminary Background and Definitions

1.1 Introduction

Elliptic curve cryptography is now widely being used in designing many cryptographic protocols. One of the main properties that is used in overwhelming number of protocols is the (modified) Weil or Tate pairing. Pairing-based protocols are used in a variety of protocols and pairing has found applications in the solution of ID-based cryptograhic schemes and short signature schemes. Although elliptic curves have other uses in cryptography (like the ElGamal encryption based on the hardness of discrete log problem in elliptic curve groups) we would mainly concentrate on pairing-based cryptography. (For more on elliptic curve cryptography see *e.g.* [35]).

We first include some background material on elliptic curves. We then concentrate on specifying several versions of Diffie-Hellman problems. Security of various protocols are based on the hardness of these problems. We also deal with the security notion and security models for different cryptographic primitives.

1.2 Elementary Concepts on Elliptic Curves

Let K be a field and $\overline{K}$ its algebraic closure. An elliptic curve over K is defined by a Weierstrass equation

$$E/K : y^2 + a_1xy + a_3y \quad = \quad x^3 + a_2x^2 + a_4x + a_6$$

where $a_1, a_2, a_3, a_4, a_6 \in K$ and there are no "singular points". If $L \supset K$, then the set of L-rational points on E is

$$E(L) = \{(x, y) \in L \times L : y^2 + a_1xy + a_3y \quad = \quad x^3 + a_2x^2 + a_4x + a_6\} \cup \{\mathcal{O}\},$$

where $\mathcal{O}$ is a special point , called the *point at infinity*. If $L \supset K$, then $E(L) \supset E(K)$. We denote $E(\overline{K})$ by E. Simplified Weierstrass equation is as follows.

Case 1: If $\mathrm{char}(K) \neq 2, 3$, then the equation simplifies to $y^2 = x^3 + ax + b$, $a, b \in K$ and $4a^3 + 27b^2 \neq 0$.

Case 2: If $\mathrm{char}(K) = 2$, then the equation simplifies to

$y^2 + xy = x^3 + ax^2 + b$, $a, b \in K$, $b \neq 0$, non-supersingular, or
$y^2 + cy = x^3 + ax + b$, $a, b, c \in K$, $c \neq 0$, supersingular.

For any $L \supset K$, the set $E(L)$ is an abelian group under the "chord-and-tangent law". Consider $E/K : y^2 = x^3 + ax + b$. Addition formulae are as follows:

1. $P + \mathcal{O} = \mathcal{O} + P = P$, for all $P \in E(L)$.
2. $-\mathcal{O} = \mathcal{O}$.
3. If $P = (x, y) \in E(L)$, then $-P = (x, -y)$.
4. If $Q = -P$, then $P + Q = \mathcal{O}$.
5. If $P = (x_1, y_1) \in E(L)$, $Q = (x_2, y_2) \in E(L)$, $P \neq -Q$, then $P + Q = (x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$, and
$$\begin{aligned} \lambda &= \frac{y_2 - y_1}{x_2 - x_1} \quad \text{if } P \neq Q; \\ \lambda &= \frac{3x_1^2 + a}{2y_1} \quad \text{if } P = Q. \end{aligned}$$

For the purpose of cryptography, assume henceforth that $K = \mathbb{F}_q$, i.e. the finite field of characteristic p and of order $q = p^m, m$ odd and $\overline{K} = \cup_{m \geq 1} \mathbb{F}_{q^m}$. The following are three important results on the group order of elliptic curve groups.

Theorem 1.2.1 *(Hasse's Theorem)* $\#E(\mathbb{F}_q) = q + 1 - t$, $|t| \leq 2\sqrt{q}$. *Consequently,* $\#E(\mathbb{F}_q) \approx q$.

Theorem 1.2.2 *(Schoof's Algorithm)* $\#E(\mathbb{F}_q)$ *can be computed in polynomial time.*

Theorem 1.2.3 *(Weil Theorem) Let* $t = q+1-\#E(\mathbb{F}_q)$. *Let* α, β *be complex roots of* $T^2 - tT + q \in Z[T]$. *Then* $\#E(\mathbb{F}_{q^k}) = q^k + 1 - \alpha^k - \beta^k$ *for all* $k \geq 1$.

The structure of elliptic curve groups is summarized by the following results.

– Let E be an elliptic curve defined over $\mathbb{F}_q$. Then $E(\mathbb{F}_q) \cong Z_{n_1} \oplus Z_{n_2}$, where $n_2 | n_1$ and $n_2 | (q-1)$.

– $E(\mathbb{F}_q)$ is cyclic if and only if $n_2 = 1$.

– $P \in E$ is an n-torsion point if $nP = \mathcal{O}$ and $E[n]$ is the set of all n-torsion points.

– If $\gcd(n, q) = 1$, then $E[n] \cong Z_n \oplus Z_n$.

1.2.1 Supersingular Elliptic Curves

An elliptic curve $E/\mathbb{F}_q$ is supersingular if $p|t$ where $t = q + 1 - \#E(\mathbb{F}_q)$.

Theorem 1.2.4 *(Waterhouse) $E/\mathbb{F}_q$ is supersingular if and only if $t^2 = 0, q, 2q, 3q$ or $4q$. The group structure is given by the following result.*

Theorem 1.2.5 *(Schoof) Let $E/\mathbb{F}_q$ be supersingular with $t = q + 1 - \#E(\mathbb{F}_q)$. Then*

1. *If $t^2 = q, 2q$ or $3q$, then $E(\mathbb{F}_q)$ is cyclic.*
2. *If $t^2 = 4q$ and $t = 2\sqrt{q}$, then $E(\mathbb{F}_q) \cong Z_{\sqrt{q}-1} \oplus Z_{\sqrt{q}-1}$.*
3. *If $t^2 = 4q$ and $t = -2\sqrt{q}$, then $E(\mathbb{F}_q) \cong Z_{\sqrt{q}+1} \oplus Z_{\sqrt{q}+1}$.*
4. *If $t = 0$ and $q \not\equiv 3 \bmod 4$, then $E(\mathbb{F}_q)$ is cyclic.*
5. *If $t = 0$ and $q \equiv 3 \bmod 4$, then $E(\mathbb{F}_q)$ is cyclic or $E(\mathbb{F}_q) \cong Z_{\frac{q+1}{2}} \oplus Z_2$.*

1.2.2 Divisors and Weil Pairing

Let E be an elliptic curve defined over a field $\mathbb{F}_q$ and given by the equation $C(x, y) = 0$, where $C(x, y)$ is the polynomial defining E. We consider the set of $\mathbb{F}_{q^n}$-rational points of E. The group of divisors of $E(\mathbb{F}_{q^n})$ is the free abelian group generated by the points of $E(\mathbb{F}_{q^n})$. Thus any divisor D is of the form

$$D = \sum_{P \in E(\mathbb{F}_{q^n})} n_P \langle P \rangle.$$

where $n_P \in Z$. The support of a divisor D is the set of points $\{P \in E | n_P \neq 0\}$. We will only consider *zero divisors*, i.e., divisors where $\sum n_P = 0$.

A rational function f on E is an element of the field of fractions of the ring $\mathbb{F}_{q^n}[x, y]/(C(x, y))$. If $P = (x, y)$, then by $f(P)$ we mean $f(x, y)$. The divisor of a rational function f is defined by

$$\mathrm{div}(f) = \sum_{P \in E(\mathbb{F}_{q^n})} \mathrm{ord}_P(f) \langle P \rangle$$

where $\mathrm{ord}_P(f)$ is the order of the zero/pole that f has at P. A divisor D is said to be *principal* if $D = \mathrm{div}(f)$, for a rational function f.

Theorem 1.2.6 *A divisor $D = \sum_{P \in E(\mathbb{F}_{q^n})} n_P \langle P \rangle$ is principal if and only if*

1. *$\sum n_P = 0$ and*

2. $\sum n_P P = \mathcal{O}$.

Two divisors D_1 and D_2 are said to be *equivalent* ($D_1 \sim D_2$) if $D_1 - D_2$ is principal.

Theorem 1.2.7 *Any zero divisor $D = \sum n_P \langle P \rangle$ is equivalent to a (unique) divisor of the form $\langle Q \rangle - \langle \mathcal{O} \rangle$ for some $Q \in E(\mathbb{F}_{q^n})$.*

Given a rational function f and a zero divisor $D = \sum n_P \langle P \rangle$, define

$$f(D) \quad = \quad \prod_{P \in E(\mathbb{F}_{q^n})} f(P)^{n_P}.$$

Now we define the important notion of *Weil pairing.*

Definition 1.2.8 *Let $P, Q \in E[n]$, the subset of all n-torsion points of the $\mathbb{F}_q$-rational points of an elliptic curve E defined over $\mathbb{F}_q$. Let D_P be a divisor which is equivalent to $\langle P \rangle - \langle \mathcal{O} \rangle$. Then nD_P is principal and hence there exists a rational function f_P such that $div(f_P) = nD_P$. Define D_Q and f_Q analogously. Let D_P and D_Q have disjoint support. Then* Weil pairing *$\widehat{e}(P, Q)$ is defined as*

$$\widehat{e}(P, Q) \quad = \quad \frac{f_P(D_Q)}{f_Q(D_P)}.$$

(One can define another pairing function, called *Tate pairing* [5, 31, 29]*(cf chapter 3)*,with properties very similar to the Weil pairing).

1.2.3 Weil Pairing and Bilinear Map

Let E be an elliptic curve defined over $\mathbb{F}_q$. Let n be an integer with $\gcd(n, q) = 1$ and $\mathbb{F}_{q^k}$ be the smallest extension of $\mathbb{F}_q$ such that $E[n] \subseteq E(\mathbb{F}_{q^k})$. (This implies that $n^2 | \#E(\mathbb{F}_{q^k})$ and $n | (q^k - 1)$.) Let μ_n be the subgroup of order n in $\mathbb{F}_{q^k}^*$. Then Weil pairing $\widehat{e}$ is a map $\widehat{e} : E[n] \times E[n] \to \mu_n$ with the following properties.

1. *(Bilinearity)* : For all $R, S, T \in E[n]$,
   (a) $\widehat{e}(S + R, T) = \widehat{e}(S, T) \cdot \widehat{e}(R, T)$,
   (b) $\widehat{e}(S, T + R) = \widehat{e}(S, T) \cdot \widehat{e}(S, R)$.
2. *(Non-degeneracy)* : Let $S \in E[n]$. If $\widehat{e}(S, T) = 1$ for all $T \in E[n]$, then $S = \mathcal{O}$.
3. *(Computability)* : $\widehat{e}$ can be computed efficiently.
4. *(Identity)* : $\widehat{e}(S, S) = 1$ for all $S \in E[n]$.
5. *(Alternation)* : $\widehat{e}(S, T) = \widehat{e}(T, S)^{-1}$.

The property of bilinearity defined above implies the following: $\widehat{e}(aS, bT) = \widehat{e}(S,T)^{ab}$ for all $S, T \in G_1$ and $a, b \in Z$.

Lemma 1.2.9 *Let $P \in E[n]$ have order n. Then $P_1, P_2 \in E[n]$ are in the same coset of $\langle P \rangle$ within $E[n]$ if and only if $\widehat{e}(P, P_1) = \widehat{e}(P, P_2)$.*

Theorem 1.2.10 *Let $P \in E[n]$ have order n. Let $R \in E[n]$ be such that $\widehat{e}(P, R)$ has order n. Then $f : \langle P \rangle \to \mu_n$ defined by $f(Q) = \widehat{e}(Q, R)$ is a group isomorphism.*

As a consequence of the above, we have the following important fact.

$$E[n]/\langle P \rangle \cong Z_n \cong \mu_n.$$

We next define bilinear pairing or bilinear map.

Definition 1.2.11 *Let G_1, G_2 be two groups of the some large prime order q. We view G_1 as an additive group and G_2 as a multiplicative group. Let P be an arbitrary generator of G_1. (aP* denotes P added to itself a times*). Assume that discrete logarithm problem (DLP) is hard in both G_1 and G_2. A mapping $e : G_1^2 \to G_2$ satisfying the following properties is called a* cryptographic bilinear map.

(Bilinearity) : $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_q^*$.
(Non-degeneracy): $e(P, P) \neq 1$. i.e. if P is a generator of G_1, then $e(P, P)$ is a generator of G_2.
(Computability) : There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

Modified Weil Pairing [17] and Tate Pairing [5], [31] are examples of cryptographic bilinear maps. Currently, active research is being carried out to obtain efficient algorithms to compute pairings.

Following are some important properties of bilinear pairings.

1. $e(S, \mathcal{O}) = 1$ and $e(\mathcal{O}, S) = 1$ for all $S \in G_1$.
2. $e(S, -T) = e(-S, T) = e(S, T)^{-1}$ for all $S, T \in G_1$.
3. $e(S, T) = e(T, S)$ for all $S, T \in G_1$.
4. Let $S \in G_1$. If $e(S, T) = 1$ for all $T \in G_1$, then $S = \mathcal{O}$.

We now illustrate by an example how bilinear map can be derived from Weil pairing. We fix a supersingular curve E over $\mathbb{F}_p$, $p > 3$ and $p \equiv 2 \bmod 3$, given by $y^2 = x^3 + 1$. $E(\mathbb{F}_p)$ contains $p + 1$ points. Let $P \in E(\mathbb{F}_p)$ be a point of order n where $n|(p + 1)$. The set $E(\mathbb{F}_{p^2})$ contains a

point Q of order n which is linearly independent of the points in $E(\mathbb{F}_p)$. Hence $E(\mathbb{F}_{p^2})$ contains a subgroup isomorphic to Z_{n^2}. Denote this by $E[n]$. Let G_1 the subgroup of points generated by P. Let $\zeta \in \mathbb{F}_{p^2}$ be a non-trivial root of $x^3 + 1 \equiv 0 \bmod p$. Then $\phi(x, y) = (\zeta x, y)$ is an automorphism on E. The map ϕ is called a distortion map.
Note : $E[n]$ is the group generated by P and $\phi(P)$.

Let G_2 be the subgroup of $\mathbb{F}_{p^2}^*$ of order n and $\widehat{e} : E[n] \times E[n] \to G_2$ be the Weil pairing map. The *modified Weil pairing* $e : G_1 \times G_1 \to G_2$ is defined by

$$e(P, Q) \quad = \quad \widehat{e}(P, \phi(Q)).$$

It is not hard to check that e is a cryptographic bilinear pairing.

Henceforth, we take G_1, G_2, e as defined in Definition 1.2.11 for the rest of the article unless mentioned otherwise.

For a set S, we use the notation $a \in_R S$ or $a \longleftarrow S$ to mean that a is randomly chosen from S and the notation $|$ to denote concatenation of data items (*cf.* $A|B|C$). Unless otherwise stated, we assume that the messages are arbitrary length finite binary strings and the above setup holds for the cryptographic protocols throughout the presentation. We define a function $f(m)$ to be *negligible* if it is less than $\frac{1}{m^l}$ for every fixed $l > 0$ and sufficiently large integer m.

1.3 Diffie-Hellman Problems

1. Computational Diffie-Hellman (CDH) problem in G_1 :
   Instance : (P, aP, bP) for some $a, b \in Z_q^*$.
   Output : abP.

   The success probability of any probabilistic, polynomial-time algorithm $\mathcal{A}$ in solving CDH problem in G_1 is defined to be :

   $$\mathsf{Succ}^{\mathsf{CDH}}_{\mathcal{A},G_1} = \mathsf{Prob}[\mathcal{A}(P, aP, bP) = abP : a, b \in_R Z_q^*].$$

   CDH assumption : For every probabilistic, polynomial-time algorithm $\mathcal{A}$, $\mathsf{Succ}^{\mathsf{CDH}}_{\mathcal{A},G_1}$ is negligible.

2. Decisional Diffie-Hellman (DDH) problem in G_1 :
   Instance : (P, aP, bP, cP) for some $a, b, c \in Z_q^*$.
   Output : yes if $c = ab \bmod q$ and output no otherwise.
   Comments : DDH problem in G_1 is easy. DDH problem in G_1 can be solved in polynomial time by verifying $e(aP, bP) = e(P, cP)$. This is the well known MOV reduction [17] : The DLP in G_1 is no harder than the DLP in G_2.

The advantage of any probabilistic, polynomial-time, 0/1-valued algorithm $\mathcal{A}$ in solving DDH problem in some group $G =< P >$ of order q is defined to be :

$$\mathsf{Adv}^{\mathsf{DDH}}_{\mathcal{A},G} = |\mathsf{Prob}[\mathcal{A}(P, aP, bP, cP) = 1] - \mathsf{Prob}[\mathcal{A}(P, aP, bP, abP) = 1] : a, b, c \in_R Z_q^*|.$$

DDH assumption in G: For every probabilistic, polynomial-time, 0/1-valued algorithm $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{DDH}}_{\mathcal{A},G}$ is negligible.

Gap Diffie-Hellman (GDH) group : A prime order group G_1 is a GDH group if there exists an efficient polynomial-time algorithm which solves the DDH problem in G_1 and there is no probabilistic polynomial-time algorithm which solves the CDH problem with non-negligible probability of success. The domains of bilinear pairings provide examples of GDH groups. The MOV reduction [42] essentially introduces a method to solve DDH in G_1, whereas there is no known efficient algorithm for solving CDH in G_1, in general.

3. Weak Diffie-Hellman (W-DH) problem in a group G_1 :
   Instance : (P, Q, sP) for $P, Q \in G_1$ and for some $s \in Z_q^*$.
   Output : sQ.
   Comments : W-DH problem is equivalent to CDH problem.

4. Reversion of CDH (RCDH) problem in G_1 :
   Instance : (P, aP, rP) for some $a, r \in Z_q^*$.
   Output : $bP, b \in Z_q^*$ satisfying $a = rb \bmod q$.
   Comments : RCDH problem is equivalent to CDH problem in G_1 [25].

5. $(k+1)$-exponent problem ($(k+1)$-EP) in G_1:
   Instance : $(P, yP, y^2P, \ldots, y^kP)$for a random $y \in Z_q^*$.
   Output : $y^{k+1}P$.
   Comments : $(k+1)$-EP is no harder than CDH problem.

6. k-Diffie-Hellman Inversion (k-DHI) problem in G_1 :
   Instance : $(P, yP, y^2P, \ldots, y^kP)$ for a random $y \in Z_q^*$.
   Output : $\frac{1}{y}P$.
   Comments : k-DHI problem is polynomially equivalent to $(k+1)$-EP.

7. k-Strong Diffie-Hellman (k-SDH) problem in G_1 :
   Instance : $(P, yP, y^2P, \ldots, y^kP)$ for a random $y \in Z_q^*$.
   Output : $(c, \frac{1}{y+c}P)$ where $c \in Z_q^*$.
   Comments : k-SDH problem is a stronger version of k-DHI problem. When c is pre-specified, k-SDH problem is polynomially equivalent to k-DHI. k-SDH problem has a simple random self reduction in G_1.

1.4 Bilinear Diffie-Hellman Problems

1. Bilinear Diffie-Hellman (BDH) problem in (G_1, G_2, e) :
   Instance : (P, aP, bP, cP) for some $a, b, c \in Z_q^*$.
   Output : $e(P, P)^{abc}$.

2. Decisional Bilinear Diffie-Hellman (DBDH) problem in (G_1, G_2, e) :
   Instance : (P, aP, bP, cP, r) for some $a, b, c \in_R Z_q^*$, $r \in_R G_2$.
   Output : yes if $r = e(P, P)^{abc}$ and output no otherwise.

3. Decisional Hash Bilinear Diffie-Hellman (DHBDH) problem in (G_1, G_2, e) :
   Instance : (P, aP, bP, cP, r) for some $a, b, c, r \in Z_q^*$ and a one way hash function $H : G_2 \rightarrow Z_q^*$.
   Output : yes if $r = H(e(P, P)^{abc}) \bmod q$ and output no otherwise.
   Comments : The DHBDH problem in (G_1, G_2, e) is a hash version of the decisional BDH problem in (G_1, G_2, e) .

4. k-Bilinear Diffie-Hellman Inversion (k-BDHI) problem in (G_1, G_2, e) :
   Instance : $(P, yP, y^2P, \ldots, y^kP)$ for some $y \in Z_q^*$.
   Output : $e(P, P)^{\frac{1}{y}} \in G_2$.
   Comments : 1-BDHI assumption is polynomially equivalent to the standard BDH assumption. It is not known if the k-BDHI assumption, for $k > 1$, is polynomially equivalent to BDH.

5. k-Decisional Bilinear Diffie-Hellman Inversion (k-DBDHI) problem in (G_1, G_2, e) :
   Instance : $(P, yP, y^2P, \ldots, y^kP, r)$ for some $y \in Z_q^*$, $r \in_R G_2$.
   Output : yes if $r = e(P, P)^{\frac{1}{y}} \in G_2$ and output no otherwise.

1.5 Security Models

In this section, we briefly review the security models for the three fundamental cryptographic primitives: Encryption, Digital Signature and Key Agreement. There are several variants of public key encryption: ID-based encryption (IBE), Searchable Public Key Encryption (SPKE), Hierarchical ID-based Encryption (HIDE); and depending on the nature and requirement of practical applications, there are a wide variety of signature schemes: Blind signature, Multi-signature, Aggregate signature, Verifiably encrypted signature, Ring signature, Group signature, Unique signature *etc.* Apart from these three basic primitives, there are protocol designs for Signcryption, Threshold decryption, Key sharing, Identification schemes, Chameleon hashes *etc.* Describing the security notions and the security models of each of them is beyond the scope.

1.5.1 Security Model for ID-Based Encryption Schemes

The standard notion of security for public key encryption scheme is the indistinguishability of encryptions against adaptive chosen ciphertext attack (IND-CCA) [8, 45].

Boneh and Franklin [17] strengthened the IND-CCA model to IND-ID-CCA model which is the standard notion of security for ID-based encryption schemes. In an ID-based encryption scheme there are four algorithms.

1. Setup : Creates system parameters and *master key*.
2. Extract : Uses master key to generate the private key corresponding to an arbitrary public key string ID.
3. Encrypt : Encrypts messages using the public key ID.
4. Decrypt : Decrypts the message using the corresponding private key of ID.

The IND-ID-CCA model deals with an adversary who possesses private keys corresponding to identities of its choice $\mathsf{ID}_1, \ldots, \mathsf{ID}_n$ and attacks an identity ID in an ID-based system. Consider the following game between the challenger and an adversary $\mathcal{A}$.

Setup: The challenger takes a security parameter k and runs the Setup algorithm. It gives the adversary the resulting system parameters params and keeps the master key secret to itself.

Phase 1: The adversary issues queries $q_1, \ldots, q_m$ where q_i is one of the following two queries.

- Extraction query $\langle \mathsf{ID}_i \rangle$. The challenger responds by running algorithm $\mathsf{Extract}$ to generate the private key d_i corresponding to the public key $\langle \mathsf{ID}_\mathsf{i} \rangle$. It then sends d_i to the adversary.
- Decryption query $\langle \mathsf{ID}_i, C_i \rangle$. The challenger responds by running algorithm $\mathsf{Extract}$ to generate the private key d_i corresponding to ID_i. It then runs algorithm $\mathsf{Decrypt}$ to decrypt the ciphertext C_i using the private key d_i. It sends the resulting plaintext to the adversary.

These queries may be asked adaptively, that is, each query q_i may depend on the replies to $q_1, \ldots, q_{i-1}$.

Challenge: Once the adversary decides that Phase 1 is over, it outputs two equal length plaintext M_0, M_1, an identity ID on which it wishes to be challenged. The only constraint is that ID did not appear in any private key extraction queries in Phase 1. The challenger picks a random bit $b \in \{0, 1\}$ and sets $C = \mathsf{Encrypt}(\mathsf{params}, \mathsf{ID}, M_b)$. It sends C as the challenge to the adversary.

Phase 2: The adversary issues more queries $q_{m+1}, \ldots, q_n$ where q_i is one of the following two queries.

– Extraction query $\langle \mathsf{ID}_i \rangle$ where $\mathsf{ID}_i \neq \mathsf{ID}$. Challenger responds as in Phase 1.

– Decryption query $\langle \mathsf{ID}_i, C_i \rangle \neq \langle \mathsf{ID}, C \rangle$. Challenger responds as in Phase 1.

Guess: Finally, the adversary outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$.

We refer to such an adversary $\mathcal{A}$ as an IND-ID-CCA adversary. The advantage for the adversary $\mathcal{A}$ in attacking the scheme is defined as

$$\mathsf{Adv}(\mathcal{A}) = |\mathsf{Prob}[b = b'] - \frac{1}{2}|.$$

We say that an identity based scheme is semantically secure against an adaptive chosen ciphertext attack (IND-ID-CCA) if no polynomially bounded adversary $\mathcal{A}$ has non-negligible advantage against the challenger.

The IND-ID-CCA model is the strongest acceptable notion of security and the security model for other public key encryption schemes (*e.g.* SPKE, HIDE *etc.*) are based on it. A less stronger security notion of a public key encryption is chosen plaintext attack (IND-CPA) where the queries are with chosen plaintexts nad the adversary is not allowed to perform Phase 2.

Boneh and Boyen [14] gave Selective ID model, which is slightly weaker than the model described above. In this model the adversary must commit ahead of the time to the identity that it intends to attack, whereas in the standard model described above, the adversary is allowed to choose this identity adaptively.

The security notion for (ID-based) threshold decryption is a modified extension of the security model described above in the threshold setting. More details can be found in [3].

1.5.2 Security Model for Digital Signature Schemes

Security against existential forgery under adaptive chosen message attack is the strongest notion of security for digital signature schemes. This was defined by Goldwasser, Micali and Rivest [33].

A standard digital signature scheme $\mathsf{DSig} = (\mathcal{G}, \mathcal{K}, \mathcal{S}, \mathcal{V})$ consists of four algorithms.

1. $\mathcal{G}$: generates randomly system parameters params.
2. $\mathcal{K}$: generates randomly public/secret key pair PK, SK of a signer.
3. $\mathcal{S}$: signature generation algorithm that generates a signature on a given message m using the secret key SK of a signer.
4. $\mathcal{V}$: signature verification algorithm that checks the validity of a signature on a given message using the public key of a signer.

We say that the signature scheme DSig is existentially unforgeable under an adaptive chosen message attack if it is infeasible for a forger who only knows the public key to produce a valid

message-signature pair after obtaining polynomially many signatures on messages of its choice from the signer. The advantage in existentially forging a signature of a forger algorithm $\mathcal{F}$, given access of a signing oracle $\mathcal{S}$, is defined to be

$$\mathsf{ADV}_{\mathsf{DSig}}(\mathcal{F}) := \mathsf{Prob}[\mathcal{V}(\mathsf{PK}, m, \sigma) = 1 : (\mathsf{PK}, \mathsf{SK}) \xleftarrow{R} \mathcal{K}, (m, \sigma) \xleftarrow{R} \mathcal{F}^{\mathcal{S}}(\mathsf{PK})].$$

The probability is taken over the coin tosses of the key generation algorithm and of the forger. Here the forger $\mathcal{F}$ is allowed to query the signing oracle adaptively: any of its query may depend on previous answers, but it may not emit a signature for a message on which it had previously queried the oracle. The forger may additionally have access to a hash oracle, which can be used as a random oracle.

Formally a signature scheme is secure against existential forgery on adaptive chosen-message attacks if for every probabilistic polynomial-time forger algorithm $\mathcal{F}$, there does not exist a non-negligible probability ϵ such that $\mathsf{ADV}_{\mathsf{DSig}}(\mathcal{F}) \geq \epsilon$.

We now describe the security notion of a Multi-signature scheme based on the basic digital signature scheme DSig.

Security of Multi-signature

A multi-signature scheme consists of three algorithms $\mathsf{MSig} = (\mathcal{MK}, \mathcal{MS}, \mathcal{MV})$, where $\mathcal{MK}$ is the key generation algorithm, $\mathcal{MS}$ is the multi-signature generation algorithm and $\mathcal{MV}$ is the multi-signature verification algorithm. We say that a multi-signature scheme is secure against *existential forgery under chosen message attack* if the following task is computationally infeasible for the adversary:

The adversary is given a public key PK_1; then outputs $(n-1)$ pairs of public and secret keys $\mathsf{PK}_2, \ldots, \mathsf{PK}_n$ and $\mathsf{SK}_2, \ldots, \mathsf{SK}_n$ respectively; is allowed to run the multi-signature generation algorithm $\mathcal{MS}$ with user U_1 having public key PK_1 on messages of the adversary's choosing; finally has to produce a message m, a subset L of users with $U_1 \in L$ and a signature σ such that $\mathcal{MV}$ returns 1 on input (L, m, σ) and U_1 did not participate in multi-signature generation algorithm for message m.

The multi-signature scheme MSig is said to be secure if there is no probabilistic, polynomial-time forger with non-negligible advantage.

1.5.3 Model for Key Agreement Schemes

Security Model for Key Agreement

A sound formalization of security model for the authenticated key agreement is introduced by Bresson *et al.* [20, 21, 22]. These works are based on the initial work of Bellare and Rogaway [9]

and Bellare, Canetti and Krawczyk [7]. We describe below the adversarial model following Bresson *et al.*'s [20] formal security model which is more general in the sense that it covers authenticated key agreement in group setting and suited for dynamic groups.

Let $\mathcal{P} = \{U_1, \ldots, U_n\}$ be a set of n (fixed) users or participants. At any point of time, any subset of $\mathcal{P}$ may decide to establish a session key. Thus a user can execute the protocol for group key agreement several times with different partners, can join or leave the group at his desire by executing the protocols for Join or Leave. We identify the execution of protocols for key agreement, member(s) join and member(s) leave as different sessions. The adversarial model consists of allowing each user an unlimited number of instances with which it executes the protocol for key agreement or inclusion or exclusion of a user or a set of users. We assume adversary never participates as a user in the protocol. This adversarial model allows concurrent execution of the protocol. The interaction between the adversary $\mathcal{A}$ and the protocol participants occur only via oracle queries, which model the adversary's capabilities in a real attack. Let S, S_1, S_2 be three sets defined as:

$$S = \{(V_1, i_1), \ldots, (V_l, i_l)\}, S_1 = \{(V_{l+1}, i_{l+1}), \ldots, (V_{l+k}, i_{l+k})\}, S_2 = \{(V_{j_1}, i_{j_1}), \ldots, (V_{j_k}, i_{j_k})\}$$

where $\{V_1, \ldots, V_l\}$ is any non-empty subset of $\mathcal{P}$. We will require the following notations.

Π_U^i	:	i-th instance of user U.
sk_U^i	:	session key after execution of the protocol by Π_U^i.
sid_U^i	:	session identity for instance Π_U^i. We set $\mathsf{sid}_U^i = S = \{(U_1, i_1), \ldots, (U_k, i_k)\}$ such that $(U, i) \in S$ and $\Pi_{U_1}^{i_1}, \ldots, \Pi_{U_k}^{i_k}$ wish to agree upon a common key.
pid_U^i	:	partner identity for instance Π_U^i, defined by $\mathsf{pid}_U^i = \{U_1, \ldots, U_k\}$, such that $(U_j, i_j) \in \mathsf{sid}_U^i$ for all $1 \leq j \leq k$.
acc_U^i	:	0/1-valued variable which is set to be 1 by Π_U^i upon normal termination of the session and 0 otherwise.

We assume that the adversary has complete control over all communications in the network. All information that the adversary gets to see is written in a transcript. So a transcript consists of all the public information flowing across the network. The following oracles model an adversary's interaction with the users in the network:

- $\mathsf{Send}(U, i, m)$: This query models an active attack, in which the adversary may intercept a message and then either modify it, create a new one or simply forward it to the intended participant. The output of the query is the reply (if any) generated by the instance Π_U^i upon receipt of message m. The adversary is allowed to prompt the unused instance Π_U^i to initiate the protocol with partners $U_2, \ldots, U_l, l \leq n$, by invoking $\mathsf{Send}(U, i, \langle U_2, \ldots, U_l \rangle)$.
- $\mathsf{Execute}(S)$: This query models passive attacks in which the attacker eavesdrops on honest execution of group key agreement protocol among unused instances $\Pi_{V_1}^{i_1}, \ldots, \Pi_{V_l}^{i_l}$ and outputs the transcript of the execution. A transcript consists of the messages that were exchanged during the honest execution of the protocol.

– $\mathsf{Join}(S, S_1)$: This query models the insertion of user instances $\Pi_{V_{l+1}}^{i_{l+1}}, \ldots, \Pi_{V_{l+k}}^{i_{l+k}}$ in the group $\{\Pi_{V_1}^{i_1}, \ldots, \Pi_{V_l}^{i_l}\}$ for which $\mathsf{Execute}$ have already been queried. The output of this query is the transcript generated by the invocation of algorithm Join. If $\mathsf{Execute}(S)$ has not taken place, then the adversary is given no output.

– $\mathsf{Leave}(S, S_2)$: This query models the removal of user instances $\Pi_{V_{j_1}}^{i_{j_1}}, \ldots, \Pi_{V_{j_k}}^{i_{j_k}}$ from the group $\{\Pi_{V_1}^{i_1}, \ldots \Pi_{V_l}^{i_l}\}$. If $\mathsf{Execute}(S)$ has not taken place, then the adversary is given no output. Otherwise, algorithm Leave is invoked. The adversary is given the transcript generated by the honest execution of procedure Leave.

– $\mathsf{Reveal}(U, i)$: This unconditionally outputs session key sk_U^i if it has previously been accepted by Π_U^i, otherwise a value NULL is returned. This query models the misuse of the session keys, *i.e* known session key attack.

– $\mathsf{Corrupt}(U)$: This outputs the long-term secret key (if any) of player U. The adversarial model that we adopt is a weak-corruption model in the sense that only the long-term secret keys are compromised, but the ephemeral keys or the internal data of the protocol participants are not corrupted. This query models (perfect) forward secrecy.

– $\mathsf{Test}(U, i)$: This query is allowed only once, at any time during the adversary's execution. A bit $b \in \{0, 1\}$ is chosen uniformly at random. The adversary is given sk_U^i if $b = 1$, and a random session key if $b = 0$. This oracle computes the adversary's ability to distinguish a real session key from a random one.

An adversary which has access to the $\mathsf{Execute}$, Join, Leave, Reveal, $\mathsf{Corrupt}$ and Test oracles, is considered to be passive while an active adversary is given access to the Send oracle in addition. (For static case, there are no Join or Leave queries as a group of fixed size is considered.)

The adversary can ask Send, $\mathsf{Execute}$, Join, Leave, Reveal and $\mathsf{Corrupt}$ queries several times, but Test query is asked only once and on a fresh instance. We say that an instance Π_U^i is *fresh* unless either the adversary, at some point, queried $\mathsf{Reveal}(U, i)$ or $\mathsf{Reveal}(U', j)$ with $U' \in \mathsf{pid}_U^i$ or the adversary queried $\mathsf{Corrupt}(V)$ (with $V \in \mathsf{pid}_U^i$) before a query of the form $\mathsf{Send}(U, i, *)$ or $\mathsf{Send}(U', j, *)$ where $U' \in \mathsf{pid}_U^i$.

Finally adversary outputs a guess bit b'. Such an adversary is said to win the game if $b = b'$ where b is the hidden bit used by the Test oracle.

Let Succ denote the event that the adversary $\mathcal{A}$ wins the game for a protocol XP. We define

$$\mathsf{Adv}_{\mathcal{A},\mathsf{XP}} := |2\ \mathsf{Prob}[\mathsf{Succ}] - 1|$$

to be the advantage of the adversary $\mathcal{A}$ in attacking the protocol XP.

The protocol XP is said to be a *secure unauthenticated group key agreement* (KA) protocol if there is no polynomial time *passive* adversary with non-negligible advantage. In other words, for every probabilistic, polynomial-time, 0/1 valued algorithm $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A},\mathsf{XP}} < \frac{1}{M^L}$ for every fixed

$L > 0$ and sufficiently large integer M. We say that protocol XP is a *secure authenticated group key agreement* (AKA) protocol if there is no polynomial time *active* adversary with non-negligible advantage.

Next we define

$\mathsf{Adv}^{\mathsf{KA}}_{\mathsf{XP}}(t, q_E)$:= the maximum advantage of any passive adversary attacking protocol XP, running in time t and making q_E calls to the $\mathsf{Execute}$ oracle.

$\mathsf{Adv}^{\mathsf{AKA}}_{\mathsf{XP}}(t, q_E, q_S)$:= the maximum advantage of any active adversary attacking protocol XP, running in time t and making q_E calls to the $\mathsf{Execute}$ oracle and q_S calls to the Send oracle.

$\mathsf{Adv}^{\mathsf{AKA}}_{\mathsf{XP}}(t, q_E, q_J, q_L, q_S)$:= the maximum advantage of any active adversary attacking protocol XP, running in time t and making q_E calls to the $\mathsf{Execute}$ oracle, q_J calls to Join oracle, q_L calls to the Leave oracle and q_S calls to the Send oracle.

Remark 1.5.1 *Session identity is required to identify a session uniquely and all participants executing a session should hold the same session identity. Conventionally, session identity* sid^i_U *for an instance* Π^i_U *is set to be the concatenation of all (broadcasted) messages sent and received by* Π^i_U *during its course of execution. This essentially assumes that all the partners of* Π^i_U *hold the same concatenation value of sent and received messages which may not be the case in general. Our definition of session identity is different and can be applied for more general protocols.*

Remark 1.5.2 *We will make the assumption that in each session at most one instance of each user participates. Further, an instance of a particular user participates in exactly one session. This is not a very restrictive assumption, since a user can spawn an instance for each session it participates in. On the other hand, there is an important consequence of this assumption. Suppose there are several sessions which are being concurrently executed. Let the session ID's be* $\mathsf{sid}_1, \ldots, \mathsf{sid}_k$. *Then for any instance* Π^i_U, *there is exactly one* j *such that* $(U, i) \in \mathsf{sid}_j$ *and for any* $j_1 \neq j_2$, *we have* $\mathsf{sid}_{j_1} \cap \mathsf{sid}_{j_2} = \emptyset$. *Thus at any particular point of time, if we consider the collection of all instances of all users, then the relation of being in the same session is an equivalence relation whose equivalence classes are the session IDs. Moreover, an instance* Π^i_U *not only knows* U, *but also the instance number* i *– this being achieved by maintaining a counter.*

Chapter 2

Overview of Pairing-Based Cryptographic Protocols

The bilinear pairing such as Weil pairing or Tate pairing on elliptic curves have many applications in design of cryptographic protocols. We have tried to cover different cryptographic protocols based on bilinear pairings which possess proper security proofs in the existing security models.

2.1 Introduction

The concept of identity-based cryptosystem is due to Shamir [48]. Such a scheme has the property that a user's public key is an easily calculated function of his identity, while a user's private key can be calculated for him by a trusted authority, called private key generator (PKG). The ID-based public key cryptosystem can be an alternative for certificate-based public key infrastructure (PKI), especially when efficient key management and moderate security are required.

Earlier bilinear pairings, namely Weil pairing and Tate pairing of algebraic curves were used in cryptography for the MOV attack [43] using Weil pairing and FR attack [29] using Tate pairing. These attacks reduce the discrete logarithm problem on some elliptic curves to the discrete logarithm problem in a finite field. In recent years, bilinear pairings have found positive application in cryptography to construct new cryptographic primitives.

Protocols from pairings can be broadly classified into two types:

- Construction of primitives which apparently cannot be constructed using other techniques (*e.g.* ID-based encryption, non-trivial aggregate signature *etc*).
- Construction of primitives which can be constructed using other techniques, but for which pairings provide improved functionality (*e.g.* Joux's three-party key agreement, threshold scheme, searchable public key encryption *etc*).

Joux [37], in 2000, showed that the Weil pairing can be used in a protocol to construct three-party one-round Diffie-Hellman key agreement. This was one of the breakthroughs in key agreement protocols. After this, Boneh and Franklin [17] presented in Crypto 2001 an ID-based encryption scheme based on properties of bilinear pairings on elliptic curves which is the first fully functional, efficient and provably secure identity-based encryption scheme. In Asiacrypt 2001, Boneh, Lynn and Shacham proposed a basic signature scheme using pairing, the BLS [19] scheme, that has the shortest length among signature schemes in classical cryptography. Subsequently numerous cryptographic schemes based on BLS signature scheme were proposed.

Apart from the three fundamental cryptographic primitives: encryption, signature and key agreement, there are protocol designs for signcryption, threshold decryption, key sharing, identification scheme, chameleon hashes *etc.*

Barreto's pairing based crypto lounge [4] is an excellent compilation of existing work on pairing based cryptography.

2.2 Encryption Schemes

In identity-based public key encryption, the public key distribution problem is eliminated by making each user's public key derivable from some known aspect of his identity, such as his email address. When Alice wants to send a message to Bob, she simply encrypts her message using Bob's public key which she derives from Bob's identifying information. Bob, after receiving the encrypted message, obtains his private key from a third party called a Private Key Generator (PKG) after authenticating himself to PKG and can then decrypt the message. The private key that PKG generates on Bob's query is a function of its master key and Bob's identity.

Shamir [48] introduced this concept of identity-based cryptosystem. The first ID-based encryption was proposed by Boneh and Franklin [17] in 2001 that uses bilinear pairing.

Let G_1, G_2, e be the same as in Definition 1.2.11 of cryptographic bilinear pairings. P is a generator of the additive group G_1 of order q (a large prime), G_2 is a multiplicative group of same order q and e is the bilinear map from $G_1 \times G_1 \to G_2$. We use these notations throughout.

2.2.1 ID-Based Encryption Scheme

(Boneh, Franklin, [17], 2001)

- Protocol Description :

  Setup : Choose $s \in_R Z_q^*$ and set $P_{pub} = sP$. Choose cryptographic hash functions $H_1 : \{0,1\}^* \to G_1^*$ and $H_2 : G_2 \to \{0,1\}^n$, n is the bit length of messages. The master key is s and the global public key is P_{pub}.

  Extract : Given a public identity $\mathsf{ID} \in \{0,1\}^*$, compute the public key $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in G_1$ and the private key $S_{\mathsf{ID}} = sQ_{\mathsf{ID}}$. The computation $Q_{\mathsf{ID}} = H_1(\mathsf{ID})$ maps an arbitrary

string to a point of the group G_1. This operation is called Map-to-point and is more expensive than computation of usual message digest.

Encrypt : Choose a random $r \in Z_q^*$, set the cipher text for the message M to be $C = \langle rP, M \oplus H_2(g_{\mathsf{ID}}^r) \rangle$ where $g_{\mathsf{ID}} = e(Q_{\mathsf{ID}}, P_{pub})$.

Decrypt : Given $C = \langle U, V \rangle$, compute $V \oplus H_2(e(S_{\mathsf{ID}}, U))$.

- Assumption :
  BDH problem is hard.

- Security :
  This is the basic scheme. Security against adaptive chosen cipher text attack in the random oracle model under the BDH assumption is obtained after the Fujisaki-Okamoto [30] transformation.

- Efficiency :

  Setup : 1 scalar multiplication in G_1.

  Extract : 1 Map-to-point hash operation; 1 scalar multiplication in G_1.

  Encrypt : 1 Map-to-point hash operation; 1 scalar multiplication in G_1; 1 hash function (H_2) evaluation; 1 XOR operation; 1 pairing computation; 1 group exponent in G_2.

  Decrypt : 1 hash function (H_2) evaluation; 1 XOR operation; 1 pairing computation.

2.2.2 Searchable Public Key Encryption

(Boneh, Crescenzo, Ostrovsky, Persiano, [16], 2003)

Suppose Alice wishes to read her email on a number of devices : laptop, desktop, pager, etc. Alice's mail gateway is supposed to route email to the appropriate device based on the keywords in the email. Suppose Bob sends an email with keyword "urgent". The gateway routes the email to Alice's pager, after testing whether the email contains this keyword "urgent" without learning anything else about the mail. This mechanism is referred to as *Searchable Public Key Encryption* (SPKE).

To send a message M with keywords $W_1, \ldots, W_n$, Bob sends

$$E_{A_{pub}}(M) | \mathsf{SPKE}(A_{pub}, W_1) | \ldots | \mathsf{SPKE}(A_{pub}, W_n)$$

where $E_{A_{pub}}(M)$ is the encryption of M using Alice's public key A_{pub}. The point of searchable encryption is that given $\mathsf{SPKE}(A_{pub}, W')$ and a certain trapdoor T_W (that is given to the gateway by Alice), the gateway can test whether $W = W'$. If $W \neq W'$ the gateway learns nothing more about W'.

A SPKE scheme using bilinear map :

- Protocol Description :

  KeyGen : Choose $s \in_R Z_q^*$ and set $P_{pub} = sP$. The secret key is s and the public key is P_{pub}. Let K be the set of all keywords and $H_1 : K \to G_1$, $H_2 : G_2 \to Z_q^*$ be two hash functions.

  SPKE : Given a keyword W and the public key P_{pub}, choose a random $r \in Z_q^*$ and output $\langle rP, H_2(e(H_1(W), P_{pub})^r)\rangle$.

  Trapdoor : Given a keyword W and the secret key s, output $T_W = sH_1(W)$.

  Test : Given Trapdoor T_W, a SPKE $S = \langle U, V\rangle$ and the public key P_{pub}, test if $V = H_2(e(T_W, U))$. If true, output yes, else output no.

- Assumption :
  BDH problem is hard.

- Security :
  Semantically secure against a chosen keyword attack in the random oracle model assuming BDH problem is intractable.

- Efficiency :

  KeyGen : 1 scalar multiplication in G_1.

  SPKE : 1 Map-to-point hash operation; 1 scalar multiplication in G_1; 1 hash function (H_2) evaluation; 1 pairing computation; 1 group exponent in G_2.

  Trapdoor : 1 scalar multiplication in G_1.

  Test : 1 pairing computation; 1 hash function (H_2) evaluation.

2.2.3 An ID-Based Encryption Scheme Without Random Oracle

(Waters [49], 2005)

- **Protocol Description** :

  Setup : Choose a secret $\alpha \in Z_q^*$ at random and set $P_1 = \alpha P$. Choose P_2 randomly in G_1. Additionally, choose a random value $Q' \in G_1$ and a random n-length vector $U = (Q_i)$, whose elements are chosen at random from G_1. The published public parameters are P, P_1, P_2, Q' and U and the master secret is αP_2.

  Extract : Let $\mathsf{ID} \in \{0,1\}^n$ be an n-bit string representing a public identity. Let v_i be the i-th bit of ID and $\mathcal{V} \subseteq \{1, \ldots, n\}$ be the set of all i for which $v_i = 1$. Select a random $r \in Z_q^*$. Output the private key

$$S_{\mathsf{ID}} = (\alpha P_2 + r(Q' + \sum_{i \in \mathcal{V}} Q_i), rP).$$

Encrypt : To encrypt a message $M \in G_2$ under public identity ID, pick a random $t \in Z_q^*$ and output the cipher test

$$C = (e(P_1, P_2)^t M, tP, t(Q' + \sum_{i \in \mathcal{V}} Q_i)).$$

Decrypt : To decrypt a cipher text $C = (C_1, C_2, C_3)$ using the private key $S_{\mathsf{ID}} = (A, B)$, output

$$C_1 \frac{e(A, C_3)}{e(B, C_2)} = M.$$

- **Assumption :**
  DBDH problem is hard.
- **Security :**
  Secure against adaptive chosen ciphertext attacks without the random oracle model.
- **Efficiency :**
  If the value $e(P_1, P_2)$ is cached then encryption requires on average $\frac{n}{2}$ (and at most n) group operations in G_1, 2 exponentiations in G_1, 1 exponentiation in G_2 and 1 group operation in G_2. Decryption requires 2 bilinear map computation, 1 group operation in G_2 and 1 inversion in G_2.

Note : Boneh and Boyen [14] presented an identity-based encryption scheme that is provably secure in the selective-ID model without random oracle. In [15], Boneh and Boyen describe a scheme that is fully secure without random oracles, but is too inefficient to be of practical use. The construction of Waters [49] presented here is a modified version of Boneh and Boyen's scheme in [14] that makes the identity-based scheme fully secure and is efficient as compared to the scheme in [15]. Water's scheme has been generalized by Chatterjee and Sarkar [23] that has shorter public parameters.

2.3 Signature Schemes

Digital signatures are one of the most important cryptographic primitives. In traditional public key signature algorithms, the binding between the public key and the identity of the signer is obtained via a digital certificate. Shamir [48] first noticed that it would be more efficient if there was no need for such bindings, in that case given the user's identity, the public key could be easily derived using some public deterministic algorithm. This makes efficient ID-based signature schemes desirable. In ID-based signature schemes, verification function is easily obtained from the identity, possibly the same key and the same underlying computation primitives can be used. Boneh, Lynn, Shacham [19] proposed a pairing based short signature scheme in 2001. This was followed by a large number of pairing based signature schemes for different applications.

2.3.1 BLS Short Signature Scheme

(Boneh, Lynn, Shacham, [19], 2001)

Short signatures are needed in environments with space and bandwidth constraints. For example, when a human is asked to type in a digital signature the shortest possible signatures are desired. Two most frequently used signature schemes are RSA and DSA. If one uses 1024 bit modulus, RSA signatures are 1024 bit long and standard DSA or ECDSA (elliptic curve DSA) signatures are 320 bit long. These signatures are too long to be keyed. The following signature scheme provides short signature of length approximately 160 bits with a level of security similar to 320 bit DSA signatures.

- Protocol Description :

  KeyGen : Let $H : \{0,1\}^* \rightarrow G_1$ be a Map-to-point hash function. The secret key is $x \in_R Z_q^*$ and the public key is $P_{pub} = xP$ for a signer.

  Sign : Given secret key x and a message $m \in \{0,1\}^*$, compute the signature $\sigma = xH(m)$.

  Verify : Given public key $P_{pub} = xP$, a message m and a signature σ, verify $e(P, \sigma) = e(P_{pub}, H(m))$.

- Assumption :
  Existence of GDH group.

- Security :
  Secure against existential forgery under adaptive chosen message attack in the random oracle model assuming CDH problem is hard in G_1.

- Efficiency :

  KeyGen : 1 scalar multiplication in G_1.

  Sign : 1 Map-to-point hash operation; 1 scalar multiplication in G_1.

  Verify : 1 Map-to-point hash operation; 2 pairing computations.

2.3.2 Multi signature Scheme

(Boldyreva, [11], 2003)

A multi-signature scheme allows any subgroup of a group of users to jointly sign a document such that a verifier is convinced that each member of the subgroup participated in signing.

- Protocol Description :

KeyGen : Let $H; \{0,1\}^* \rightarrow G_1$ be a Map-to-point hash function. Consider a set U of n users. The secret key is $x_i \in_R Z_q^*$ and the public key is $P_{pub_i} = x_i P$, for user $u_i \in U, 1 \leq i \leq n$.

Multi-signature Creation : Any user $u_i \in U$ with secret key x_i that wishes to participate in signing a message $m \in \{0,1\}^*$, computes $\sigma_i = x_i H(m)$ and sends it to a designated signer D (which can be implemented by any user). Let $L = \{u_{i_1}, \ldots u_{i_l}\} \subseteq U$ be a subset of users contributed to the signing. After getting all the σ_j for $j \in J = \{i_1, \ldots, i_l\}$, D computes the multi-signature $\sigma = \sum_{j \in J} \sigma_j$ and outputs (m, L, σ).

Multi-signature Verification : Given $T = (m, L, \sigma)$ and the list of public keys of the users in L: $P_{pub_j} = x_j P, j \in J = \{i_1, \ldots, i_l\}$, the verifier computes $P_{pub_L} = \sum_{j \in J} P_{pub_j} = \sum_{j \in J} x_j P$ and verifies $e(P, \sigma) = e(P_{pub_L}, H(m))$.

- Assumption :
  Existence of GDH group.

- Security :
  Secure against existential forgery under chosen message attack in the random oracle model under the assumption that the CDH problem is hard in G_1.

- Efficiency :

  KeyGen : n scalar multiplications in G_1.

  Multi-signature Creation : If $l \leq n$ users are participating in signing, then 1 Map-to-point hash operation; l scalar multiplications in G_1; $(l-1)$ additions in G_1.

  Multi-signature Verification : If number of users in the list L is l, then $(l-1)$ additions in G_1; 2 pairing computations.

2.3.3 Aggregate Signature

(Boneh, Gentry, Lynn, Shacham [18], 2003)

An aggregate signature scheme is a digital signature that supports aggregation : Given n signatures on n distinct messages m_i from n distinct users i, $1 \leq i \leq n$, it is possible to aggregate all these signatures into a single short signature. This single signature and the n original messages $m_i, 1 \leq i \leq n$ will convince the verifier that user i indeed signed message m_i , $1 \leq i \leq n$.

- Protocol Description :

  KeyGen : Consider the Co-GDH setup. Let U be a set of n users and $H : \{0,1\}^* \rightarrow G_2$ be a Map-to-point hash function. The secret key is $x_i \in_R Z_q^*$ and the public key is $P_{pub_i} = x_i P_1$ for user $u_i \in U, 1 \leq i \leq n$.

  Aggregation : User $u_i \in U$ signs message $m_i \in \{0,1\}^*$ to generate BLS signature $\sigma_i = x_i H(m_i)$, $1 \leq i \leq n$. The messages m_i must be all distinct. The aggregate signature is $\sigma = (\sigma_1 + \sigma_2 + \cdots + \sigma_n) \in G_2$.

Aggregate verification : Given public keys P_{pub_i}, distinct messages $m_i, 1 \leq i \leq n$ and an aggregate signature σ, verify $e(P_1, \sigma) = \prod_{i=1}^{n} e(P_{pub_i}, H(m_i))$.

- Assumption :
  Existence of Co-GDH group and a bilinear map.
- Security :
  Secure against existential forgery in the aggregate chosen key model assuming that the Co-CDH problem is hard in (G_1, G_2).
- Efficiency :

  KeyGen : n scalar multiplications in G_1.

  Aggregation : n Map-to-point hash operations; n scalar multiplications in G_2; $(n-1)$ additions in G_2.

  Aggregate verification : n Map-to-point hash operations; $(n+1)$ pairing computations.

2.3.4 ZSS Short Signature Scheme

(Zhang, Safavi-Naini, Susilo, [50], 2004)

- Protocol Description :

  KeyGen : Let $H : \{0,1\}^* \rightarrow Z_q^*$ be a hash function. The secret key is $x \in_R Z_q^*$ and the public key is $P_{pub} = xP$ for a signer.

  Sign : Given a secret key x and a message $m \in \{0,1\}^*$, compute signature $S = \frac{1}{H(m)+x} P$.

  Verify : Given a public key P_{pub}, a message m and a signature S, verify $e(H(m)P + P_{pub}, S) = e(P, P)$.
- Assumption :
  $(k+1)$-exponent problem is hard.
- Security :
  Existentially unforgeable under an adaptive chosen message attack in the random oracle model assuming that $(k+1)$-exponent problem is hard.
- Efficiency :

  KeyGen : 1 scalar multiplication in G_1.

  Sign : 1 inversion in Z_q^*; 1 hash function (H) evaluation; 1 scalar multiplication in G_1.

  Verify : 2 pairing computation (one of which, $e(P, P)$ can be precomputed); 1 scalar multiplication in G_1; 1 hash function (H) evaluation; 1 addition in G_1.
  This scheme is more efficient than BLS scheme as it requires less pairing computation and no computation of the expensive special hash function Map-to-point that encodes finite strings to elements of group G_1.

2.3.5 ID-Based Signature from Pairing

(Hess, [36], 2002)

- Protocol Description :

  Setup : Choose $s \in_R Z_q^*$ and set $P_{pub} = sP$. The master key is s and the global public key is P_{pub}. Let $H_1 : \{0,1\}^* \rightarrow G_1$ be a Map-to-point hash function and $H : \{0,1\}^* \times G_2 \rightarrow Z_q^*$ be another hash function.

  Extract : Given a public identity $\mathsf{ID} \in \{0,1\}^*$, compute the public identity $Q_{\mathsf{ID}} = H_1(\mathsf{ID})$ and the secret key $S_{\mathsf{ID}} = sQ_{\mathsf{ID}}$.

  Sign : Given a secret key S_{ID} and a message $m \in \{0,1\}^*$, the signer chooses an arbitrary $P_1 \in G_1^*$ and a random $k \in Z_q^*$ and computes

  1. $r = e(P_1, P)^k$,
  2. $v = H(m, r)$,
  3. $u = vS_{\mathsf{ID}} + kP_1$.

  The signature is then the pair $(u, v) \in G_1 \times Z_q^*$.

  Verify : Given a public key Q_{ID}, a message m and a signature (u, v) the verifier computes :

  1. $r = e(u, P)e(Q_{\mathsf{ID}}, -P_{pub})^v$
  2. Accept the signature if and only if $v = H(m, r)$.

  Assumption :
  Weak-DH problem is hard.

  Security :
  Secure against existential forgery under adaptive chosen message attack in the random oracle model assuming Weak-DH problem is hard.

  Efficiency :

  Setup : 1 scalar multiplication in G_1.

  Extract : 1 Map-to-point hash operation; 1 scalar multiplication in G_1.

  Sign : The signing operation can be optimized by the signer pre-computing $e(P_1, P)$ for P_1 of his choice, for example $P_1 = P$, and storing this value with the signing key. This means that the signing operation involves one exponentiation in the group G_2, one hash function (H) evaluation and one simultaneous multiplication in the group G_1.

  Verify : The verification operation requires one exponentiation in G_2, one hash function (H) evaluation and two evaluations of the pairing. One of the pairing evaluation can be eliminated, if a large number of verifications are to be performed for the same identity, by pre-computing $e(Q_{ID}, -P_{pub})$.
  This scheme is very efficient in terms of communication requirements. One needs to transmit one element of the group G_1 and one element of Z_q^*.

2.3.6 Short Signature Scheme Without Random Oracle

(Boneh, Boyen [13], 2004)

- Protocol Description :

  KeyGen : The secret key is $(x, y) \in_R Z_q^* \times Z_q^*$ and the public key is (P, U, V) where $U = xP$ and $V = yP$ for a signer. The messages are assumed to be elements of Z_q^*.

  Sign : Given a secret key (x, y), a message $m \in Z_q^*$, choose a random $r \in Z_q^*$ and compute $\sigma = \frac{1}{x+m+yr}P$. Here $\frac{1}{x+m+yr}$ is computed modulo q and the unlikely event $x+m+yr = 0$ is avoided by choosing a different r. The signature is (σ, r).

  Verify : Given a public key (P, U, V), a message $m \in Z_q^*$ and a signature (σ, r), verify $e(\sigma, U + mP + rV) = e(P, P)$.

- Assumption :
  q-SDH problem is hard.

- Security :
  Secure against existential forgery under chosen message attack under SDH assumption and without using the random oracle model.

- Efficiency :

  KeyGen : 2 scalar multiplications in G_1.

  Sign : 1 inversion in Z_q^*; 1 scalar multiplication in G_1.

  Verify : 2 scalar multiplication in G_1; 2 additions in G_1; 2 pairing computations one of which, $e(P, P)$ can be precomputed.

Note : Recently, Waters [49] proposed an efficient signature scheme that depends only upon the CDH assumption in the standard model (*i.e.* without using any random oracle).

2.4 Key Agreement Schemes

Key agreement is one of the fundamental cryptographic primitives. This is required when two or more parties want to communicate securely. In one of the breakthroughs in key agreement, Joux [37] proposed a three party single round key agreement protocol using pairing. This was the first positive application of bilinear pairing in cryptography. Afterwords, pairings were used widely to get a large number of cryptographic protocols some of which have been previously mentioned. Several key agreement protocols were proposed that prevents man-in-the-middle attack against a passive adversary. These protocols are called unauthenticated. The protocols for authenticated key agreement enables a group of parties within a large and completely insecure public network to establish a common secret key and furthermore ensures that they are indeed sharing this key

with each other. Achieving authenticated key agreement are crucial for allowing symmetric-key encryption/authentication of data among the parties. Authenticated key agreement protocols are the basic tools for group-oriented and collaborative applications such as, distributed simulation, multi-user games, audio or video-conferencing, and also peer-to-peer application that are likely to involve a large number of users. These are used to construct secure channels which are the base for designing, analyzing and implementing higher-level protocols in a modular approach. A formal model of security for group authenticated key agreement can be found in [20].

2.5 Threshold Schemes

The idea behind the (t, n)-threshold cryptosystem approach is to distribute secret information (*i.e.* the secret key) and computation (*i.e.* signature generation or decryption) among n parties in order to remove single point failure. The goal is to allow a subset of more than t players to jointly reconstruct a secret and perform the computation while preserving security even in the presence of an active adversary which can corrupt up to t (a threshold) parties. The secret key is distributed among n parties with the help of a trusted dealer or without it by running an interactive protocol among all parties.

2.5.1 Threshold Signature Scheme

(Boldyreva, [11], 2003)

- Protocol Description :

  KeyGen : Let $H : \{0,1\}^* \rightarrow G_1$ be a Map-to-point hash function. Suppose there are n servers $u_i, 1 \leq i \leq n$. The private key $x \in Z_q^*$ is shared among these users using Shamir's secret sharing scheme such that any subset S of $t+1$ servers can reconstruct x using Lagrange interpolation : $x = \sum_{i \in S} L_i x_i$, where $L_i = \prod_{j \in S} \frac{-x_j}{(x_i - x_j)}$ is the Lagrange co-efficient, x_i is the private key share and $P_{pub_i} = x_i P$ is the public key share of user u_i.

  Signature Share Generation : To sign a message $m \in \{0,1\}^*$, user u_i outputs $\sigma_i = x_i H(m)$.

  Signature Share Verification : Given m, σ_i, P_{pub_i}, anyone can check whether user u_i is honestly behaving in giving it's share σ_i of signature by checking $e(P, \sigma_i) = e(P_{pub_i}, H(m))$. If σ_i passes through this test, call it an acceptable share.

  Signature Reconstruction : Suppose a set S of $(t+1)$ honest servers are found and accordingly $(t+1)$ acceptable shares $\sigma_i, i \in S$. The resulting signature on m is $\sigma = \sum_{i \in S} L_i \sigma_i$. The correctness of the scheme is easy to verify since $e(P, \sigma) = e(H(m), xP)$.

- Assumption :
  Existence of GDH group.

- Security :
  Secure in the random oracle model against an adversary which is allowed to corrupt any $t < n/2$ players under the assumption that the underlying group is GDH.

- Efficiency :

  KeyGen : n scalar multiplications in G_1.

  Signature Share Generation : For each user, 1 Map-to-point hash operation; 1 scalar multiplication in G_1.

  Signature Share Verification : 2 pairing computations; 1 Map-to-point hash operation.

  Signature Reconstruction : $(t+1)$ scalar multiplications in G_1; t additions in G_1; $(t+1)$ Lagrange co-efficient (L_i) computations.

2.5.2 Signcryption Schemes

The idea of this primitive is to perform encryption and signature in a single logical step in order to obtain confidentiality, integrity, authentication and non-repudiation more efficiently than the sign-then-encrypt approach. The drawback of this latter situation is to expand the final cipher text size and increase the sender's and receiver's computing time which may be impractical for low bandwidth network. Malone-Lee [40] defines extended security notions for ID-based signcryption schemes.

(a) Identity-Based Signcryption
(Malone-Lee [40], 2003)

- Protocol Description

  Setup : Choose $s \leftarrow^R Z_q^*$ and set $P_{\mathsf{Pub}} = sP$. The master key generated by the trusted party is s and the public key is P_{pub}. Consider three hash functions : $H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : \{0,1\}^* \rightarrow Z_q^*$ and $H_3 : G_2 \rightarrow \{0,1\}^l$.

  Extract(ID) : Compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID})$, $S_{\mathsf{ID}} = sQ_{\mathsf{ID}}$. The secret key corresponding to identity $\mathsf{ID} \in \{0,1\}^*$ is S_{ID} and the public key is Q_{ID}.

  Signcrypt($S_{\mathsf{ID}_a}, \mathsf{ID}_b, m$) : For a message $m \in \{0,1,\}^l$, compute $Q_{\mathsf{ID}_b} = H_1(\mathsf{ID}_b)$. Choose $x \leftarrow^R Z_q^*$ and set $U = xP$. Compute $r = H_2(U||m)$, $W = xP_{pub}$, $V = rS_{\mathsf{ID}_a} + W$, $y = e(W, Q_{\mathsf{ID}_b})$, $\kappa = H_3(y)$, $c = \kappa \oplus m$. Send $\sigma = (c, U, V)$

  Unsigncrypt ($\mathsf{ID}_a, S_{\mathsf{ID}_b}, \sigma$) : Compute $Q_{\mathsf{ID}_a} = H_1(\mathsf{ID}_a)$. Parse σ as (c, U, V). Compute $y = e(S_{\mathsf{ID}_b}, U)$, $\kappa = H_3(y)$, $m = \kappa \oplus c$, $r = H_2(U||m)$. Return m if and only if $e(V, P) = e(Q_{\mathsf{ID}_a}, P_{pub})^r \; e(U, P_{pub})$.
  Consistency constraint :
  if $\sigma = \mathsf{Signcrypt}(S_{\mathsf{ID}_a}, \mathsf{ID}_b, m)$, then $m = \mathsf{Unsigncrypt}(\mathsf{ID}_a, S_{\mathsf{ID}_b}, \sigma)$. This scheme is the result of a combination of the simplified version of Boneh and Franklin's IBE cryptosystem with a variant of Hess's identity based signature.

- Assumption :
  BDH problem is hard.
- Security :
  This protocol achieves the security IND-ISC-CCA (indistinguishability of identity-based signcryptions under chosen cipher text attack) and also the security EF-ISC-ACMA (existentially unforgeability of identity-based signcryptions under adaptive chosen message attack) in the random oracle model assuming BDH problem is hard.
- Efficiency :

  Setup : 1 scalar multiplication in G_1.

  Extract : 1 Map-to-point hash operation; 1 scalar multiplication in G_1.

  Signcrypt : 1 Map-to-point hash operation; 3 scalar multiplications in G_1; 1 hash function H_2 evaluation; 1 pairing computation; 1 hash function H_3 evaluation; 1 XOR operation, 1 addition in G_1.

  Unsigncrypt : 1 Map-to-point hash operation; 4 pairing computations; 1 hash function H_3 evaluation; 1 XOR operation; 1 hash function H_2 evaluation; 1 exponentiation in G_2. The size of the cryptogram is $n + 2|G_1|$ when a message of n-bit is sent.

2.5.3 Identification Scheme based on GDH

(Kim, Kim [38], 2002)

Identification scheme is a very important and useful cryptographic tool. It is an interactive protocol where a prover $\mathcal{P}$ tries to convince a verifier $\mathcal{V}$ of his identity. Only $\mathcal{P}$ knows the secret value corresponding to his public one and the secret value allows to convince $\mathcal{V}$ of his identity.

- Protocol Description :

  KeyGen : Choose randomly $a, b, c \in Z_q^*$ and compute aP, bP, cP, $v = e(P, P)^{abc}$. The secret key is (a, b, c) and make aP, bP, cP, v public.

  Protocol actions between $\mathcal{P}$ and $\mathcal{V}$: This scheme consists of several rounds, each of which is performed as follows :

  1. $\mathcal{P}$ chooses randomly $r_1, r_2, r_3 \in Z_q^*$ and computes $x = e(P, P)^{r_1 r_2 r_3}$, $Q_1 = r_1 P$, $Q_2 = r_2 P$ and $Q_3 = r_3 P$ and sends $\langle x, Q_1, Q_2, Q_3 \rangle$ to $\mathcal{V}$.
  2. $\mathcal{V}$ picked $w \in Z_q^*$ at random and sends w to $\mathcal{P}$.
  3. $\mathcal{P}$ computes $y = e(wP, P)^{abc}\ e(P, P)^{r_1 r_2 r_3}$ and sends to $\mathcal{V}$; $\mathcal{V}$ accepts if $y = v^w\ x$ and rejects otherwise.
- Assumption : Existence of GDH group.
- Security :
  Secure against active attacks assuming that the underlying group is a GDH group.

- Efficiency :

  KeyGen : 3 scalar multiplications in G_1; 1 pairing computations.

  Protocol actions between $\mathcal{P}$ *and* $\mathcal{V}$: 3 pairing computations and 4 scalar multiplications in G_1 for $\mathcal{P}$; 1 exponentiation in G_2 and 1 multiplication in G_2 for $\mathcal{V}$.

2.5.4 Other Signature Schemes

There are a large number of cryptographic protocols that uses pairings. Discussing every protocol is beyond the scope of the paper. This subsection includes a list of few other interesting signature schemes that have various cryptographic applications in digital world.

1. Optimistic Fair Exchange [26].
3. A New Varifiably Encrypted Signature Scheme [51].
4. Partially Blind Signature Scheme [51].
5. ID-Based Group Signature Scheme [25].
6. Delegation-By-Certificate Proxy Signature Scheme [12].
7. Hierarchical ID-Based Signatures (HIDS) Scheme [32].

We refer Barreto's Pairing-Based Crypto Lounge [4] for an overview (references) of recent developments in cryptosystems based on pairings.

2.6 Conclusion

Several cryptographic primitives using pairings have been described and many have been left out. This is a very active area and almost every conference includes some new proposals involving pairing.

Chapter 3

Efficient Computation of Tate Pairing

3.1 Introduction

The first major breakthrough in key agreement was obtained by Joux [37]who used Weil pairing on elliptic curves to obtain a one-round three party key agreement protocol generalising the Diffie-Hellman two-party key agreement protocol. In 2003, Boneh and Franklin [17] used Weil pairing to obtain the first feasible identity-based encryption scheme–thus settling a long standing problem of Shamir. This has led to a spurt of activities in pairing-based cryptography. For a recent survey of such activities see the survey of Dutta *et al* [27]. Thus pairing-based crytographic schemes are recognised as a major area of research in recent times. The only two known (bilinear) pairing functions are the Weil and Tate pairings. The Tate pairing is believed to be more efficient than Weil pairing and in this note we shall exclusively deal with Tate pairing. A major impediment in the implementation of pairing-based crytosystems seems to be the computation of the pairing map which is quite expensive. Here we will survey some of the most recent works in the efficient computation of Tate pairing

3.2 Tate Pairing and Miller's Algorithm

Let m to be an odd positive integer, $\mathbb{F}_q$ the finite field with q elements, where $q = p^m$, E is an elliptic curve over $\mathbb{F}_p$ and $E(\mathbb{F}_q)$ is the set of all $\mathbb{F}_q$-rational points of E. Let r be a large prime divisor of the curve order $\#E(\mathbb{F}_q)$, such that r is coprime to $q, q-1$ and for some $k > 0$, $r|q^k - 1$ but $r \not| q^s - 1$ for any $1 \leq s < k$; k is called the security multiplier (or MOV/FR degree). Suppose P is a point of order r on the elliptic curve $E(\mathbb{F}_q)$ and Q is a point of same order on the elliptic curve $E(\mathbb{F}_{q^k})$, linearly independent of P. We denote the (modified) Tate pairing of order r as $e_r(P, Q) \in \mathbb{F}_{q^k}$, which we shall define below.

Recall that a divisor is a formal sum of the form $D = \sum_{P \in E} a_P \langle P \rangle$, where $a_P \in \mathbb{Z}$. The degree of a divisor D is $deg(D) = \sum_{P \in E} a_P$. The set of divisors forms an abelian group by the addition of their coefficients in the formal sum. Let f be a rational function on E,i.e. a rational function *modulo* the defining equation for the curve. Then the divisor of f, $\langle f \rangle = \sum_P ord_P(f) \langle P \rangle$, where $ord_P(f)$ is the order of the zero or pole of f(i.e. zero of $1/f$) at P. Recall also that a divisor $D = \sum_{P \in E} a_P \langle P \rangle$ is a principal divisor if and only if it is a divisor of degree 0 (*zero divisor*) and $\sum_{P \in E} a_P P = \mathcal{O}$ (Theorem 1.2.6). If D is principal then there is some rational function f such that $D = \langle f \rangle$. Two divisors D_1 and D_2 are said to be equivalent if $D_1 - D_2$ is a principal divisor.

Let $P \in E(\mathbb{F}_q)$ and $Q \in E(\mathbb{F}_{q^k})$, which are linearly independent. Let $\mathcal{A}_\mathcal{P}$ be a divisor equivalent to $\langle P \rangle - \langle \mathcal{O} \rangle$ (similarly define $\mathcal{A}_\mathcal{Q}$). Then it is easy to see that $r\mathcal{A}_\mathcal{P}$ is principal. Thus there is a rational function f_P with $\langle f_P \rangle = r\mathcal{A}_\mathcal{P} = r\langle P \rangle - r\langle \mathcal{O} \rangle$.

The (modified) Tate pairing of order r is defined as

$$e_r(P, Q) = f_P(\mathcal{A}_\mathcal{Q})^{(q^k - 1)/r}.$$

To compute $f_P(\mathcal{A}_\mathcal{Q})$, $Q \neq \mathcal{O}$ one uses Miller's algorithm [44]. Let f_a be a rational function with divisor $\langle f_a \rangle = a\langle P \rangle - \langle aP \rangle - (a-1)\langle \mathcal{O} \rangle, a \in Z$. It can be shown that $f_{2a}(Q) = f_a(Q)^2 . g_{aP,aP}(Q)/g_{2aP}(Q)$ and

$$f_{a+b}(Q) = f_a(Q) f_b(Q) . g_{aP,bP}(Q)/g_{(a+b)P}(Q) \tag{3.1}$$

where, $g_{aP,bP}$(resp. $g_{aP,aP}$) is the line joining aP, bP (resp. tangent to $E(\mathbb{F}_q)$ at aP)— it intersects $E(\mathbb{F}_q)$ at the point $-(a+b)P$ (resp. $-2aP$), and g_{cP} is the (vertical) line that intersects $E(\mathbb{F}_q)$ at cP and $-cP$. Equation (3.1) is known as **Miller's Formula**. Now,

$$\langle f_r \rangle = r\langle P \rangle - \langle rP \rangle - (r-1)\langle \mathcal{O} \rangle = \langle f_P \rangle,$$

since $rP = \mathcal{O}$. Given P and the binary representation of r, Miller's algorithm computes $f_P(Q) = f_r(Q)$ in $\lg r$ steps by the standard double-and-add through line-and-tangent method for elliptic curve scalar multiplication. Under the condition, $r \not| (q-1)$ we can further have $e_r(P, Q) = f_P(Q)^{(q^k - 1)/r}$ for $Q \neq \mathcal{O}$, as long as $k > 1$ (See Theorem 1 of [5]).

Thus for an efficient computation of the Tate pairing, it is enough to have an efficient algorithm for computing $f_r(Q)$. Several optimizations for Miller's algorithm have been suggested. The first of these were given by Baretto *at al* [5].

They observed that for the supersingular curve

$$E_1 : y^2 = x^3 + (1-b)x + b, b \in \{0, 1\};$$

where the underlying field is $\mathbb{F}_p, p > 3$, one choses the point $Q \in E(\mathbb{F}_p)$ and then obtain the point $Q' = \phi(Q) \in \mathbb{F}_{p^2}$, linearly independent of P, where ϕ is the *distortion map*

$$\phi(x, y) = (-x, \mathbf{i}y),$$

Table 3.1: Algorithm 1

Algorithm: **BKLS (Modified Miller) algorithm**

input : $P, Q \in E(\mathbb{F}_p)$, and $r = r_t r_{t-1} \dots r_0; r_t = 1, r_i \in \{0, 1\}$.
output : $f_r(\phi(Q)) = f_P(\phi(Q))$
Set $f \longleftarrow 1$ and $V \longleftarrow P$
for $j \leftarrow t-1$ **down to** 0 **do**{
set $f \leftarrow f^2.g_{V,V}(\phi(Q))$ and $V \leftarrow 2V$
if $r_j = 1$ then set $f \longleftarrow f.g_{V,P}(\phi(Q))$ and $V \longleftarrow V + P$
}
return f

with $\mathbf{i}^2 = -1$. In such a situation, they have shown that the denominator in Miller's Formula is irrelevant in the final computation of the Tate pairing and hence can be ignored while computing f_r. For the curve E_1 note that $\#E(\mathbb{F}_p) = p + 1$ and the MOV degree is $k = 2$. Thus the modified Miller's algorithm for E_1 is given by Table 3.1.

Recently, in Chaterjee *et al*[24], this has been extended to other supersingular curves as well. There it was shown that the best optimization is obtained by encapsulating iterated point doubling and line computation as well as point addition and line computation in Jacobian coordinates.

3.3 Optimizations in characteristic 3

In this section we consider fields of characteristic 3 and take $p = 3$.

For supersingular curves over fields of characteristic 3 , one can obtain even better optimizations. Several optimizations in the computation of Tate pairing are possible [5], [31]. For the supersingular elliptic curve

$$E_2 : y^2 = x^3 - x + b, \qquad b \in \{-1, 1\} \tag{3.2}$$

one can achieve the highest possible MOV degree, $k = 6$ [42]. If $P = (x, y)$ is a point on $E_2(\mathbb{F}_q)$ then $3P = (x^9 - b, -y^9)$. Since, cubing is almost free (linear time) in characteristic three, point tripling is very efficient here. So, the double-and-add method of Miller's algorithm can be replaced by the more efficient triple-and-add/subtract method [5] with the signed ternary representation of r.

In the implementation of Tate pairing over E_2, as noted above, the usual practice is to take $Q \in E(\mathbb{F}_q)$ of order r and then use a distortion map ϕ, to get a point $\phi(Q) \in E(\mathbb{F}_{q^k})$ of order r which is linearly independent of P. A distortion map for E_2 is given as $\phi_2(x, y) = (\rho - x, iy)$

Table 3.2: Algorithm 2

Algorithm: **Algorithm of Duursma-Lee to compute** $f_P(\phi(Q))$**[28]**

input : $P = (\alpha, \beta)$ and $Q = (x, y)$
output : $f_P(\phi(Q))$
$f \longleftarrow 1$
for $j \leftarrow 1$ to m do
$\alpha \leftarrow \alpha^3$, $\beta \leftarrow \beta^3$
$\mu \leftarrow \alpha + x + b$
$\lambda \leftarrow \beta y\bar{i} - \mu^2$
$g \longleftarrow \lambda - \mu\rho - \rho^2$, $f \longleftarrow f.g$
$x \leftarrow x^{1/3}$, $y \leftarrow y^{1/3}$
end for
return f

where $\rho, i \in \mathbb{F}_{q^6}$ and $\rho^3 - \rho - b = 0, \quad i^2 = -1$ [5],[31]. Another major finding in [5] is that, for E_2 also under the distortion map ϕ_2, we can completely ignore the denominator part in the computation of Tate pairing. All these optimizations make E_2 an attractive choice for Tate pairing implementation in characteristic three.

3.3.1 BKLS algorithm and its modifications

Note that (Section 5.1, [5]), discarding the denominators, the recursive formula for $f_{3a}(Q)$ is obtained as –

$$f_{3a}(Q) = f_a^3(Q).g_{aP,aP}(Q).g_{2aP,aP}(Q).$$

It was further noted in [5] that it is not necessary to actually compute $2aP$, because the coefficients of $g_{2aP,aP}$ can be obtained from aP and $3aP$.

This method has been further improved upon by Duursma and Lee. They showed ([28], Theorem 5) that the Tate pairing for points $P = (\alpha, \beta)$ and $\phi_2(Q) = (\rho - x, iy)$ can be written as a product of factors of form $g = \beta y\bar{i} - (\alpha + x - \rho + b)^2$.

The Duursma-Lee algorithm is given in Table 3.2.

The Duursma-Lee algorithm (Algorithm 4, [28]) has been modified by Scott-Barreto [47] to compute $\mathrm{tr}(f)$, where the $\mathbb{F}_{q^2}$-trace of an element $f \in \mathbb{F}_{q^6}$ is $tr(f) = f + f^{q^2} + f^{q^4}$ (Definition 2, [47]). They maintain a ladder of three values $[\mathrm{tr}(f), \mathrm{tr}(f\rho), \mathrm{tr}(f\rho^2)]$. As f is initialised to 1, one can compute the initial ladder from ρ alone, which is $[0, 0, (2m^2) \bmod 3]$ (Theorem 1, [47]). Then at each iteration of the for loop, one can compute $[tr(fg), tr(fg\rho), tr(fg\rho^2)]$ using the

Table 3.3: Algorithm 3

Algorithm: **Laddering algorithm of Scott-Barreto(SB) [47] for computation of** $tr(f_P(\phi(Q)))$

input : $P = (\alpha, \beta)$ and $Q = (x, y)$
output : $tr(f_P(\phi(Q)))$
$L \leftarrow [0, 0, (2m^2) mod 3]$
for $j \leftarrow 1$ to m do
$\alpha \leftarrow \alpha^3$, $\beta \leftarrow \beta^3$
$\mu \leftarrow \alpha + x + b$
$\lambda \leftarrow \beta y\bar{i} - \mu^2$
$L \leftarrow A.L$
$x \leftarrow x^{1/3}$, $y \leftarrow y^{1/3}$
end for
return L_0

following relationship (see Theorem 2, [47]), where $\mu \equiv \alpha + x + b \in \mathbb{F}_q$ and $\lambda \equiv \beta y\bar{i} - \mu^2 \in \mathbb{F}_{q^2}$.

$$\begin{bmatrix} tr(fg) \\ tr(fg\rho) \\ tr(fg\rho^2) \end{bmatrix} = A. \begin{bmatrix} tr(f) \\ tr(f\rho) \\ tr(f\rho^2) \end{bmatrix} , \; where \; A \equiv \begin{bmatrix} \lambda & -\mu & -1 \\ -b & (\lambda - 1) & -\mu \\ -b\mu & -(\mu + b) & (\lambda - 1) \end{bmatrix}$$

and g is as in Algorithm 2.

Define $L = [tr(f), tr(f\rho), tr(f\rho^2)]^T$, then the implicit pairing is computed using the matrix A as shown in Table 3.3.

The Duursma-Lee algorithm is the most efficient algorithm to compute $f_P(\phi_2(Q))$, while the Scott-Barreto algorithm is the most efficient if one computes $tr(f_P(\phi_2(Q)))$. The Duursma-Lee algorithm in each step takes 20 base field multiplications, the laddering algorithm of Scott-Barreto takes 17 base field multiplications. However, in each of these two algorithms one has to take two cube roots in each iteration. This cube root computation for polynomial basis in characteristic three can be efficiently done by just two base field multiplication using two precomputed values – for details see [34].

3.4 Other super-singular curves over characteristic 3 fields

In what follows we shall show that the Duursma-Lee and the Scott-Baretto algorithms can be extended to the whole set of supersingular curves and further optimizations obtained.

3.4.1 Definition and important properties

From the set of all possible non-singular EC over $\mathbb{F}_3$, observe that the equation of a supersingular elliptic curve over fields of characteristic three can be expressed as –

$$E_{a,b} : y^2 = x^3 + ax + b \quad where \quad a \in \{-1, 1\}, \quad b \in \{-1, 0, 1\} \tag{3.3}$$

This gives six candidate elliptic curves of which only two (i.e. E_2) – probably because of their highest possible k – have so far got the attention for Tate-pairing implementation. The relevant properties of the other four curves $E_3(\mathbb{F}_q)$ are given in Table 3.4 (see Section 5.2.2 [41]). Note that, $E_3(\mathbb{F}_q)$ is either cyclic or contains a cyclic subgroup of order $(q+1)/2$. Let,

Table 3.4: Important properties of E_3

Curve equation	$E_3 : y^2 = x^3 + ax + b$ if $a = 1$, $b \in \{-1, 0, 1\}$; if $a = -1$, $b = 0$
trace	0
curve order	$q + 1$
distortion map	$\phi_3(x, y) = (-x - b, iy)$
condition	$i \in \mathbb{F}_{q^2}$; $i^2 = -1$

n be a multiple of r i.e $n = lr$, then from definition

$$\begin{aligned} \langle f_n \rangle &= n\langle P \rangle - \langle nP \rangle - (n-1)\langle \mathcal{O} \rangle \\ &= n\langle P \rangle - n\langle \mathcal{O} \rangle \\ &= l(r\langle P \rangle - r\langle \mathcal{O} \rangle) \\ &= l(r\langle P \rangle - \langle rP \rangle - (r-1)\langle \mathcal{O} \rangle \end{aligned}$$

So, $\langle f_n \rangle = l\langle f_r \rangle$ and $f_r(Q) = (f_n(Q))^{1/l}$. Thus, $f_r(Q)^{(q^k-1)/r} = f_n(Q)^{(q^k-1)/n}$. In case of E_2, n is taken to be $\#E_2(\mathbb{F}_q)$ and one computes $f_n(Q)^{(q^k-1)/n}$. But it suffices to take any n that is a multiple of r and divides the curve order.

One can now extend the result of Baretto *at al* to the curves E_3 to show that one can completely ignore the denominator part in Miller's Formula for the computation of Tate pairing.

Lemma 3.4.1 *For all the supersingular curves in characteristic three, the denominators in Miller's formula can be discarded altogether without affecting the final pairing value.*

In [28], it has been observed (*Remark 2.3*) for E_2 that,

$$\langle g_{aP,aP} g_{aP,-3aP} / g_{2aP} \rangle = 3\langle aP \rangle + \langle -3aP \rangle - 4\langle \mathcal{O} \rangle$$

For $aP = (\alpha, \beta)$ the function $G_{aP}(x, y) = \beta^3 y - (\alpha^3 - x + b)^2$ has the same divisor. However this result can be easily generalised as shown in Lemma 3.4.2.

Lemma 3.4.2 *For the generic supersingular elliptic curve $E_{a,b}$ over characteristic three we can replace $g_{cP,cP}.g_{cP,-3cP}/g_{2cP}$ by the equation of a parabola which is of the form $G_{cP}(x, y) = \beta^3 y - (\alpha^3 + ax + b)^2$, where $cP = (\alpha, \beta)$.*

Proof: We write,

$$\begin{aligned}\left\langle \frac{g_{cP,cP}g_{cP,-3cP}}{g_{2cP}} \right\rangle &= 2\langle cP\rangle + \langle -2cP\rangle - 3\langle \mathcal{O}\rangle + \langle cP\rangle + \langle 2cP\rangle + \langle -3cP\rangle - 3\langle \mathcal{O}\rangle \\ &\quad -\{\langle 2cP\rangle + \langle -2cP\rangle - 2\langle \mathcal{O}\rangle\} \\ &= 3\langle cP\rangle + \langle -3cP\rangle - 4\langle \mathcal{O}\rangle\}\end{aligned}$$

The rational function $G_{cP}(x, y) = \beta^3 y - (\alpha^3 + ax + b)^2$ has the same divisor.
To prove this, we solve the two equations : $y^2 = x^3 + ax + b$ and $\beta^3 y = (\alpha^3 + ax + b)^2$ simultaneously to get a zero of order three at (α, β) and another zero of order one at $(\alpha^9 + b - ab, \beta^9)$, but the later is nothing but $-3cP$.

3.5 Representation of extension field element

We now describe how to represent the extension field $\mathbb{F}_{q^k}$ elements in terms of $\mathbb{F}_q$ elements, for $k = 2$ as well as $k = 6$. This has great bearing on efficient computation of $f_P(\phi(Q))$, since $\phi(Q)$ is an element of $\mathbb{F}_{q^k}$. A similar kind of analysis can also be found in [31].

3.5.1 Case of $k = 2$:

Here the extension field is $\mathbb{F}_{q^3}$, $q^3 = 3^{3m}$ where m is odd. So, -1 is a quadratic non-residue in $\mathbb{F}_{q^3}$ and hence $i^2 + 1$ is irreducible over $\mathbb{F}_{q^3}$. Thus $B_1 = \{i, 1\}$ forms a basis of $\mathbb{F}_{q^6}$ over $\mathbb{F}_{q^3}$. A generic multiplication of two elements $(a + ib)$, $(c + id) \in \mathbb{F}_{q^6}$ with $a, b, c, d \in \mathbb{F}_{q^3}$ should take four multiplications. As $(a+ib)\times(c+id) = (ac-bd)+i(ad+bc)$, we compute ac, bd, $(a+b)(c+d)$, then $(ad+bc) = (a+b)(c+d) - ac - bd$. This requires three multiplications over $\mathbb{F}_{q^3}$.

3.5.2 Case of $k = 6$:

This is the case for E_2. Given the distortion map and the corresponding conditions for E_2 a field element of $\mathbb{F}_{q^6}$ can be represented using the basis $B_2 = \{i\rho^2, \rho^2, i\rho, \rho, i, 1\}$. The generic multiplication of two elements in $\mathbb{F}_{q^6}$ should thus take 36 base field multiplications, but can be accomplished in 18 multiplications (see A.1 in [5]). In our case we will need (see Section 3.6.1) to multiply γ, $\delta \in \mathbb{F}_{q^6}$, where $\gamma = b_5 i\rho^2 + b_4\rho^2 + b_3 i\rho + b_2\rho + b_1 i + b_0$

and $\delta = -\rho^2 + a_2\rho + a_1 i + a_0$. In the naive method it takes 18 multiplications($[M]$), but can be accomplished in only 13$[M]$, as shown below (see also the distortion map and the corresponding conditions of E_2 in Section).
Suppose $\gamma \times \delta$ is $b'_5 i\rho^2 + b'_4\rho^2 + b'_3 i\rho + b'_2\rho + b'_1 i + b'_0$. Then

$$\begin{aligned}
b'_5 &= -b_5 - b_1 + a_2b_3 + a_1b_4 + a_0b_5 \\
b'_4 &= -b_4 - b_0 + a_2b_2 - a_1b_5 + a_0b_4 \\
b'_3 &= -bb_5 - b_3 + a_2b_1 + a_1b_2 + a_0b_3 + a_2b_5 \\
b'_2 &= -bb_4 - b_2 + a_2b_0 + a_0b_2 + a_2b_4 - a_1b_3 \\
b'_1 &= a_1b_0 + a_0b_1 + ba_2b_5 - bb_3 \\
b'_0 &= -bb_2 + a_0b_0 + ba_2b_4 - a_1b_1
\end{aligned}$$

We use the following explicit multiplications –
$t_0 = a_0b_0$ $t_1 = a_1b_1$ $t_2 = a_2b_2$ $t_3 = a_0b_3$ $t_4 = a_0b_4$
$t_5 = a_1b_3$ $t_6 = a_1b_5$ $t_7 = a_2b_4$ $t_8 = a_2b_5$
$t_9 = (a_0 + a_1 + a_2)(b_3 + b_4 + b_5)$
So, $a_0b_5 + a_1b_4 + a_2b_3 = t_9 - (t_3 + t_4 + t_5 + t_6 + t_7 + t_8)$
$t_{01} = (a_0 + a_1)(b_0 + b_1)$
$a_0b_1 + a_1b_0 = t_{01} - (t_0 + t_1)$
$t_{12} = (a_1 + a_2)(b_1 + b_2)$
$a_1b_2 + a_2b_1 = t_{12} - (t_1 + t_2)$
$t_{02} = (a_0 + a_2)(b_0 + b_2)$
$a_0b_2 + a_2b_0 = t_{02} - (t_0 + t_2)$

$$\begin{aligned}
b'_5 &= -b_5 - b_1 + t_9 - (t_3 + t_4 + t_5 + t_6 + t_7 + t_8) \\
b'_4 &= -b_4 - b_0 + t_2 - t_6 + t_4 \\
b'_3 &= -bb_5 - b_3 + t_{12} - (t_1 + t_2) + t_3 + t_8 \\
b'_2 &= -bb_4 - b_2 + t_{02} - (t_0 + t_2) + t_7 - t_5 \\
b'_1 &= t_{01} - (t_0 + t_1) + bt_8 - bb_3 \\
b'_0 &= -bb_2 + t_0 + bt_7 - t_1
\end{aligned}$$

3.6 Pairing Computation

Here we discuss in details how to compute $f_P(\phi_i(Q))$ for E_i ($i \in \{2, 3\}$) i.e., all six supersingular curves in characteristic three.

3.6.1 Efficient Algorithm for E_2 and E_3

As already noted above, the existence of irrelevant denominators, efficient point tripling, coupled with the triple and add/subtract method and highest possible k make E_2 an attractive

choice for Tate pairing implementation. The above optimizations combined with the improvement suggested in [28] produce the most efficient algorithm in Tate pairing computation on E_2. All these optimizations are also valid for E_3, as is evident from our discussion in Section 3.4.1.

Case E_2 :

From Lemma 3.4.2, for E_2 we can write –

$$\begin{aligned} G_{cP}(\phi_1(Q)) &= \beta^3(iy) - (\alpha^3 + a(\rho - x) + b)^2 \\ &= -\rho^2 + a(\alpha^3 - ax + b)\rho + \beta^3 yi - (\alpha^3 - ax + b)^2 \\ &= -\rho^2 + a_2\rho + a_1 i + a_0 \quad (say) \end{aligned}$$

Note that, for E_2, $a = -1$ and $b \in \{-1, 1\}$, so $b - ax = x \pm 1$ and $a_0 = a_2^2$. So, a_i, $0 \leq i \leq 2$ can be computed using just one base field multiplication and one squaring.

Case E_3 :

Similarly, for E_3 we have,

$$\begin{aligned} G_{cP}(\phi_3(Q)) &= \beta^3(iy) - (\alpha^3 + a(-b - x) + b)^2 \\ &= a_1 i + a_0 \quad (say) \end{aligned}$$

Hence, $G_{cP}(\phi_3(Q))$ can be computed using just one multiplication and one squaring in the base field. The modified algorithm is given in Table 3.5.

Correctness of Algorithm 4 :

First note that Algorithm 4 uses the same left-to-right triple-and-add/subtract method as was proposed in [5]. The only difference being that the evaluation of $g_{cP,cP}()$ and $g_{cp,2cP}$ at $\phi_j(Q)$ in each iteration is replaced by the evaluation of the parabola $G_{cP}()$ at $\phi_j(Q)$. This one can do as long as the denominators are irrelevant, as is the case for both E_1 and E_2. In our algorithm we use three subroutines $CompG_j(Z)$, $Update_j()$ and $CombAl_j()$. Given the EC point Z, $CompG_j(Z)$ computes $G_Z(\phi_j(Q))$ together with $3Z$ and and assigns them to G and Z' respectively. In the subroutine $Update_j(f, G)$ we first cube f and then multiply f^3 with $G(\phi_j(Q))$ and store the result in f. The $CombAl_j()$ is invoked whenever there is 1 or -1 in the signed ternary expansion of n. There are precisely two such instances at $k = m$ and $k = 0$. In the first call of the $CombAl_j()$ we compute the coefficients of the line passingthrough $\pm P$ and Z and evaluate it at $\phi_j(Q)$, together with the coordinates of the point $(Z \pm P)$, which is then assigned to Z. In the second call of $CombAl_j()$ we compute the coefficients of the line passing through P and Z and evaluate it at $\phi_j(Q)$, this call in addition also computes the point $3^{2m-1}P \pm 3^m P + P$ which should be equal to $\mathcal{O}$.

Table 3.5: Algorithm 4

Algorithm: **Computation of** $f_P(\phi_j(Q))$, $j \in \{2,3\}$

input : $P = (x, y)$, $Q = (x_Q, y_Q)$ and $n = 3^{2m-1} \pm 3^m + 1$

output : $f_P(\phi_j(Q)) = f_n(\phi_j(Q))$ where $\phi_2(x, y) = (\rho - x, iy)$ and $\phi_3(x, y) = (-b - x, iy)$

set $f \leftarrow 1$, $Z \leftarrow P$ and $h \leftarrow 1$
for $k \leftarrow 2m - 2$ down to 0 do
$(G, Z') \leftarrow CompG_j(Z)$
$f \leftarrow Update_j(f, G)$
$Z \leftarrow Z'$
if $(l_k = \pm 1)$ {
$(Z', h) \leftarrow CombAl_j(Z, \pm P)$
$f \leftarrow f.h$
$Z \leftarrow Z'$
}
end for
return f

3.6.2 Computing the trace

To increase security, it has been suggested in [31] to take the trace of the pairing value. Scott-Barreto in [47] propose a method of implicit exponentiation for trace computation. Here we show how to compute the trace of the ultimate pairing value. In our comparison with the Scot-Barreto method in the next section we will show our method of first computing the pairing and then taking the trace is more efficient.

Case of E_2 :

The $\mathbb{F}_{q^2}$-trace of $f_P \in \mathbb{F}_{q^6}$ is the value $tr(f_P) = f_P + f_P^{q^2} + f_P^{q^4} \in \mathbb{F}_{q^2}$ [Definition 2, SB04]. Now, f_P will be of the form $Ai\rho^2 + B\rho^2 + Ci\rho + Di + E\rho + F$. So,

$$\begin{aligned} tr(f_P) &= tr(Ai\rho^2 + B\rho^2 + Ci\rho + Di + E\rho + F) \\ &= tr(Ai\rho^2) + tr(B\rho^2) + tr(Ci\rho) + tr(Di) + tr(E\rho) + tr(F) \\ &= A(i\rho^2 + (i\rho^2)^{q^2} + (i\rho^2)^{q^4}) + B(\rho^2 + (\rho^2)^{q^2} + (\rho^2)^{q^4}) \\ &\quad + C(i\rho + (i\rho)^{q^2} + (i\rho)^{q^4}) + D(\rho + (\rho)^{q^2} + (\rho)^{q^4}) \\ &\quad E(i + (i)^{q^2} + (i)^{q^4}) + (F + F^{q^2} + F^{q^4}) \end{aligned}$$

Now, $q = 3^m$, where m is odd. So, $i^{q^2} = i^{q^4} = i$ and, $i + i^{q^2} + i^{q^4} = 0$
From Theorem 1 of [47] $tr(\rho) = \rho + \rho^{q^2} + \rho^{q^4} = 0$ and, $tr(\rho^2) = \rho^2 + (\rho^2)^{q^2} + (\rho^2)^{q^4} = -1$
Using these results we get, $tr(f_P) = -(B + iA) \in \mathbb{F}_{q^2}$. This can be obtained without any extra computation.

Case of E_3 :

The $\mathbb{F}_{q^3}$ trace of $f \in \mathbb{F}_{q^6}$ is defined as $f + f^{q^3}$. Now, f is represented as $\alpha + i\beta$, $\alpha, \beta \in \mathbb{F}_{q^3}$. Since $i^q = (i^2)^{(q-1)/2} i = -i$, we get $i^{q^3} = -i$ and $tr(f) = -\alpha \in \mathbb{F}_{q^3}$. Note that, the size of trace for E_3 is more than that for E_2.

Bibliography

[1] S. Al-Riyami and K. G. Paterson. *Tripartite Authenticated Key Agreement Protocols from Pairings.* In proceedings of IMA Conference of Cryptography and Coding, LNCS 2898, pp. 332-359. Also available at http://eprint.iacr.org/2002/035.

[2] G. Ateniese, M. Steiner, and G. Tsudik. *Authenticated Group Key Agreement and Friends.* In proceedings of ACM CCS 1998[1], pp. 17-26, ACM Press, 1998.

[3] J. Baek, Y. Zheng. *Identity-Based Threshold Decryption.* In proceedings of PKC 2004, LNCS 2947, pp. 262-276, Springer-Verlag, 2004. Also available at http://eprint.iacr.org/2003/164.

[4] P. S. L. M. Barreto. *Pairing Based Crypto Lounge.* Available at http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html

[5] P. S. L. M. Barreto, H. Y. Kim, B. Lynn and M. Scott. *Efficient Algorithms for Pairing Based Cryptosystems.* In proceedings of Crypto 2002, LNCS 2442, pp. 354-368, Springer-Verlag, 2002. Also available at http://www.iacr.org/2002/008.

[6] R. Barua, R. Dutta, P. Sarkar. *Extending Joux Protocol to Multi Party Key Agreement.* In proceedings of Indocrypt 2003, LNCS 2904, pp. 205-217, Springer-Verlag, 2003. Also available at http://eprint.iacr.org/2003/062.

[7] M. Bellare, R. Canetti, and H. Krawczyk. *A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols.* In proceedings of the 30th Annual Symposium on the Theory of Computing, pp. 419-428. ACM Press, 1998. Also available at http://www.cs.edu/users/mihir/papers/key-distribution.html/.

[8] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. *Relations Among Notions of Security for Public-key Encryption Schemes.* In proceedings of Crypto 1998, LNCS 1462, pp. 26-45, Springer-Verlag, 1998.

[9] M. Bellare and P. Rogaway. *Random Oracles are Practical : A Paradigm for Designing Efficient Protocols.* In proceedings of ACM CCS 1993, pp. 62-73, ACM Press, 1993.

[10] S. Blake-Wilson and A. Menezes. *Security Proofs for Entity Authentication and Authenticated Key Transport Protocols Employing Asymmetric Techniques.* In proceedings of the 5th International Workshop on Security Protocols, LNCS 1361, pp. 137-158, Springer-Verlag, 1997.

[11] A. Boldyreva. *Efficient Threshold Signature, Multi-signature and Blind Signature Schemes Based on the Gap-Diffie-Hellman-Group Signature Scheme.* In proceedings of PKC 2003, LNCS 2139, pp. 31-46, Springer-Verlag, 2003. Also available at http://www.iacr.org/2002/118.

[12] A. Boldyreva, A. Palacio and B. Warinschi *Secure Proxy Signature Schemes for Delegation of Signing Rights* Available at http://www.iacr.org/2003/096.

[13] D. Boneh, X. Boyen. *Short Signatures Without Random Oracles.* In proceedings of Eurocrypt 2004, LNCS 3027, pp. 56-73, Springer-Verlag, 2004.

[14] D. Boneh, X. Boyen. *Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles.* In proceedings of Eurocrypt 2004, LNCS 3027, pp. 223-238, Springer-Verlag, 2004.

[15] D. Boneh, X. Boyen. *Secure Identity-Based Encryption Without Random Oracles.* In proceedings of Crypto 2004, LNCS 3152, pp. 443-459, Springer-Verlag, 2004.

[16] D. Boneh, G. D. Crescenzo, R. Ostrovsky, G. Persiano. *Public Key Encryption with Keyword Search.* In proceedings of Eurocrypt 2004, LNCS 3027, pp. 506-522, Springer-Verlag, 2004. Also available at http://eprint.iacr.org/2003/195.

[17] D. Boneh and M. Franklin. *Identity-Based Encryption from Weil Pairing.* In proceedings of Crypto 2001, LNCS 2139, pages 213-229, Springer-Verlag, 2001.

[18] D. Boneh, C. Gentry, B. Lynn and H. Shacham. *Aggregate and Verifiably Encrypted Signature from Bilinear Maps.* In proceedings of Eurocrypt 2003, LNCS 2248, pp. 514-532, Springer-Verlag, 2003.

[19] D. Boneh, B. Lynn, and H. Shacham. *Short Signature from Weil Pairing.* In proceedings of Asiacrypt 2001, LNCS 2248, pp. 213-229, Springer-Verlag, 2001.

[20] E.Bresson, O. Chevassut, and D. Pointcheval. *Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions.* In proceedings of Eurocrypt 2002, LNCS 2332, pp. 321-336, Springer-Verlag, 2002.

[21] E. Bresson, O. Chevassut, and D. Pointcheval. *Provably Authenticated Group Diffie-Hellman Key Exchange - The Dynamic Case.* In proceedings of Asiacrypt 2001, LNCS 2248, pp. 290-309, Springer-Verlag, 2001.

[22] E. Bresson, O. Chevassut, D. Pointcheval, and J. J. Quisquater. *Provably Authenticated Group Diffie-Hellman Key Exchange.* In proceedings of ACM CCS 2001, pp. 255-264, ACM Press, 2001.

[23] S. Chatterjee, P. Sarkar. *Trading Time for Space: Towards an Efficient IBE Scheme with Shorter Public Parameters in the Standard Model.* In proceedings of ICISC 2005, to appear.

[24] S. Chatterjee, P. Sarkar, R Barua. *Efficient Computation of Tate Pairing in Projective Coordinates Over General Characteristic Fields.* In proceedings of ICISC 2004, pp. 168-181, LNCS .

[25] X. Chen, F. Zhang, K. Kim. *A New ID-based Group Signature Scheme from Bilinear Pairings.* In proceedings of WISA 2003, pp. 585-592, August 2land (KR). Also available at http://eprint.iacr.org/2003/116.

[26] Y. Dodis, L. Reyzin. *Breaking and Repairing Optimistic Fair Exchange from PODC 2003.* In proceedings of ACM Workshop on Digital Rights Management 2003, pp. 47-54, ACM Press.

[27] R. Dutta, R. Barua and P. Sarkar. *Pairing Based Cryptographic Protocols : A Survey.* Manuscript 2004, submitted. Available at http://eprint.iacr.org/2004/064.

[28] I. Duursma, H.S. Lee. *Tate Pairing Implementation for Hyperelliptic Curves* $y^2 = x^p - x + d$. In Advances in Cryptology-Asiacrypt 2003, LNCS 2894, pp. 111-123, Springer-Verlag, 2003.

[29] G. Frey, H. Ruck. *A Remark Concerning m-divisibility and The Discrete Logarithm in the Divisor Class Group of Curves.* In Mathematics of Computation, 62, pp. 865-874, 1994.

[30] E. Fujisaki, T. Okamoto. *Secure Integration of Asymmetric and Symmetric Encryption Schemes.* In proceedings of Crypto 1999, LNCS 1666, pp. 577-554, Springer-Verlag, 1999.

[31] S. Galbraith, K. Harrison and D. Soldera. *Implementing the Tate Pairing.* In proceedings of Algorithm Number Theory Symposium - ANTS V, LNCS 2369, pp. 324-337, Springer-Verlag, 2002.

[32] C. Gentry and A. Silverberg. *Hierarchical ID-based Cryptography.* In proceedings of Asiacrypt 2002, LNCS 2501, pp. 548-566, Springer-Verlag, 2002.

[33] S. Goldwasser, S. Micali and R. Rivest. *A Digital Signature Scheme Secure against Adaptive Chosen -Message Attacks.* In SIAM Journal of Computing, 17(2), pp. 281-308, April 1998.

[34] R. Granger, D. Page, M. Stam *Hardware and Software Normal Basis Arithmetic for Pairing- based Cryptography in Characteristic Three.* Available at http://eprint.iacr.org/2004/157.

[35] D.R. Hankerson, A.J. Menezes and S.A. Vanstone *Guide to Elliptic Curve Cryptography.* Springer-Verlag, New York 2004.

[36] F. Hess. *Efficient Identity Based Signature Schemes Based on Pairings.* In proceedings of SAC 2002, LNCS 2595, pp. 310-324, Springer-Verlag, 2002.

[37] A. Joux. *A One Round Protocol for Tripartite Diffie-Hellman.* In proceedings of ANTS 4, LNCS 1838, pp. 385-394, Springer-Verlag, 2000.

[38] M. Kim, K. Kim. *A New Identification Scheme Based on the Gap Diffie-Hellman Problem.* In SCIS 2002, 1/2, pp. 349-352, Shirahama, Japan, 2003.

[39] M. Kim, K. Kim. *A New Identification Scheme Based on the Bilinear Diffie-Hellman Problem.* In proceedings of ACISP 2002, LNCS 2384, pp. 368-378, Springer-Verlag, 2002.

[40] J. Malone-Lee. *Identity-Based Signcryption.* Available at http://eprint.iacr.org/2002/098.

[41] A. Menezes. *Elliptic Curve Public Key Cryptosystems.* Kluer Academic Publishers, 1993.

[42] A. Menezes, T. Okamoto, and S. Vanstone. *Reducing Elliptic Curve Logarithms to Logarithms in a finite field.* In IEEE Transaction on Information Theory, 39, pp. 1639-1646, 1993.

[43] A. Menezes, P. C. Van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, 1997. Also available at http://cacr.math.uwaterloo.ca/hac.

[44] V.S. Miller. *Short Programs for Functions on Curves.* Journal of Cryptology, Vol 17, No. 4, 235-262, 2004

[45] C. Racoff and D. Simon. *Noninteractive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack.* In proceedings of Crypto 1991, LNCS 576, pp. 433-444, Springer-Verlag, 1991.

[46] R. Sakai, K. Ohgishi and M. Kasahara. *Cryptosystems Based on Pairing.* In SCIS 2000-c20, Okinawa, Japan, January 2000.

[47] M. Scott, P.L.S.M. Baretto. *Compressed Pairings.* In proceedings of Crypto 2004, LNCS 3152, pp 140-156, Springer-Verlag, 2004.

[48] A. Shamir. *Identity-based Cryptosystems and Signature Schemes.* In proceedings of Crypto 1984, LNCS 196, pp. 47-53, Springer-Verlag, 1984.

[49] B. Waters. *Efficient Identity-Based Encryption Without Random Oracles.* In proceedings of Eurocrypt 2005, LNCS 3494, pp. 114-127, Springer-Verlag, 2005.

[50] F. Zhang, R. Safavi-Naini and W. Susilo. *An Efficient Signature Scheme from Bilinear Pairings and its Applications.* In proceedings of PKC 2004, LNCS 2947, pp. 277-290, Springer-Verlag, 2004.

[51] F. Zhang, R. Safavi-Naini and W. Susilo. *ID-based Chameleon Hashes from Bilinear Pairings. Available at http://eprint.iacr.org/2003/208*