SECURITY PROOF FOR SHENGBAO WANG'S IDENTITY-BASED ENCRYPTION SCHEME

Sunder Lal and Priyam Sharma
Derpartment of Mathematics, Dr. B.R.A.(Agra), University,
Agra-282002(UP), India.
E-mail- sunder_lal2@rediffmail.com, priyam_sharma.ibs@rediffmail.com

Abstract: This paper analyzes the security of an IBE scheme proposed by Wang in 2007. It is shown that under BDHP (which is polynomially time equivalent to BIDHP) assumption the scheme is secure in random oracle model.

Key-Words: Public Key Encryption, Identity-Based Encryption (IBE), One-Way Encryption (OWE), One-Way Identity-Based Encryption (ID-OWE).

1. Introduction:

In 1984, Shamir [2] introduced the idea of identity-Based cryptosystem, in which the public key of a user is derived from his identity. The idea is to eliminate the need for directory and certificates, which are used in traditional public key cryptosystems where public keys are generated by the users at random.

Since 1984, there have been several proposals to realize identity-based encryption (IBE) schemes. However, it was only in 2001, that Dan Boneh and Matt Franklin [1] came up with first fully functional solution for IBE. It was realized using bilinear pairings over elliptic curves. Boneh and Franklin also gave the security proofs for their scheme. The scheme relies on the BDH problem for its security.

Recently, Shengbao Wang [3] has proposed another IBE scheme based on bilinear pairing. This scheme is more practical in a multiple Private Key Generator (PKG) environment. However, the security aspect of this scheme is left open [3].

In this paper, we analyze the security of the IBE scheme by Wang. We show that the security of the scheme is secure under the BDH assumption in the random oracle model. It may be noted that according to Zhang, Safavi-Naini and Susilo [4] BDHP is polynomially equivalent to BIDHP.

2. Preliminaries:

2.1 Identity-Based Encryption (IBE) Scheme:

An *identity-Based Encryption Scheme* consists of four randomized algorithms: **Setup**, **Extract**, **Encrypt**, **and Decrypt.**

Setup: It takes a security parameter k and returns system parameters **params** and **master-key.** The **params** which is known publically includes the description of a finite message space $\mathcal{M}$ and the description of a finite ciphertext space $\mathcal{C}$. The master-key is known only to the private key generator (PKG).

Extract: This algorithm extracts private key from the given public key. It takes as input **params**, the **master-key** and a string $ID \in \{0, 1\}^*$, and returns a private key d_{ID}. ID which is an arbitrary string will be used as public key, and d_{ID} as the corresponding private key.

Encrypt: Takes as input the **params,** ID and $M \in \mathcal{M}$ and returns a ciphertext $C \in \mathcal{C}$.

Decrypt: Takes as input **params**, a private key d_{ID}, and $C \in \mathcal{C}$. and returns $M \in \mathcal{M}$.

If **params** is the system parameters produced by the **Setup** algorithm, d_{ID} is the private key, corresponding to ID, which is generated by the algorithm **Extract,** then for $M \in \mathcal{M}$,

$$\textbf{Decrypt (params, } d_{ID} \textbf{, Encrypt (params, ID, M)) = M.}$$

2.2 One-Way Identity-Based Encryption:

For a public-key encryption One-Way Encryption (OWE) is defined by the following game:
The adversary is given a public-key K_{Pub}, which is random and a ciphertext C, which is the encryption of a random plaintext M using K_{Pub}. The goal of the adversary is to recover the corresponding plaintext M. A public key encryption scheme is said to be a OWE scheme if no polynomially bounded adversary has a non-negligible advantage in attacking the scheme.
This definition of OWE may be strengthened to ID-OWE allowing the adversary to obtain some of the private keys. Thus, One-Way Identity-Based Encryption (ID-OWE) is defined through the following game:

Setup: The challenger takes a security parameter k and runs the **Setup** algorithm. She then returns public system parameters **params** to the adversary and keeps the **master-key** to itself.

Phase1: The adversary issues private-key extraction queries ID_1, ID_2, …………..., ID_n. The challenger responds by running the algorithm **extract** to generate the private-key d_i corresponding to the public-key ID_i and returns to the adversary.

Challenge: The adversary outputs a public-key ID, different from ID_1, ID_2, …………, ID_n , on which she wishes to be challenged. The challenger picks a random plaintext $M \in \mathcal{M}$ and encrypts it using the public-key ID and sends the resulting ciphertext to the adversary.

Phase2: The adversary issues more private-key extraction queries ID_{n+1}, ID_{n+2}, ……., ID_t different from ID. The challenger responds as in Phase1.

Guess: The adversary outputs a guess $M' \in \mathcal{M}$ and wins if M' = M.

Such an adversary is referred as **ID-OWE adversary** and the advantage of such an adversary against the scheme is define to be Pr [M’ = M] where the probability is over the random choices made by the adversary and the challenger. An IBE scheme is an ID-**OWE scheme** if no polynomially bounded adversary has non-negligible advantage against the challenger in the game described above.

2.3 Bilinear Pairings:

Let G_1 be an additive group of order p, a prime and let P be a generator of G_1. Let G_2 be a multiplicative group of the same order p. A map $e : G_1 \times G_1 \rightarrow G_2$ is said to be a bilinear pairing if it satisfies the following properties:

(Bilinearity): For all P, Q $\in$ G_1 and a, b $\in$ Z_p^*, $e(aP, bP) = e(P, P)^{ab}$.

(Non-Degeneracy): For a given R $\in$ G_1, $e(Q, R) = 1$, for all Q $\in$ G_1 if and only if $R = 0$, where 1 is the identity of G_2 and 0 is the identity of G_1.

(Computability): For all P, Q$\in$$G_1$, then there is an efficient algorithm to compute $e(P, Q)$ in polynomial time.

Some mathematical problems in G_1, G_2 are described as follows:

- **Computational Diffie-Hellman Problem (CDHP):** Given P, aP, bP in G_1, for some (unknown) a, b$\in$$Z_p^*$, compute abP in G_1.
- **Bilinear Diffe-Hellman Problem (BDHP):** Given P, aP, bP, cP in G_1, for some (unknown) a, b$\in$$Z_p^*$compute $e(P, P)^{abc}$ in G_2.
- **Bilinear Inverse Diffie-Hellman Problem (BIDHP):** Given P, aP, bP in G_1, for some (unknown) a, b, c$\in$$Z_p^*$, compute $e(P, P)^{a^{-1}b}$ in G_2.
- **Bilinear Square Diffie-Hellman Problem (BSDHP):** Given P, aP, bP in G_1 for some (unknown) a, b$\in$$Z_p^*$, compute $e(P, P)^{a^2 b}$ in G_2.

It is easy to show that, if we have an algorithm to solve the CDHP in G_1 or G_2, then we can use this algorithm to solve BDHP in <G_1, G_2, e>. In other words, the BDHP in <G_1, G_2, e> is no harder than the CDHP in G_1 or G_2. But, the problem that the CDHP in G_1 or G_2 is no harder than the BDHP is still an open problem. Also, it is shown in [4] that BDHP, BIDHP, and BSDHP are all polynomial time equivalent.

2.4 IBE Scheme by Wang:

We first now describe the IBE scheme proposed by Wang [3]. The scheme consists of the following four algorithms:

Setup: The algorithm works as follows:

1. Run $\mathcal{IG}$ on input k to generate two prime order groups G_1 and G_2 and a bilinear map $e: G_1 \times G_1 \rightarrow G_2$. Here $|G|_1$=$|G_2|$=p and G_1= < P >

2. Choose $s \in Z_p^*$ and computes $P_{Pub} = s^{-1}P \in G_1^*$.

3. For a suitable $n \in \mathbb{N}$, choses the message space $\mathcal{M}=\{0,1\}^n$, the ciphertext space $\mathcal{C}= G_1^* \text{x} \{0,1\}^n$ and two cryptographic hash functions $H_1: \{0,1\}^* \rightarrow G_1$ and $H_2: G_2 \rightarrow \{0, 1\}^n$.

The params is <G_1, G_2, e, n, p, P, P_{Pub}, H_1, H_2>, and the master-key is s.

Extract: For an identity $ID \in \{0,1\}^n$, PKG computes

1. $Q_{ID} = H_1(ID) \in G_1^*$ as public key, and

2. $d_{ID} = sQ_{ID}$ as private key.

Encrypt: To encrypt message $m \in \mathcal{M}$ for user with identity ID the sender

1. picks a random $r \in Z_p^*$

2. computes $Q_{ID} = H_1(ID)$ and $g_{ID} = e(P, Q_{ID}) \in G_2$, and

3. sets the ciphertext $C=< rP_{Pub}, m \oplus H_2 (g_{ID}^r) >$.

Decrypt: To decrypt a ciphertext $C=<U, V> \in \mathcal{C}$, the receiver using the private key d_{ID}, and params < G_1, G_2, e, n, p, P, $P_{Pub,}$ Q_{ID}, H_2>

1. computes $m = V \oplus H_2 (e (U, d_{ID}))$, and

2. returns m.

The correctness follows from $e(U,d_{ID}) = e(rs^{-1}P, sQ_{ID}) = e(P,Q_{ID})^r$

2 Security Analysis:

We now show that the IBE scheme of Wang above is a One-Way Identity-Based Encryption Scheme (ID-OWE) assuming that the BIDHP is hard. We prove the following theorem:

Theorem: Let H_1, H_2 be random oracles. Suppose there is an ID-OWE attacker $\mathcal{A}$ that has advantage ε against the IBE scheme of Wang which makes atmost $q_E > 0$ private key extraction queries to H_1 and $q_{H_2} > 0$ hash queries to H_2. Then there is an algorithm $\mathcal{B}$ that solves BDHP in $\mathcal{IG}$ with advantage at least

$$\frac{\varepsilon}{e(1+q_E).q_{H_2}} - \frac{1}{2^n.q_{H_2}}$$

where $e \approx 2.71$ is the base of natural logarithm. The running time of algorithm $\mathcal{B}$ is O (time ($\mathcal{A}$)).

To prove the above theorem we make use of the following public-key encryption scheme called as BasicPub-Wang:

3.1BasicPub:

The scheme has three algorithms: **Keygen, Encrypt, and Decrypt**. Algorithms Encrypt and Decrypt are same as that of IBE scheme of Wang.
The scheme is as follows:

Keygen: The algorithm works as follows:

1. As in the **Setup** algorithm of IBE scheme of Wang, $\mathcal{IG}$ generates two prime order groups G_1, G_2 and a bilinear map $e: G_1 \times G_1 \to G_2$. Also, the PKG computes its public key P_{Pub} and secret key s in the same way.

2. The message space $\mathcal{M} = \{0,1\}^n$, the ciphertext space $\mathcal{C} = G_1^* x \{0,1\}^n$ and a cryptographic hash function $H_2: G_2 \to \{0, 1\}^n$ are chosen in the same way.

3. The algorithm now picks a random point Q_{ID} in G_1^*, the group generated by P.

4. The public key is $<G_1, G_2, e, n, p, P, P_{Pub}, Q_{ID}, H_2>$, the private key is $d_{ID}=sQ_{ID} \in G_1^*$.

Encrypt: To encrypt $m \in \{0, 1\}^n$, the algorithm choses random $r \in Z_p^*$ and computes $C=< rP_{Pub}, m \oplus H_2 (g_{ID}^r) >$, where $g_{ID} = e (P, Q_{ID}) \in G_2^*$

Decrypt: To decrypt $C=<U, V>$ the algorithm takes $< G_1, G_2, e, n, p, P, P_{Pub}, Q_{ID}, H_2>$ and private key d_{ID} as input,

1. computes $m = V \oplus H_2 (e (U, d_{ID}))$, and

2. returns m.

To prove the theorem, we proceed in three steps. In the first step we show that ID-OWE attack on Wang's scheme can be converted into OWE attack on BasicPub-Wang. This will show that private key extraction queries do not help the adversary. In the second step we show that OWE attack on BasicPub-Wang can be converted into an algorithm to solve BIDH Problem. In the third step we use the result that the BIDHP is pollynomially time equivalent to BDHP [4]. Therefore, OWE attack on BasicPub-Wang can be converted into an algorithm to solve BDHP.

Theorem1.1: Let H_1 be a random oracle from $\{0, 1\}^*$ to G_1^*. Let $\mathcal{A}$ be an adversary that has advantage ε against the IBE scheme of Wang. Suppose $\mathcal{A}$ makes atmost $q_E > 0$ private key extraction queries. Then there is a OWE adversary $\mathcal{B}$ against BasicPub-Wang having advantage at least $\frac{\varepsilon}{e(1+q_E)}$. The running time of $\mathcal{B}$ is O (time ($\mathcal{A}$)).

Proof: The game starts with the challenger who generates a random public key by running algorithm **keygen** of BasicPub-Wang. The result is a public key by K_{Pub} = <G_1, G_2, e, n, p, P, P_{Pub}, Q_{ID}, H_2> and a private key $d_{ID} = sQ_{ID}$. Let $|G_1|=p=|G_2|$.
The challenger picks a random $M \in \{0,1\}^n$ and encrypts it using algorithm **encrypt** of BasicPub-Wang. It gives K_{pub} and the resulting ciphertext C=<U, V> to adversary $\mathcal{B}$.

Setup: $\mathcal{B}$ gives algorithm $\mathcal{A}$ the system parameters of the IBE scheme of Wang <G_1, G_2 ,e ,n , p, P, P_{Pub} ,Q_{ID} ,H_1, H_2> where G_1 , G_2 ,e ,n , p, P, P_{Pub} ,Q_{ID} , H_2 are taken from K_{Pub}, and H_1 is a random oracle controlled by $\mathcal{B}$.

H_1-queries: At any time algorithm $\mathcal{A}$ can query the random oracle H_1.To respond to these queries algorithm $\mathcal{B}$ maintains a list say H_1-list of tuples < ID_j , Q_j , b_j , $coin_j$ > as explained below :

When algorithm A queries the oracle H_1 at a point ID_j algorithm $\mathcal{B}$ responds as follows:

- If the query ID_j already appears on the H_1-list in a tuple < ID_j , Q_j , b_j , $coin_j$ > then algorithm $\mathcal{B}$ responds with $H_1(ID_i) = Q_i \in G_1$

- Otherwise, $\mathcal{B}$ generates a random bit coin$\in \{0,1\}$ so that Pr [coin=0] = δ for some $\delta = 1 - \frac{1}{q_E + 1}$.

- Algorithm $\mathcal{B}$ picks a random $0 \neq b \in Z_p^*$. If coin = 0 it computes $Q_i = b\, P_{Pub} \in G_1^*$, and if coin =1 it computes $Q_i = b\, Q_{ID} \in G_1^*$.

- Algorithm $\mathcal{B}$ adds the tuple < ID_i , Q_i , b_i , $coin_i$ > to the H_1-list and responds to $\mathcal{A}$ with $H_1(ID_i) = Q_i$

In both the cases, the distribution Q_i of is uniform in G_1^* and independent of $\mathcal{A}$'s view.

Phase 1: Algorithm $\mathcal{A}$ issues private key extraction queries. Algorithm $\mathcal{B}$ responds to these queries as follows:

- Runs the above algorithm for responding to H_1-queries to obtain a $Q_i \in G_1^*$ such that $H_1 (ID_i) = Q_i$. Let $< ID_i, Q_i, b_i, coin_i >$ be the corresponding tuple on the H_1-list. If $coin_i = 1$, then $\mathcal{B}$ reports failure and terminates. The attack on the BasicPub-Wang fails.

- If $coin_i = 0$, then $Q_i = b_i P_{Pub}$. Define $d_i= b_i P$.

  $(d_i = sQ_i = s\, b_i\, P_{Pub}= s\, b_i\, s^{-1}P = b_i\, s\, s^{-1}P = b_i\, P)$

Challenge: Once algorithm $\mathcal{A}$ decides that Phase1 is over, it outputs a public key $ID \in \{0, 1\}^*$ on which it wishes to be challenged. $\mathcal{B}$ responds as follows:

1. Run the above algorithm for responding to H_1-queries to obtain $Q \in G_1^*$ such that $H_1 (ID) = Q$. Let $< ID, Q, b, coin >$ be the corresponding tuple on the H_1-list. If $coin = 0$, then $\mathcal{B}$ reports failure and terminates. The attack on BasicPub-Wang failed.

2. We know, if $coin = 1$, $Q = bQ_{ID}$.

3. $C=< U, V >$ is the challenged ciphertext given to algorithm $\mathcal{B}$. Algorithm $\mathcal{B}$ sets $C' = < b^{-1}U, V >$ where b^{-1} is the inverse of b mod p. Algorithm $\mathcal{B}$ responds to algorithm $\mathcal{A}$ with the challenge C'.

C' is an encryption of M using Wang's scheme under the public key ID as required. Since $H_1 (ID) = Q$. The corresponding private key is $d_{ID}' = sQ$
Also,
$e(b^{-1}U , d_{ID}) = e(b^{-1}U , sQ) = e(U , b^{-1} sQ) = e(U , b^{-1} sbQ_{ID}) = e(U , sQ_{ID}) = e(U , d_{ID})$.

Hence, the decryption of C', using Wang's scheme, using d_{ID}' is the same as the BasicPub-Wang encryption of C using d_{ID}.

Phase 2: Algorithm $\mathcal{B}$ responds to private key extraction query in the same way as it did in Phase 1.

Guess: Eventually, algorithm $\mathcal{A}$ will produce a guess M'. Algorithm $\mathcal{B}$ outputs M' as its guess for the decryption of C.

■

Claim: If algorithm $\mathcal{B}$ does not abort during the simulation, then algorithm $\mathcal{A}$'s view is identical to its view in the real attack. And, if algorithm $\mathcal{B}$ does not abort, then

Pr [M=M']$\geq \varepsilon$, where probability ε is over the random bits use by algorithms $\mathcal{A}$, $\mathcal{B}$ and the challenger.

Proof: If algorithm $\mathcal{B}$ does not abort, then all responses given by the H_1-oracle are uniformly an independently distributed in G_1*, all responses to the private key extraction queries are valid and the challenged ciphertext C' is the encryption of a random plaintext $M \in \mathbb{M}$. Thus, algorithm $\mathcal{A}$'s view is identical to its view in the real attack. The challenge ciphertext C' given to algorithm $\mathcal{A}$ is the encryption of M using Wang's under the public identity ID chosen by algorithm $\mathcal{B}$. Hence, by definition of algorithm $\mathcal{A}$, it will make the correct guess with probability at least ε.

■

Now we shall compute the probability that algorithm $\mathcal{B}$ does not abort during the simulation. If algorithm $\mathcal{A}$ makes at most q_E private key extraction queries, then the probability does not abort while treating one of those queries is δ^{q_E}. The probability that algorithm $\mathcal{B}$ does not abort during the simulation is $1-\delta$.Therefore, the probability that algorithm B does not abort during the simulation is $\delta^{q_E}(1-\delta)$ .The value of δ is chosen to be $\delta = 1 - \frac{1}{q_E + 1}$ as we want to maximize this function. The probability that algorithm $\mathcal{B}$ does not abort is at least $\frac{1}{e(1+q_E)}$.

Note that, probability that algorithm $\mathcal{B}$ does not abort is $\delta^{q_E}(1-\delta)$

$$\delta^{q_E}(1-\delta) = [1-(q_E+1)^{-1}]^{q_E}[1-1+(q_E+1)^{-1}] = [1-(q_E+1)^{-1}]^{q_E}(q_E+1)^{-1}$$

also, $Lim_{q_E \to 0}[1-(q_E+1)^{-1}]^{q_E} = e^{-1}$

Theorem1.2: Let H_2 be a random oracle from G_2 to $\{0, 1\}^n$. Let algorithm $\mathcal{A}$ be a OWE adversary that has advantage ε against BasicPub-Wang. Suppose algorithm $\mathcal{A}$ makes a total $q_{H_2} > 0$ queries to H_2. Then there is an algorithm $\mathcal{B}$ that can solve BIDHP in $\mathcal{IG}$ with advantage at least

$$\frac{(\varepsilon - \frac{1}{2^n})}{q_{H_2}}$$

and running time O (time $\mathcal{A}$)).

Proof: Algorithm $\mathcal{B}$ is given an input the BIDH parameters <G_1, G_2, e > produced by $\mathcal{IG}$ and a random instance <P, aP, bP> of the BIDH problem for these parameters i.e., $P \in_R G_1^*$ where a, $b \in_R \mathbb{Z}_p^*$. $|G_1| = p = |G_2|$. Let $D = e(P,P)^{a^{-1}b} \in G_2$ be the solution to this problem. Algorithm $\mathcal{B}$ finds D by interacting with algorithm A as follows:

Challenge: Algorithm $\mathcal{B}$ creates the BasicPub public key K_{Pub} = < G_1, G_2, e, n, P, P_{Pub}, Q_{ID}, H_2> by setting P_{Pub} = aP, Q_{ID}= bP. Algorithm $\mathcal{B}$ then picks a random string $R \in \{0, 1\}^n$ and defines C to be the ciphertext, C = <U, V> where U = P and V = R. Algorithm $\mathcal{B}$ gives K_{Pub} and C as the challenge to algorithm $\mathcal{A}$.
Observe that, the private key associated to K_{Pub} is $d_{ID} = a^{-1} Q_{ID}$
$= a^{-1}bP$. Also, the decryption of C is $V \oplus H_2 (e (u, d_{ID}))$ =
$V \oplus H_2 (e (P, a^{-1}bP)) = V \oplus H_2\ (e(P,P)^{a^{-1}b}) = V \oplus H_2 (D)$
We set $M = V \oplus H_2 (D)$.

H_2-queries: At any time algorithm $\mathcal{A}$ may issue queries to H_2. To respond to these queries algorithm $\mathcal{B}$ maintains a list of pairs called the H_2-list. Each entry in the list is a pair of the form < X_j, H_j >. Initially the list is empty.

To respond to query X_j algorithm $\mathcal{B}$ does the following:

1. If the query X_j already appears on the H_2-list, then he responds with $H_2 (X_j) = H_j$.

2. Otherwise, algorithm $\mathcal{B}$ just picks a random string $H_j \in \{0, 1\}^n$ and adds the tuple <X_j, H_j> to the list. It responds to algorithm $\mathcal{A}$ with $H_2 (X_j) = H_j$.

Guess: Algorithm $\mathcal{A}$ outputs its guess M' to the decryption of C. At this point algorithm $\mathcal{B}$ picks a random pair <X_j , H_j> from the H_2-list and outputs X_j as the solution to the given instance of BIDHP. ■

Again, it is easy to see that $\mathcal{A}$'s view is identical to its view in the real attack. The setup is as in the real attack. Since a and b are random in Z_p^* so is the challenge.

Since, P is a random in G_1^* and therefore, the resulting encryption message is a random plaintext. Since it is exclusive-or of two random strings in $\{0,1\}^n$. Thus $\Pr[M' = M] \geq \varepsilon$. It still remains to calculate the probability that algorithm $\mathcal{B}$ outputs the correct result.

Let H denote the event that at the end of the simulation D appears in a pair on H_2-lists. Let $\Pr[H] = \delta$. If D does not appear in H_2-lists, then the decryption of C is independent of $\mathcal{A}$'s view, since H_2 (D) is a random strisng in $\{0,1\}^n$ independent of $\mathcal{A}$'s view. Thus, $\Pr[M' = M | \neg H] \leq \frac{1}{2^n}$

Therefore,

$$\varepsilon \leq \Pr[M' = M] = \Pr[M' = M | \neg H].\Pr[H] + \Pr[M' = M | \neg H].\Pr[\neg H]$$

$$\leq \Pr[H] + \Pr[M' = M | \neg H].\Pr[\neg H] \leq \delta + \frac{1}{2^n}(1-\delta) = \delta - \frac{\delta}{2^n} + \frac{1}{2^n}$$

$$\Rightarrow \varepsilon \le \delta - \frac{\delta}{2^n} + \frac{1}{2^n} \Rightarrow \delta - \frac{\delta}{2^n} \ge \varepsilon - \frac{1}{2^n}$$

$$\Pr[H] = \delta \ge \delta - \frac{\delta}{2^n} \ge \varepsilon - \frac{1}{2^n}$$

Also, since we pick a random element from H_2-list, the probability that algorithm $\mathcal{B}$ produces the right answer is at least

$$\Pr[H] \ge \frac{(\varepsilon - \frac{1}{2^n})}{q_{H_2}}$$

Note that, if algorithm $\mathcal{A}$ answers correctly, then $V \oplus M' = H_2(D)$. So algorithm $\mathcal{B}$ could scan through the H_2-list, and pick a random pair $<X_j, H_j>$ such that $H_j = H_2(D)$, and output X_j instead of picking a random pair in all the H_2-list. Suppose that n is very large, so that $2^{n/2}$ represents an infeasible number of computations. Then, if we knew that whenever algorithm $\mathcal{A}$ makes less that $2^{n/2}$ H_2-queries, the probability that more than k of these queries result in the same hash value is a negligible function $f(D)$, then the probability that algorithm $\mathcal{B}$ produces the right answer is at least

$$\frac{\left(\varepsilon - \frac{1}{2^n} - f(D)\right)}{k}$$

In Theorem 2 of [3], Zhang, Safavi-Naini and Susilo have shown that BDHP, BIDHP and BSDHP are all polynomial time equivalent. Using this result we infer, that the scheme proposed by Wang is secure so long as the BDHP is difficult. Therefore, we get the following theorem,

Theorem1.3: Let H_2 be a random oracle from G_2 to $\{0, 1\}^n$. Let algorithm $\mathcal{A}$ be a OWE adversary that has advantage ε against BasicPub-Wang. Suppose algorithm $\mathcal{A}$ makes a total $q_{H_2} > 0$ queries to H_2. Then there is an algorithm $\mathcal{B}$ that can solve BDH problem in $\mathcal{IG}$ with advantage at least $\dfrac{\left(\varepsilon - \frac{1}{2^n}\right)}{q_{H_2}}$ and running time O (time ($\mathcal{A}$)).

Proof of the Theorem: Directly from the results from Theorem1.1, Theorem1.2 and Theorem1.3, we get that, if there exists an ID-OWE adversary against algorithm $\mathcal{A}$ that has advantage ε against IBE scheme of Wang, then there is an algorithm $\mathcal{B}$ that can solve BDHP for $\mathcal{IG}$ with advantage at least

$$.\left[\frac{\left(\frac{\varepsilon}{e(1+q_E)} - \frac{1}{2^n}\right)}{q_{H_2}}\right] = \frac{\varepsilon}{e(1+q_E).q_{H_2}} - \frac{1}{2^n.q_{H_2}} \text{ as required.}$$

■

4. Summary:

In this paper we give the security proofs for the IBE scheme proposed by Wang, which is more practical in multiple PKG environments than the famous IBE scheme proposed by Boneh and Franklin.

References:

1) D.Boneh and M.Franklin,"Identity-based encryption from weil pairing", In Proc. Of CRYPTO 2001, LNCS # 2139, pp.213-229. Springer-Verlag, 2001.

2) A.Shamir, "Identity-based cryptosystems and signature schemes", In Proc. Of CRYPTO 1984, LNCS # 196, pp.47-53. Springer-Verlag, 1984. Also available on http://www.iseca.org/downloads/shamir47.pdf.

3) S.Wang,"Practical Identity-Based Encryption (IBE) in Multiple PKG Environments and Its Applications", http://eprint.iacr.org/2007/100.pdf.

4) F.Zhang, R.Safavi-Naini, and W.Susilo,"An efficient signature scheme from bilinear pairings and its applications", In International Workshop on Practice and Theory in Public Key Cryptography-PKC'2004. LNCS # 2947, pp.277-290, Springer-Verlag, 2004.