ID-based cryptography from bilinear pairings

Tatiana Issaeva

Draft, 11 February 2003

1 Introduction

Identity-based systems are public-key encryption schemes, but users do not have explicit public keys. Instead, public key is derived, in a publicly known way, from a user's publicly available identity. For example, user's name or e-mail address can be used to obtain public key strings.

Originally the identity-based encryption (IBE) scheme was proposed in 1984 by Shamir [7] with idea to simplify certificate management in e-mail systems by avoiding the explicit authentification of public keys by means of public key certificates. The motivaton behind developing identity-based encryption system is to stimulate the deployment of a public key infrastructure. By using identity-based encryption scheme, there is no need for users to exchange their public keys, but the services of trusted third party (TTP) is required during set-up phase to generate system parameters and to compute user's private key. From a user's identity, which is publicly known and in a standardized form, a TTP computes the corresponding private key and securely transmits it to user. The TTP in this case must be unconditionaly trusted.

Since the idea emerged in 1984, there was no satisfactory solution to it, despite several proposals [3,6,9,8]. The problem of constructing a usable IBE was considered an open issue until recently, when a fully fnctional IBE scheme was proposed in [1]. Surprisingly, the satisfactoty solutions existed to related notions of identity-based signatures, for example in [4].

The IBE scheme from [1] is constructed from any biliner map $e : G_1 \times G_1 \to G_2$ between two groups G_1 and G_2, such that a variant of Computational Deffie-Hellman (CDH) problem is hard in G_1. The performance of IBE scheme proposed in [1] is comparable to the performance of ElGamal encryption in F_p^*. The IBE scheme from [1] has chosen ciphertext security in the random oracle model. As a concrete example of a bilinear pairing, which can be used for IBE construction, the Weil pairings on elliptic curves are proposed in [1].

This survey is organized in a following way. First, all required definitions for Bilinear Diffie-Hellman Assumption (BDH) are given in Section 2. Then, in Section 3, some of the formal definitions for security required from general IBE systems are presented. In Section 4 we will look at the basic IBE scheme from bilinear pairings and define and prove its security. Also we will briefly look at the full IBE scheme, which is derived from basic IBE scheme and which is secure from chosen ciphertext attacks. Section 5 will provide some conclusions to this survey.

2 The Bilinear Diffie-Hellman Assumption.

The security of the IBE system, which presented in this survey is based on Bilinear Diffie-Hellman Assumption (BDH). In this section we will give all the needed definitions for BDH. Let G_1 and G_2 be two groups of prime order q. Let $e : G_1 \times G_1 \to G_2$be an admissible bilinear map and let P be a generator of G_1.

The BDH problem in $\langle G_1, G_2, e\rangle$ is given $\langle P, aP, bP, cP\rangle$ compute $e = (P, P)^{abc}$ for some $a, b, c \in \mathbb{Z}_p^*$.

Definition 1. A randomized algorithm $\mathcal{G}$ is a *BDH parameter generator* if (1) $\mathcal{G}$ takes a security parameter $k \in \mathbb{Z}^+$, (2) $\mathcal{G}$ runs in polynomial time in k, and (3) $\mathcal{G}$ outputs a prime number q, the description of two groups G_1, G_2 of order q, and the description of admissible bilinear map $e : G_1 \times G_1 \to G_2$.

By $\mathcal{G}\left(1^k\right) = \langle q, G_1, G_2, e\rangle$ we will denote the out-

put of $\mathcal{G}$.

We will make following assumptions. The description of each group contains group generator and algorithm, which is polynomial in k time, for computing the group operations. The generator of group enables us to generate uniformly random elements in this group. We will also asume that desscription of e contains a polynomial time algorithm to compute e.

Definition 2. An algorithm $\mathcal{A}$ has an advantage $\epsilon(k)$ in solving the BDH problem for $\mathcal{G}$ if for sufficiently large k, $\mathsf{adv}_{\mathcal{G},\mathcal{A}}(k) \geq \epsilon(k)$, where $\mathsf{adv}_{\mathcal{G},\mathcal{A}}(k) =$

$\mathsf{Pr}\left[\mathcal{A}(P, aP, bP, cP) = e(P,P)^{abc} \middle| a, b, c \leftarrow \mathbb{Z}_q^*\right]$

Is is said that $\mathcal{G}$ satisfies the BDH assumption if $\mathsf{adv}_{\mathcal{G},\mathcal{A}}(k) < 1/f(k)$ for any randomized polynomial time algorithm $\mathcal{A}$, for any polynomial $f \in \mathbb{Z}[x]$ and for sifficiently large k. When $\mathcal{G}$ satisfies the BDH assumption, it is said that BDH is hard in groups generated by $\mathcal{G}$.

3 Identity-Based encryption security definitions.

An identity-based encryption scheme $\mathcal{E}$ is specified by four randomized algorithms: Setup, Extract, Encrypt, Decrypt. Setup generates global system parametes params and a $\mathsf{master-key}$. Extract uses the $\mathsf{master-key}$ and params to generate the private key for an arbitrary public key string $\mathsf{ID} \in \{0,1\}^*$. Encrypt encrypts messages using public key ID and system parameters params. Decrypt decrypts messages using corresponding private key and system parameters params.

3.1 Chosen ciphertext security for IBE.

It is said that an IBE scheme $\mathcal{E}$ is semantically secure against an adaptive chosen ciphertext attack if no polynomially bounded adversary $\mathcal{A}$ has a non-negligible advantage against the Challenger in the following $\mathsf{IND-ID-CCA}$ game:

Setup: The challenger takes a security paramer k and runs the Setup algorithm. It gives the adversary the resulting system parameters params. It keeps the $\mathsf{master-key}$ to itself.

Phase 1: The adversary issues queries $q_1, \ldots, q_m$ where query q_i is one of:

- Extraction query $\langle \mathsf{ID}_i \rangle$. The challenger responds by running algorith Extract to generate the private key d_i corresponding to the public key ID_i. It sends d_i to the adversary.

- Decryption query $\langle \mathsf{ID}_i, C_i \rangle$. The challenger responds by running algorithm Extract to generate the private key d_i corresponding to the public key ID_i. It then runs algorithm Decrypt to decrypt the ciphertext C_i using the private key d_i. It sends 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 plaintexts $M_0, M_1 \in \mathcal{M}$ and an identity ID on which it wishes to be challenged. The only constraint is that ID did not appear in any private key extraction quaries in Phase 1. The challenger pick a random bit $b \in \{0,1\}$ and sets $C = \mathsf{Encrypt}(\mathsf{params}, \mathsf{ID}, M_b)$. It sends C as challenge to the adversary.

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

- Extraction query $\langle \mathsf{ID}_i \rangle$, where $\mathsf{ID}_i \neq \mathsf{ID}$. The challenger responds as in Phase 1.

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

These queries mabe asked adaptively as in Phase 1.

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

We refer to such an adversary $\mathcal{A}$ as $\mathsf{IND-ID-CCA}$ adversary.

Definition 3. The advantage of $\mathsf{IND-ID-CCA}$ adversary $\mathcal{A}$ against the scheme $\mathcal{E}$ is:

$\mathsf{adv}_{\mathcal{E},\mathcal{A}}(k) = \left|\mathsf{Pr}\left[b = b'\right] - \frac{1}{2}\right|$. The probability if over the random bits used by the challenger and the adversary.

Definition 4. We say that the IBE scheme $\mathcal{E}$ is semantically secure against an adaptive chosen ciphertext attack if for any polynomial time $\mathsf{IND-ID-CCA}$ adversary $\mathcal{A}$ the function $\mathsf{adv}_{\mathcal{E},\mathcal{A}}(k)$ is negligible. As shorthand, we say that $\mathcal{E}$ is $\mathsf{IND-ID-CCA}$ secure.

3.2 Chosen plaintext security for IBE.

Chosen plaintext security is similar to chosen ciphertext security except that the adversary issues only extraction queries $\langle \mathsf{ID}_i \rangle$ in Phase 1 an in Phase 2 of $\mathsf{IND}-\mathsf{ID}-\mathsf{CCA}$ algorithm. We will refer to such a variation of $\mathsf{IND}-\mathsf{ID}-\mathsf{CCA}$ adversary $\mathcal{A}$ as $\mathsf{IND}-\mathsf{ID}-\mathsf{CPA}$ adversary.

Definition 5. The advantage of $\mathsf{IND}-\mathsf{ID}-\mathsf{CPA}$ adversary $\mathcal{A}$ against the scheme $\mathcal{E}$ is:

$\mathsf{adv}_{\mathcal{E},\mathcal{A}}(k) = \left|\mathsf{Pr}\left[b = b'\right] - \frac{1}{2}\right|$. The probability is over the random bits used by the challenger and the adversary.

Definition 6. We say that the IBE scheme $\mathcal{E}$ is semantically secure if for any polynomial time $\mathsf{IND}-\mathsf{ID}-\mathsf{CPA}$ adversary $\mathcal{A}$ the function $\mathsf{adv}_{\mathcal{E},\mathcal{A}}(k)$ is negligible. As shorthand, we say that $\mathcal{E}$ is $\mathsf{IND}-\mathsf{ID}-\mathsf{CPA}$ secure.

4 Identity-Based encryption based on pairings.

In this section of the survey we will present IBE scheme from [1]. The description will be presented in two stages to simplify presentation. On the first stage, we will present basic IBE scheme, which is not secure against an adaptive chosen ciphertext attack. On the second stage we will extend basic IBE to full IBE scheme to get security against an adaptive chosen ciphertext attack in the random oracle model. Here we will refer the reader to the previous surveys in the seminar for reliability of random oracle model.

4.1 Basic IBE scheme BasicIdent and its security definition

Let $\mathcal{G}$ be some BDH parameter generator satisfying the BDH assumption. The basic IBE scheme BasicIdent consists of four algorithms: Setup, Extract, Encrypt, Decrypt.

Setup:

Step 1: Run $\mathcal{G}$ on input $k \in \mathbb{Z}^+$ to generate $\mathcal{G}\left(1^k\right) = \langle q, G_1, G_2, e \rangle$. Choose generator $P \in G_1$.

Step 2: Choose random $s \in \mathbb{Z}_q^*$ and set $P_{pub} = sP$.

Step 3: Choose cryptographic hash functions $H_1 : \{0,1\}^* \to G_1^*$ and $H_2 : G_2 \to \{0,1\}^n$ for some n.

The system parameters are $\mathsf{params} = \langle q, G_1, G_2, e, n, P, P_{pub}, H_1, H_2 \rangle$. The $\mathsf{master-key}$ is $s \in \mathbb{Z}_q^*$.

Extract (key-generation): For a given $\mathsf{ID} \in \{0,1\}^*$ compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in G_1^*$ and set private key $d_{\mathsf{ID}} = sQ_{\mathsf{ID}}$.

Encrypt: To encrypt message $m \in \{0,1\}^n$

(1) compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID})$, (2) compute $g_{\mathsf{ID}} = e(Q_{\mathsf{ID}}, P_{pub})$, (3) choose random $r \in \mathbb{Z}_q^*$, (4) compute $U = rP$ and $V = m \oplus H_2(g_{\mathsf{ID}}^r)$. (5) Set cipher text to be $C = \langle U, V \rangle$.

Decrypt: To decrypt C compute $m = V \oplus H_2(e(d_{\mathsf{ID}}, U)$.

4.2 Security proof of BasicIdent

In this section we will prove that this scheme is semantically secure IBE scheme ($\mathsf{IND}-\mathsf{ID}-\mathsf{CPA}$). For the definition of $\mathsf{IND}-\mathsf{ID}-\mathsf{CPA}$ security, please refer to Section 3.

Theorem 1: *Suppose the hash functions H_1,H_2 are random oracles. Then* BasicIdent *is* $\mathsf{IND}-\mathsf{ID}-\mathsf{CPA}$ *secure assuming BDH is hard in groups generated by $\mathcal{G}$. Concretely, suppose there is an adversary $\mathcal{A}$ that has advantage $\epsilon(k)$ against scheme* BasicIdent. *Suppose $\mathcal{A}$ makes at most $q_E > 0$ private key extraction queries and $q_{H_2} > 0$ hash queries to H_2. Then there is algorithm $\mathcal{B}$ that solves BDH in groups generated by $\mathcal{G}$ with advantage at least:*

$\mathsf{Adv}_{\mathcal{G},\mathcal{B}}(k) \geq \frac{2\epsilon(k)}{e(1+q_E)q_{H_2}}$,

where e is a base of natural logarithm.

The running time of $\mathcal{B}$ is $\mathcal{O}(time(\mathcal{A}))$.

The proof of theorem 1 is done in two stages. First, we build the public key encryption scheme BasicPub and show that $\mathsf{IND}-\mathsf{ID}-\mathsf{CPA}$ attack on BasicIdent can be converted to $\mathsf{IND}-\mathsf{CPA}$ attack on BasicPub, that is private key extaction queries do not help the adversary. In second stage, we show that BasicPub is $\mathsf{IND}-\mathsf{CPA}$ secure if BDH assumption holds.

Stage 1. We can convert BasicIdent scheme to public key encryption scheme BasicPub in following way. We define keygen algorithm for BasicPub that produces the public key $\langle q, G_1, G_2, e, n, P, P_{pub}, Q_{\mathsf{ID}}, H_2 \rangle$, where all the parameters are the same as in Setup algorithm from

BasicIdent, except Q_{ID}, which is randomly picked element from G_1^* during keygen. The encrypt and decrypt algorithms are the same as in BasicIdent, with only exception that Q_{ID} is beforehand provided to us by keygen algorithm and not computed at encryption stage by applying some random hash funtion to some identity string. Now we will prove the following lemma.

Lemma 1. *Let H_1 be a random oracle from $\{0,1\}^*$to G_1^*. Let $\mathcal{A}$ be an $\mathsf{IND}-\mathsf{ID}-\mathsf{CPA}$ adversary that has advantage $\epsilon(k)$ against* BasicIdent. *Suppose $\mathcal{A}$ makes at most $q_E > 0$ private key extraction queries. Then there is a $\mathsf{IND}-\mathsf{CPA}$ adversary $\mathcal{B}$ that has advantage at least $\frac{\epsilon(k)}{e(1+q_E)}$ against* BasicPub. *Its running time is $O(time(\mathcal{A}))$.*

Proof of lemma 1. The game starts with challenger generating public key $K_{pub} = \langle q, G_1, G_2, e, n, P, P_{pub}, Q_{\mathsf{ID}}, H_2\rangle$ and private key $d_{\mathsf{ID}} = sQ_{\mathsf{ID}}$. The K_{pub} is given to algorithm $\mathcal{B}$, which produces two messages m_0 and m_1 and expects to receive the BasicPub encryption of m_b under K_{pub}, where $b \in \{0,1\}$. Then $\mathcal{B}$ guesses $b^{'} \in \{0,1\}$ for b.

Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ game $\mathsf{IND}-\mathsf{ID}-\mathsf{CPA}$ as follows:

Setup: Algorithm $\mathcal{B}$ gives $\mathcal{A}$ the BasicIdent system parameters $\langle q, G_1, G_2, e, n, P, P_{pub}, H_1, H_2\rangle$, where H_1 is a random oracle controlled by $\mathcal{B}$ and the rest parameters are from K_{pub} .

H_1-queries: $\mathcal{A}$ can query H_1 at any time. To respond to these queries $\mathcal{B}$ maintains a list H_1^{list} of tuples $\langle \mathsf{ID}_j, Q_j, b_j, c_j\rangle$, which is initially empty. When $\mathcal{A}$ queries H_1 at a point ID_i:

1. If ID_i is in H_1^{list} in a tuple $\langle \mathsf{ID}_i, Q_i, b_i, c_i\rangle$, then $\mathcal{B}$ responds with $H_1(I\mathsf{D}_i) = Q_i \in G_1$.

2. If ID_i is not in H_1^{list}, $\mathcal{B}$ generates random *coin* $\in \{0,1\}$ so that $\mathsf{Pr}[coin = 0] = \delta$ for some δ.

3. $\mathcal{B}$ picks a random $b \in \mathbb{Z}_q^*$. If coin $= 0$, compute $Q_i = bP$, else compute $Q_i = bQ_{\mathsf{ID}}$.

4. $\mathcal{B}$ adds a tuple $\langle \mathsf{ID}_i, Q_i, b_i, c_i\rangle$ to H_1^{list} and responds to $\mathcal{A}$ with $Q_i = H_1(\mathsf{ID}_i)$. Note that Q_i is uniform in G_1^* and is independent from $\mathcal{A}$'s current view.

Phase 1. Let ID_i be a private key extraction query issued by $\mathcal{A}$. Then $\mathcal{B}$ does following:

1. Run algorithm above for H_1-queries to compute $Q_i = H_1(\mathsf{ID}_i)$. Let $\langle \mathsf{ID}_i, Q_i, b_i, coin_i\rangle$ be a coresponding tuple in H_1^{list}. If *coin* $= 1$, then attack on BasicPub failed, $\mathcal{B}$ reports failure and terminates.

2. If *coin* $= 0$, hence $Q_i = b_iP$. Define $d_i = b_iP_{pub} \in G_1^*$. Since $d_i = sQ_i$, d_i is a private key assosiated with public ID_i. $\mathcal{B}$ provides d_i to $\mathcal{A}$.

Challenge: When $\mathcal{A}$ decides that Phase 1 is over, it outputs public key ID_{ch} and two messages m_0 and m_1 on which it wished to be challenged. $\mathcal{B}$ responds as follows:

1. $\mathcal{B}$ gives m_0 and m_1 to challenger as the messages that it wishes to be challenged on. The challenger responds with a BasicPub ciphertext $C = \langle U, V\rangle$, such that C is encryption of m_c under K_{pub} for random $c \in \{0,1\}$.

2. Then $\mathcal{B}$ runs algorithm above for H_1-queries to compute $Q \in G_1^*$ such that $H_1(\mathsf{ID}_{ch}) = Q$. Let $\langle \mathsf{ID}_{ch}, Q, b, coin\rangle$ be a coresponding tuple in H_1^{list}. If *coin* $= 0$, then attack on BasicPub failed, $\mathcal{B}$ reports failure and terminates.

3. If *coin* $= 1$, hence $Q = bQ_{\mathsf{ID}}$. When $C = \langle U, V\rangle$, then $U \in G_1^*$. Set $C^{'} = \langle b^{-1}U, V\rangle$, where $b^{-1}$is inverse of b mod q.

Note that $C^{'}$ is a BasicIdent encryption of m_c under ID_{ch}, since $H_1(\mathsf{ID}_{ch}) = Q$ and private key corresponding to ID_{ch} is $d_{ch} = sQ$. Also, note that $e(b^{-1}U, d_{ch}) = e(b^{-1}U, sQ) = e(U, sb^{-1}Q) = e(U, sQ_{\mathsf{ID}}) = e(U, d_{\mathsf{ID}})$.

From latter follows that that the BasicIdent decryption of $C^{'}$ using d_{ch} is the same as the BasicPub decryption of C using d_{ID}.

Phase 2: $\mathcal{B}$ responds as in Phase 1 to private key extraction query issued by $\mathcal{A}$.

Guess: Eventually $\mathcal{A}$ produces a guess $c^{'}$ for c. Algorithm $\mathcal{B}$ outputs $c^{'}$ as its guess for c.

Now we can claim that if $\mathcal{B}$ does not abort during the simulation then $\mathcal{A}$'s view is identical to its view of real attack, since all responses to H_1 are uniformly distributed in $G_1^*$and are valid. Since also the challenge $C^{'}$ given to $\mathcal{A}$ is the BasicIdent encryption of m_c for random $c \in \{0,1\}$, hence probability that $\mathcal{B}$ returns the right answer at guess step is $\left|\mathsf{Pr}\left[c = c^{'}\right] - \frac{1}{2}\right| \geq \epsilon$ as it follows from definition of $\mathcal{A}$.

Now, we calculate the probability that $\mathcal{B}$ does not abort during the simulations. Such probability equals to $\delta^{q_E}(1-\delta)$, where δ^{q_E} is the probability that $\mathcal{B}$ does not abort in phases 1 and 2, and $(1-\delta)$ is probability that $\mathcal{B}$ does not abort at challenge step. The value $\delta^{q_E}(1-\delta)$ maximized at $\delta = 1 - 1/(q_E+1)$. Hence probability that $\mathcal{B}$ does not abort

is at least $1/e(q_E+1)$. So $\mathcal{B}$'s advantage is at least $\epsilon/e(q_E+1)$, as it was required to prove in Lemma 1. $\square$

Now we completed Stage 1 of proof of theorem 1. In next stage we will prove that BasicPub is $\mathsf{IND}-\mathsf{CPA}$ secure if BDH assumption holds by proving following lemma.

Lemma 2. *Let H_2 a random oracle from G_2 to $\{0,1\}^n$. Let $\mathcal{A}$ be an $\mathsf{IND}-\mathsf{CPA}$ adversary that has advantage $\epsilon(k)$ against BasicPub. Suppose $\mathcal{A}$ makes a total $q_{H_2} > 0$ queries to H_2. Then there is an algorithm $\mathcal{B}$ that solves BDH problem for $\mathcal{G}$ with advantage at least $\frac{2\epsilon(k)}{q_{H_2}}$ and the running time is $O(time(\mathcal{A}))$.*

Proof of lemma 2. Algorithm $\mathcal{B}$ is given as input the BDH parameters $\langle q, G_1, G_2, e\rangle$ and a random instance $\langle P, aP, bP, cP\rangle = \langle P, P_1, P_2, P_3\rangle$ of the BDH problem for random $P \in G_1^*$ and for random $a, b, c \in \mathbb{Z}_q^*$. Let $D = (P,P)^{abc}$ is the solution to the BDH problem. Algorithm $\mathcal{B}$ can find D by interaction with $\mathcal{A}$ as below:

Setup: Algorithm $\mathcal{B}$ creates the BasicPub $K_{pub} = \langle q, G_1, G_2, e, n, P, P_{pub}, Q_{\mathsf{ID}}, H_2\rangle$ so that $P_{pub} = P_1$ and $Q_{\mathsf{ID}} = P_2$, and gives $\mathcal{A}$ the K_{pub}. Note that the unknown private key associated with K_{pub} is $d_{\mathsf{ID}} = aQ_{\mathsf{ID}} = abP$.

H_2-queries: $\mathcal{A}$ can query H_2 at any time. To respond to these queries $\mathcal{B}$ maintains a list H_2^{list} of tuples $\langle X_j, H_j\rangle$, which is initially empty. When $\mathcal{A}$ queries H_2 at a point X_i:

1. If X_i is in H_2^{list} in a tuple $\langle X_i, H_i\rangle$, then $\mathcal{B}$ responds with $H_2(X_i) = H_i$.

2. If X_i is not in H_2^{list}, $\mathcal{B}$ generates a random string $H_i \in \{0,1\}^n$ and adds a tuple $\langle X_i, H_i\rangle$ to the H_2^{list}. Then $\mathcal{B}$ responds to $\mathcal{A}$ with $H_2(X_i) = H_i$.

Challenge: Algorithm $\mathcal{A}$ outputs two messages m_0 and m_1 on which it wishes to be challenged. Algorithm $\mathcal{B}$ picks a random string $R \in \{0,1\}^n$, defines C to be the ciphertext $C = \langle P_3, R\rangle$ and gives it as a challenge to $\mathcal{A}$. Note that by definition $C = R \oplus H_2(e(P_3, d_{\mathsf{ID}})) = R \oplus H_2(D)$.

Guess: $\mathcal{A}$ produces a guess $c' \in \{0,1\}$. At this point $\mathcal{B}$ picks a random tuple $\langle X_j, H_j\rangle$ from H_2^{list} and outputs X_j as the solution to given instance of BDH.

Now, we calculate the probability that $\mathcal{B}$ outputs correct answer D to BDH. For this, we define event $\mathcal{H}$ to be the event that $\mathcal{A}$ issues a query for $H_2(D)$ at some point during the simulation. We claim that $\mathsf{Pr}[\mathcal{H}]$ in simulation is the same as in real attack. We will leave the proof of this claim out of this survey and refer interested reader to [1]. By definition of $\mathcal{A}$, we know that in real attack $\left|\mathsf{Pr}\left[c = c'\right] - \frac{1}{2}\right| \geq \epsilon$. From here, we get

$$\mathsf{Pr}\left[c = c'\right] = \mathsf{Pr}\left[\left|c = c'\right| \neg\mathcal{H}\right]\mathsf{Pr}[\neg\mathcal{H}] +$$
$$+\mathsf{Pr}\left[\left|c = c'\right| \mathcal{H}\right]\mathsf{Pr}[\mathcal{H}] \leq$$
$$\leq \mathsf{Pr}\left[\left|c = c'\right| \neg\mathcal{H}\right]\mathsf{Pr}[\neg\mathcal{H}] + \mathsf{Pr}[\mathcal{H}] \leq$$
$$\leq \tfrac{1}{2}\mathsf{Pr}[\neg\mathcal{H}] + \mathsf{Pr}[\mathcal{H}] \leq \tfrac{1}{2} + \tfrac{1}{2}\mathsf{Pr}[\mathcal{H}]$$

From latter,

$\mathsf{Pr}\left[c = c'\right] \geq \mathsf{Pr}\left[\left|c = c'\right| \neg\mathcal{H}\right]\mathsf{Pr}[\neg\mathcal{H}] = \frac{1}{2} - \frac{1}{2}\mathsf{Pr}[\mathcal{H}]$.

Hence, $\epsilon \leq \left|\mathsf{Pr}\left[c = c'\right] - \frac{1}{2}\right| \leq \frac{1}{2}\mathsf{Pr}[\mathcal{H}]$. So in real attack $\mathsf{Pr}[\mathcal{H}] \geq 2\epsilon$. From this, it follows that at the end of simulation D appears in some tuple with probability at least 2ϵ. So $\mathcal{B}$ outputs correct answer D to BDH with probability at least $2\epsilon/q_{H_2}$. $\square$

Now we completed Stage 2 of proof of theorem 1. So now, by combining the results from Lemma 1 and Lemma 2, we receive the proof of theorem 1. $\square$

4.3 Full IBE scheme and its security definition

The IBE scheme presented above is not resistant against active attacks, that is chosen ciphertext attack. By applying the Fujisaki-Okamoto transformation [5] to an assymetric BasicIdent scheme we can produce symmetric IBE scheme that is $\mathsf{IND}-\mathsf{ID}-\mathsf{CCA}$ secure. Below we present the definition of FullIdent scheme, which is derived from BasicIdent scheme.

Setup: As in BasicIdent scheme, plus we pick hash functions $H_3 : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_q^*$ and $H_4 : \{0,1\}^n \to \{0,1\}^n$ for some n.

Extract (key-generation): As in BasicIdent scheme.

Encrypt: To encrypt message $m \in \{0,1\}^n$ with public key ID:

(1) compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID})$, (2) choose a random $\sigma \in \{0,1\}^n$, (3) set $r = H_3(\sigma, m)$, (4) compute $g_{\mathsf{ID}} = e(Q_{\mathsf{ID}}, P_{pub})$, (5) compute $U = rP$, $V = \sigma \oplus H_2(g_{\mathsf{ID}}^r)$, $W = m \oplus H_4(\sigma)$. (6) Set cipher text to be $C = \langle U, V, W\rangle$.

Decrypt: To decrypt C using private key d_{ID}: If $U \notin G_1^*$ reject ciphertext, else:

(1) compute $V \oplus H_2(e(d_{\mathsf{ID}}, U) = \sigma$, (2) compute $W \oplus H_4(\sigma) = m$, (3) set $r = H_3(\sigma, m)$ and test $U = rP$, if not, reject ciphertext. (4) If (3) is succeesful, then m is decrypttion of C.

This FullIdent scheme is $\mathsf{IND} - \mathsf{ID} - \mathsf{CCA}$ secure. We will leave the proof of the security out of this survey, since it is very much similar to the security proof of BasicIdent with some additions from results obtained in [5]. We refer interested readers to [1] for complete security proof of FullIdent scheme.

5 Conclusion

This survey presented one of the recent constructive applications of bilinear pairing in cryptography. The bilinear pairings were used to build fully functional IBE system, which is resistant to chosen ciphertext attacks under the random oracle model. Whether this IBE system is secure in standart computational model is an open question, taking into acount the validity of random oracle models. Please, refer to previous seminar for discussion about random oracle models. To build the concrete example of IBE, authors use the Weil pairings. For some implementation results, interested readers can refer to [10].

The work [1] is not the only recent work to reserach the subject. For example Cocks in [2] also proposes secure IBE system, which is built on different algebraic structure.

Also, it should be noted that building a chosen ciphertext secure IBE in the standard model and not in random oracle is still an open problem. Nevertheless, the results presented in [1] certainly demonstrate that IBE systems can be build by using very different complexity assumptions.

References

[1] D.Boneh, M.Franklin, "Identity-based encryption from the Weil pairing", In *Advances in Cryptology - Crypto 2001*, volume 2139 of LNCS, pages 213-229. Springer-Verlag, 2001.

[2] C.Cocks, "An identity-based encryption scheme based on quadratic residues", Eighth IMA International Conference on Cryptography and Coding, Dec. 2001, Royal Agricultural College, Cirencester, UK.

[3] Y.Desmedt, J.Quisquater, "Public-key systems based on the difficulty of tampering", in *Advances in Cryptology - Crypto '86*, Lecture Notes in Computer Science, Vol. 263, Springer-Verlag, pp. 111-117, 1986.

[4] A.Fiat, A.Shamir, "how to prove yourself: Practical solutions to identification and signature problms", in *Advances in Cryptology - Crypto '86*, Lecture Notes in Computer Science, Vol. 263, Springer-Verlag, pp. 186-194, 1986.

[5] E.Fujisaki, T.Okamoto, "Secure integration of asymmetric and symmetric encryption schemes", in *Advances in Cryptology - Crypto '99*, Lecture Notes in Computer Science, Vol. 1666, Springer-Verlag, pp. 537-554, 1999.

[6] U.Maurer, Y.Yacobi, "Non-interactive public-key cryptography", in *Advances in Cryptology - Crypto '91*, Lecture Notes in Computer Science, Vol. 547, Springer-Verlag, pp. 498-507, 1991.

[7] A. Shamir, "Identity-based cryptosystems and signature schemes", in *Advances in Cryptology - Crypto '84*, Lecture Notes in Computer Science, Vol. 196, Springer-Verlag, pp. 47-53, 1984.

[8] H.Tanaka, "A realization scheme for the identity-based cryptosystem", in *Advances in Cryptology - Crypto '87*, Lecture Notes in Computer Science, Vol. 293, Springer-Verlag, pp. 341-349, 1987.

[9] S.Tsuji, T.Itoh, "An ID-based cryptosystem based on the discrete logarithm problem", IEEE Journal on Selected Areas iN Communication, vol. 7, no. 4, pp. 467-473, 1989.

[10] http://crypto.stanford.edu/ibe/download.html