An ID-based Authenticated Key Exchange Protocol

Mahender Kumar[1], C.P. Katti[2], P.C. Saxena[3]
School of computer and system science,
Jawahar Lal Nehru University, Delhi, India

Abstract: **One of the main problems in cryptosystem is to distribute the secret key over an unsafe network. Several schemes have been introduced in the distribution of the secret key. Whitfield Diffie and Martin Hellman were the first to establish the first feasible approach for constructing a shared secret over an insecure communications network without meeting in advance. This scheme is restricted to key exchange only. Because it takes place in a certain mathematical environment and no user authentication is there. Therefore, this scheme is vulnerable to several attacks. Nan Li overcomes some attack problems using the services of a third party but still vulnerable to many attack problems. We ask a question: can we have a scheme which shared secret over an insecure channel without using the service of third party and of course restricted to attacks problems? In this paper, we construct such scheme and prove its security in the standard model. In comparison with the recent proposed schemes, our scheme has proved to be best in terms of security.**

Keywords: - Diffie-Hellman key exchange scheme, Authentication Server, Identity-based signature (IBS), Forking Lemma.

I. INTRODUCTION

In 1976, Whitfield Diffie and Martin Hellman published a key exchange protocol [1] based on the discrete logarithm problem. *Diffie-Hellman key exchange scheme* permits two parties to share his/her secret key over an insecure network without any knowledge of each other. Nevertheless, the advantages come with some drawbacks. However, the key exchange scheme without authentication is no longer secure against several attacks. Diffe-hellman have no user authentication and take place in a certain mathematical environment. Therefore, this scheme is subjected to man-in-middle attack, impersonation attack, replay attack etc. Since then, many schemes have been presented [3, 16, 17, 18, 25, 26] to deliver user authentication key exchange schemes. Most of these schemes use the hash algorithms. A protocol is required to authenticate the users to prevent these attack problems. Nan Li [3] proposed an improved protocol for key exchange based on hash algorithm. This scheme uses the service of a third party, known as *Authentication server*, for user authentication as a result this scheme is able to solve many of these attack problems. For our required scheme, we need such kind of signature scheme which itself authenticate the users without using any *authentication server* separately. In 1984, Shamir [2] introduced a signature scheme with an extra advantage; instead of generating the signature with the Private/Public key pair, this scheme uses the receiver identity as the public key. This scheme is known as *Identity-based signature* scheme which enables two communicating parties to securely communicate and verify each other's signatures without exchanging the pair of keys and without using the services of third parties. *Identity-based signature* scheme is considered to be suitable for our key exchange scheme as it fulfill all needed requirements. Since then, there are many ID-based signature schemes which have been presented in [7, 9, 12, 13, 14, 15, 20]. Most of them are based on integer factorization including the Shamir's scheme [2] and GQ scheme [20] and the rest of them based on bilinear pairing on elliptic curves. At a recent time, Boneh and Franklin [7] suggested an ID-based encryption scheme based on bilinear maps on an elliptic curve. This scheme was the first practical ID-based encryption, but they did not implement the ID-based signature.

In this paper, we propose and implement an authenticated Key exchange scheme that provide the mutual authentication between two parties and prove its security in the standard model. Our work is to make a protocol that eliminates the service of *Authentication Server* and solves the attack problems in [3] scenario. Our scheme is provably secure Key exchange scheme based on RSA.

The remaining of part this paper organized as follows. In section 2, we provide an overview of *Diffie-Hellman key exchange scheme*, kind of attack, and improved protocol and discuss some relevant topic to design a secure and efficient protocol. In Section 3, we discuss the *identity-based signature* scheme. And the sorts of security models are examined in section 4. The goal of this paper is how we authenticate the user without *authentication server*, implemented in Section 5. In section 6, we prove that our scheme is secure against existential forgery on adaptively chosen message and ID attack. We analyze our scheme and also proved that our scheme is free from attacks. In Section 7, we conclude the paper.

II. RELATED WORKS

A. Diffie-Hellman Key Exchange Protocol.

Infeasibility of extracting discrete logarithm defines the security of Diffie-Hellman key exchange algorithm. First, we briefly understand the discrete logarithm and some related term as below.

*Discrete logarithm***:** Suppose b is any integer less than p such that $b= \alpha^i \bmod p$, where, integer α is a primitive root of prime number p and i is the distinctive exponent is said to be discrete logarithm such that domain of i is from 1 to p-1.

Primitive root: Suppose p is prime number. Then α is a primitive root for p if $\alpha \bmod p$, $\alpha^2 \bmod p$,.., $\alpha^{p-1} \bmod p$, include all integer from 1 to p-1.

Algorithm 1: *Diffie-Hellman key exchange scheme*

1. Suppose, Alice and Bob agree on values α and p and the want to exchange a secret key (α is the primitive root of large prime number p).
2. Alice chooses a random number $Pr_A < p$, computes the public key ($Pub_A=\alpha^{PrA} \bmod p$) and sends it to Bob.
3. Similarly, Bob chooses a random number $Pr_B < p$, computes the public key ($Pub_B=\alpha^{PrB} \bmod p$) and sends it to Alice as shown in Figure 1.
4. Both sides keep Pr as private and make Pub publically available to another side.
5. Alice receives Pub_B and calculates the secret key ($Sec_{AB} = (Pub_B)^{PrA} \bmod p$).
6. Similarly, Bob receives Pub_A and calculates the secret key ($Sec_{BA} = (Pub_A)^{PrB} \bmod p$).

Finally, Alice and Bob are ready to exchange a secret value Sec_{AB}. The correctness of similarity of the secret exchange on both sides will explain in Section 6.1.

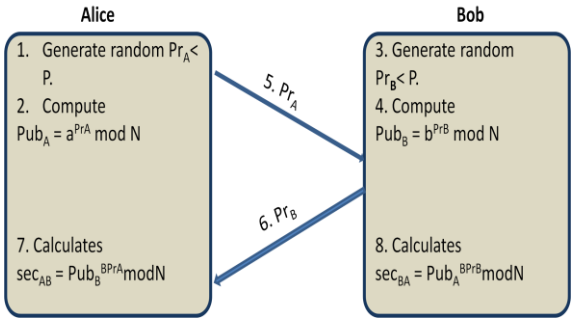

Figure 1 Diffie – Hellman key exchange [3]

From the Security point of view, we know Pr_A is the Alice's private key and p, α and Pub_A are the public parameters. An adversary (Eve) can compute the discrete logarithm $Pr_A= dlog_{\alpha, p} (Pub_A)$ to find his private key. For large prime p, it is infeasible to calculate the discrete logarithm. Thus, it is very hard to compute Sec_{AB} for an attacker, even he knows α, b and p.

Two well-known cryptographic problems are privacy: preventing the unauthorized extraction of information from communication over an insecure channel and authentication: prevents the unauthorized injection of the message into the public channel. Privacy of communication is done by public key cryptography [1]. But, due to lack of user authentication it subjected to several attacks, e.g. Man-in-middle attack, Impersonate attack, Replay attack, Non-Repudiation and Clogging attack.

B. Improved Key Exchange protocol by Nan Li.

Nan Li proposed in scheme [3] explores the Diffie-Hellman key exchange protocol and provide the user authentication with the help of *authentication server* and the hash algorithm (message digest 5).

Algorithm 2: Key exchange scheme by Nan Li.

1. Alice → AS , $ID_A||ID_B$
2. AS → Alice, $N_1 \oplus P_A$
3. AS → Bob, $N_1 \oplus P_B$

4. Alice →Bob, $Pub_A||H(Pub_A||N_1)$
5. Bob →Alice, $Pub_B||H(Pub_B||f(N_1))$
6. Alice→ Bob, $H(N1)$
7. Alice computes $K = (Pub_B)^{PrA} \bmod p$
   Bob computes $K = (Pub_A)^{PrB} \bmod p$

Where, AS is the *Authentication server* which facilitates authenticity for a user who attempts to access a network, ID_A and ID_B are Alice's and Bob's identity respectively, P_A and P_B are Alice and Bob's password respectively, N_1 is Nonce generated by the AS and is used to ensure that previous conversation cannot be reused, || is to concatenate two string, ⊕is the X-OR operator, f is simple transformation function, H is the hash function, p is very large prime number and publicly known to all.

Alice sends her and Bob identity to the AS as a response message, shown in Figure 2. On receiving both user's identities, AS responds the Alice's message by sending $(N_1 \oplus P_A)$ to her and $(N_1 \oplus P_B)$ to Bob. Alice and bob can obtain N_1 on decrypting $(N_1 \oplus P_A)$ and $(N_1 \oplus P_B)$ with $P_A$and P_B respectively. Now, $N_1$is shared between Alice and Bob. Now, Alice chooses a random number $Pr_A < p$, computes her public key and generates signature $(H(Pub_A||N_1))$ with her public key and Nonce as input parameter, concatenate them and sends it to Bob. Now, Bob generates signature $(H'(Pub_A||N_1))$ and check both signature. If both signatures are same, Bob ensure that this message is really coming from Alice, or stops this conversation. Similarly, Bob chooses a random number $Pr_B < p$, computes his public key, generates signature $(H(Pub_B||f(N_1)))$ and it Alice. Alice generates signature $(H'(Pub_B || f(N_1)))$ and check both signature. If both signature are same, Alice ensures this message is really sent by Bob and calculates the $Sec_{AB} = (Pub_B)^{Pr_A} \bmod p$, or stops this conversation. Now, Bob calculates $Sec_{BA} = (Pub_A)^{Pr_B} \bmod p = (Pub_A)^{Pr_B} \bmod p$, after obtained the confirmation message from Bob.

Use of *Authentication Server* in scheme successfully solves the following attack problems as follows:

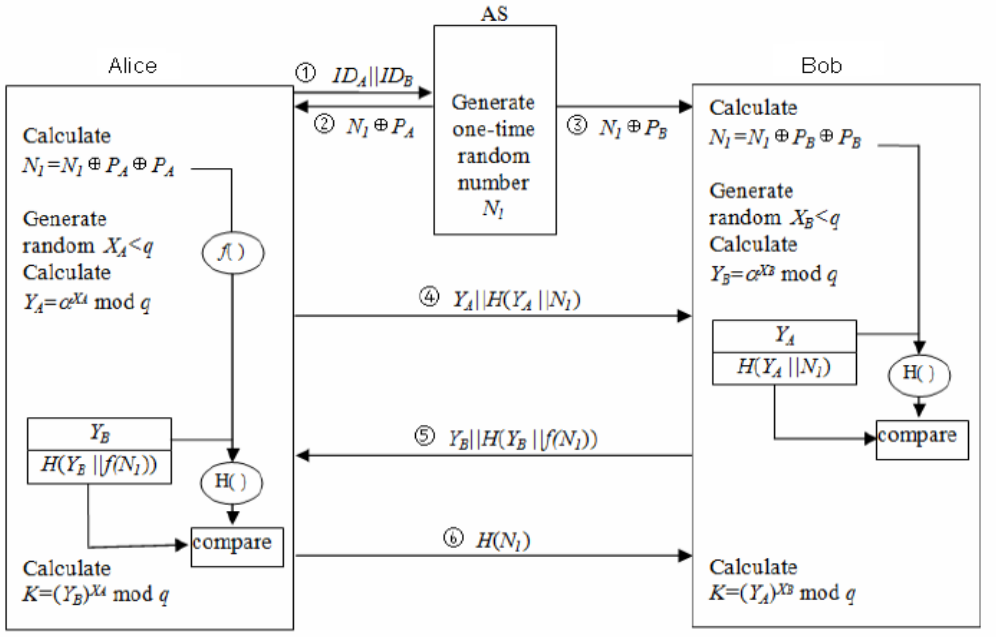

Figure 2 Improved DH key exchange protocol [3].

Nonce N1 ensured that Current response is the new. So, adversary cannot replay. Therefore, scheme is free from the replay attack.AS guaranteed that N_1 is known only known to Alice and Bob. So, both Alice and Bob ensure that they are really communicating with each other. Therefore, man-in-middle attack and impersonate attack is resisting. Scheme is also free from clogging attack because after confirmation acknowledges sent back from Alice, Bob computes the key.

With successfully elimination of man-in-middle attack, impersonate attack, replay attack and clogging attack, Nan Li did not consider about elimination of non-repudiation attack. As seen in Algorithm 2, first three steps used to provide the identity of the user to each other. There is no any identity used with message. Think of a situation after steps 3, where Nonce is leaked. Now, Alice may deny after sending a message to Bob or Bob may deny after receiving the message to Alice. Therefore, the scheme is subjected to Non repudiation. And second, this scheme is dependent on *Authentication server* for user authenticity.

Now, we have a question, "can we have a Key Exchange scheme which exchanges a secret key without communicating between two parties and without using the service of third party?" So, there is a need of such efficient scheme which generates the signature for user authentication (implement in Section 5). For the counterpart of attacks problem, we need signature a scheme which itself authenticate the users without using any *authentication server* separately.

III. IDENTITY-BASED SIGNATURE SCHEME

Signature can either be generated by identity-based encryption (IBE) or public key encryption (PKE). Both are kind of asymmetric cryptography [6]. Now, first we define some terminologies used in this section.

Public key encryption (PKE): It is kind of cryptographic algorithm which takes two different keys. One key is the private keys (K_d) which is secret to the user and the other key is a public key (K_e) which publishes publically to all. Both pairs of key are mathematically linked. Message encryption or signature verification is done by the public key, whereas the message decryption or signature generation is done by a private key as shown in Figure 3.

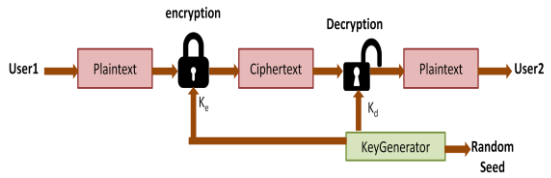

Figure 3 Public key Cryptography

Identity-based encryption (IBE): As shown in Figure 4, user's unique identification used to generates the public key. User's Identification may include name, phone number, voter Id number, e-mail address etc. Private Key generator generates the user's private key.

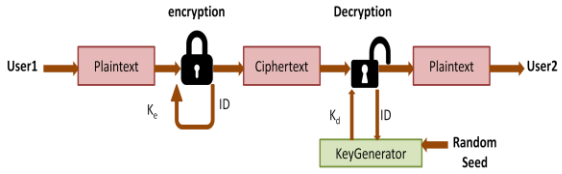

Figure 4 Identity based encryption

Identity-based signature (IBS): In *identity-based signature scheme*, message is signs with sender's private key (K_d) generated by the private key generation center, sends along with signature and the sender's identity ID, and verified with signature verification key (K_e).

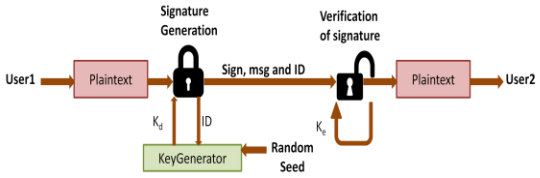

Figure 5 Identity based signature scheme

The difference between the two systems (PKE and IBE) is in the mathematical coordination and verifying between the public and private keys. In a PKE, certificate is used to achieve the coordination between the pair of key and user identity. On the contrary, in an ID-based encryption scheme, the mathematical linking between the private key and the user authenticity is managed by a Trusted Authority known as the private key generator (PKG) at the time of request. Management of the certificate and private key is the major problem in PKE. To overcome this problem, Identity-based encryption scheme was introduced by Shamir based on public key encryption. In 2001 Boneh and Franklin [7] presented a practical identity-based encryption scheme. Baek, et.al. [4] Sketch the fundamental issue regarding the IBE. They describe how practical and in which conditions IBE may be used in future environments.

A. Shamir's Identity Based Signature Scheme

Shamir was the first to propose a scheme [2] based on the public key encryption. The actual work

of Shamir gives the signature scheme based on the integer factorization problem of RSA, but could not be implemented for encryption.

Given input parameter <m, t, ID, f, N, e > where, m = message, <S, t > is the Signature, ID is user's identity, N is the product of two large number p and q, such that N=pq, e is a large prime which is relatively prime to totient function φ(N), f is the one way function is discrete logarithm problem. Shamir [1] scheme is based on verification condition: $S^e=ID.t^{f(t,m)}$ mod N. The value of N, e and f are chosen by PKG and all users have same N, e value and same algorithm of stored in their smart card and are made public, but the factorization of N is should be known only to KGC. Only difference is the value of ID (Public key) and secret key corresponding to ID that is $((K_{ID})^e$= ID.mod N), which can be easily calculated by KGC. To sign the message m, user chose a random number r and computes $t=r^e$ mod N and $S = K_{ID}.r^{f(t,m)}$)modN. The Public key is derived from user's identifiers. Therefore, it removes the requirements of the third trusted party. That was the big advantage of IBE scheme. The authenticity of the public keys is guaranteed completely as long as the transfer of the private keys to the corresponding user is kept secure.

B. GQ Identity Based Signature Scheme

Similar to Shamir scheme [2], GQ identity based signature scheme [20] is also based on the integer factorization problem of RSA, uses the trusted third party known as a private key generator, but their approach is different: instead of authenticate the users, this scheme authenticating the security device. This scheme consists of four algorithms (setup, Extract, Sign, and Verify).

Setup: PKG runs the setup algorithm, which generates N, product of two prime number and computes exponent e, d such that ed=mod φ(N). Now, d is the master secret key and (n, e) is the corresponding master public key. Choose two hash function H_1: $\{0,1\}^* \rightarrow \{0,1\}^l$ and H_2: $\{0,1\}^* \rightarrow Z_N^*$. Where H_1 and H_2 are the one-way function, l denotes the length of a message, Z_N denotes the group $\{0,\ldots,N-1\}$ and $Z_N^* = Z_N/\{0\}$. Suppose, $H_1(x,y) = x^y$ mod N take two parameters and H_2 takes one parameter defines as $H_2(x)$ = x mod N.

Extract: For any user's identity ID ∈ $\{0,1\}^*$, the PKG calculate $(K_{ID})^e = H_2(ID)$, where, K_{ID} is the private key for user ID.

Sign: Given message m, user's identity ID, user firstly chooses a random number r ∈ Z_N, and calculates signature σ = (s, t) where, $t=r^e$ mod N and $S=r.K_{ID}^{H1(t, m)}$ mod N.

Verify: Given message m, user's identity ID, t. Signer's Signature σ' = (s',t) is valid if and only if $S^e=t.H_2(ID)^{H(t,m)}$ mod N.

The basic idea of our scheme is to provide the user authentication so that he/she can prove himself/herself as a legitimate person. For user authenticity, the proposed scheme should have following requirements: signature must include user's identity, no third party is used to provide authenticity or any certificate authority to prove that the person is legitimate person and additionally, secure against an adaptive chosen cipher text attack and secure against existential forgery on adaptively chosen message and ID attack. Shamir signature scheme [2] and GQ signature scheme [20], both schemes are capable to fulfill our requirements. But from a security point of view, later scheme is more secure than first one, will discuss in the next section.

IV. SECURITY MODEL AND PROOF

Here we discuss the standard security models for ID-based signature scheme [19], Diffie-Hellman problem and assumption and *forking lemma* [21].

A. Attack model for ID-based signature scheme.

1. Secure against existential forgery on adaptively chosen message and ID attack.

Definition 1: Given some parameter (t, q_H, q_E, q_S, ε), an IBS forger $\mathcal{A}$ is said to break an IBS scheme if: $\mathcal{A}$ runs in time t', where t' ≤ t; $\mathcal{A}$ makes q'_H queries to the Hash function query, where $q'_H \leq qH$; q_E and q_S queries to the Extract function query and Sign query respectively, and Advantage of A is ε_0, where, $\varepsilon_0 \geq \varepsilon$. An IBS scheme have four algorithms (Setup, Extract, Sign and verify) and is secure against existential forgery on adaptively chosen message and ID attack if no forger breaks it.

Let an ID-based signature scheme, consist of four algorithms (setup, extract, Sign and Verify), is

secure against existential forgery on adaptively chosen message ID attacks if no polynomial time algorithm $\mathcal{A}$ has a non-negligible advantage against a challenger C in the following game:

a) C first run the setup, generates the master key-pair and the master-key pair given to $\mathcal{A}$.
b) $\mathcal{A}$ runs the given queries:
   - Hash query: Given some inputs, C runs the hash function and sends the output of the hash function to $\mathcal{A}$.
   - Extract query: C returns the Private Key corresponding to given Identity ID.
   - Sign query: C returns a signature σ′ given an identity ID and message m,
c) Eventually, $\mathcal{A}$ outputs (m, ID, σ), where m is the message, ID is user's identity, and σ is signature.

A wins the game if σ is a valid signature of m for ID.

2. Secure against existential forgery on adaptively chosen message and given ID attack

Definition 2: Given some parameter (t, q_H, q_E, q_S, ε), an IBS forger $\mathcal{A}$ is said to break an IBS scheme if: $\mathcal{A}$ runs in time t', where t' ≤ t; $\mathcal{A}$ makes q'_H queries to the Hash function query, where $q'_H \leq q_H$; q_E and qS queries to the Extract function query and Sign query respectively; and Advantage of $\mathcal{A}$ is ε_0, where, $\varepsilon_0 \geq \varepsilon$. An IBS scheme, which consist of four algorithms (Setup, Extract, Sign and verify) is secure against existential forgery on adaptively chosen message and given ID attack if no forger breaks it.

This game is similar to the previous game except in step 1, C first fix an ID, then sends master-key pair (mpk, msk) with this ID to $\mathcal{A}$, and in step 3, $\mathcal{A}$ must output the message and signature with the fixed ID.

Lemma 1: For an adaptively chosen message and ID attack togiven protocol with running time t and advantage ε, if there is an algorithm $\mathcal{A}$, then there is an algorithm $\mathcal{B}$ for an adaptively chosen message and given ID attack which has running time t' ≤ t and advantage $\varepsilon' \leq \varepsilon(1 - 1/l)/q_{H2}$, where q_{H2} is the maximum number of queries to H_2 asked by $\mathcal{A}$. In addition, the numbers of queries to hash functions, Extract, and Sign asked by $\mathcal{B}$ are the same as those of $\mathcal{A}$.

Proof: This Lemma has been proved in [19].

General Forking Lemma: M. Bellare and G. Neven in [22] state and prove the *forking lemma* that can be very fruitful to prove the security of our proposed scheme. This *forking lemma* is focus on the output response of an algorithm when run twice on similar input.

Lemma 2: [General Forking Lemma] Given an integer q at least 1 and a set H of size at least 2. Let x be the user's identity, on input x, $h_1,..,h_q$ randomized algorithm A returns two element, first one is an integer ∈{0, q} and the second one is a side output as we can say. Let $\mathcal{RA}$ be a randomize algorithm that we call the input generator. The accepting probability acc of A is the probability that I is at least 1

$$x \leftarrow \mathcal{RA} : h_1, \ldots, h_q \leftarrow H ; (I, \sigma)\$\leftarrow A(x, h_1, \ldots, h_q)$$

The forking algorithm F_A with A as the randomized algorithm that takes x as input proceed as follows:

Algorithm $F_A(x)$
Pick coins ρ for A at random
$h_1, \ldots, h_q \leftarrow H$
$(I, \sigma) \leftarrow A(x, h_1, \ldots, h_q; \rho)$
If I = 0 then return (0, ε, ε)
$h'_I, \ldots, h'_q \$\leftarrow H$
$(I', \sigma') \leftarrow A(x, h_1, \ldots, h_{I-1}, h'_I, \ldots, h'_q; \rho)$
If (I = I'and $h' \neq h'_I$) then return (1, σ, σ')
Else
return (0, ε, ε).

Let

$frk = Pr [b = 1 : x \leftarrow \mathcal{RA} ; (b, \sigma, \sigma') \leftarrow F_A(x)]$

Then

$$frk \geq acc.\left(\frac{acc}{q} - \frac{1}{h}\right) \quad (1)$$

Alternatively,

$$acc \leq \frac{q}{h} + \sqrt{q.frk} \quad (2)$$

Here, we are not going to prove the Lemma but provided in [23].

B. Diffie-Hellman Problem and Assumption

Recall that our scheme is the design to provide the mutual authentication between two users so that

attacks subjected to DH key exchange scheme, discussed in section 2, are removed. DH key exchange is based on the discrete log problem. So it is required to understand the discrete log problem and some similar related problem. In this section, we discuss the security of DL, CDH and DDH problem.

1. Discrete Log (DLG) problem: Given random integer <g, h> and large prime number p, computes α such that $g^{\alpha} = h \bmod p$.

DLG Assumption: DLG is hard to solve.

2. Computational Diffie-Hellman (CDH) problem: Given <g, g^{α}modp, g^{b}modp>, without knowing α and b, computes $g = \alpha^{b} \bmod p$.

CDH Assumption: CDH is hard to solve.

3. Decision Diffie-Hellman (DDH) problem: Distinguish (g^{α}, g^{b}, g^{ab}) from (g^{α}, g^{b}, g^{c}), where α, b and c are randomly and independent chosen.

DDH Assumption: DDH is hard to solve.

Definition 3: If one can solve the DL problem, one can solve the CDH problem. If one can solve CDH, one can solve DDH. DDH assumed difficult to solve for large p (e.g., at least 1024 bits).

C. RSA Problem (RSAP) and RSA Assumption

As we discussed earlier, our proposed scheme is based on the difficulty to break the RSA. The contribution of RSA problem and RSA assumption in our scheme plays a major role in terms of security. So, it is necessary to understand the RSA problem and RSA assumption.

RSAP: Let there are two large prime number p and q such that N=pq be an RSA modulus, $e \in Z^*_{\varphi(N)}$, $y \in Z^*_N$. From all these parameter as input, compute a such that $a = b^e \bmod N$.

Definition 4: Given input (t, ε), an algorithm A is said to solves RSAP if in t', such that t'≤ t and

$$Adv(A) = Pr[b^e = a \bmod N; (N, e) \leftarrow RSA(1^k);$$

$$y \leftarrow Z^*_n ; N \leftarrow A(N, e, y)] \geq \varepsilon$$

Where, t and ε are time and probability that an algorithm A solves the RSAP.

RSA Assumption: Given RSA problem and equation (10), it is assumed to solve RSA problem is very hard.

V. PROPOSED SCHEME: ID-BASED KEY EXCHANGE SCHEME

As we have seen, due to lack of user authentication, there are some weaknesses (man-in-middle attack, impersonation attack, replay attack etc) with Diffie-Hellman key exchange scheme. In proposed scheme [3], Nan Li implements the improved Diffie-Hellman key exchange is based on a hash function. This scheme resolves most of the attack problems using parameters (identity of parties, one time random number, and password for both parties and transformation function) and of course the services of third party known as *authentication server*. Unlike the Nan Li, Yuh-Min Tseng, et.al. proposes a mutual authentication and key exchange scheme in [24] based on bilinear pairing without uses the service of third party. This scheme enables two users with the advantage that they can mutually authenticate each other's identity while they may compute a session key. In [20] and [2], GQ and Shamir respectively proposed the identity based scheme. Both scheme based on RSA but not on bilinear pairing. Our scheme is also based on the RSA. It provides the pair of users to exchange message securely and to verify corresponding signature without communicating pair of keys.

To precede the scheme, suppose Alice and Bob, who wish to exchange key over an insecure channel. Let both agreed on public values (g and p) where g is a primitive root of prime number p. For convenience, the following notations are used to understand the scheme. Think of N= {1, 2, 3,…..}. A string means a binary string of 0 and 1. The length l of binary string is denoted $\{0, 1\}^l$. $\{0, 1\}^*$ is a binary string of infinite length. We use Z_N to denote the group {0, 1,… N-1} under addition modulo N and Z_N^* to denote the set $Z_N^* = Z_N / \{0\}$, where 0 is the identity element in the Z_N. Let φ(N) be the Euler's toteint function (the number of positive integers that are relatively prime to N). Let message m = Pub||T consist of user public parameter with timestamp, where || denote the arithmetic operator (addition, subtraction multiplication operator etc.)

Algorithm 3: An ID-based key exchange scheme.

PKG first runs the setup algorithm and then extract algorithm once the users joins the network

Setup: PKG generates an RSA modulus N and exponent e, d such that e.d=1mod φ(N). Where, d is the master secret key and (N, e) is the corresponding master public key. Choose two hash function H_1: $\{0,1\}^* \rightarrow \{0,1\}^l$ and H_2: $\{0,1\}^* \rightarrow Z_N^*$. Where H_1 and H_2 is the one-way function such as modular exponentiation, which takes two parameters, defines as $H_1(x,y) = x^y \bmod N$ and takes one parameter defines as $H_2(x)= K_x$,where, K_x is private key and l denotes the length of a plaintext.

Extract: For any user's identity ID $\in \{0,1\}^*$, the PKG calculates $K_{IDA}= H_2(ID_A)$ such that $(K_{IDA})^e= ID_A \bmod N$. Similarly, $K_{IDB}= H_2(ID_B)$such that $(K_{IDA})^e= ID_A \bmod N$. Where, K_{IDA} and K_{IDB} are the private keys of Alice and Bob respectively.

Now, we are ready to present our algorithm as shown in Figure 6.

1. Alice chooses a random integer $\alpha < p$ and $\alpha \in Z_N$. Calculates $A=g^{\alpha} \bmod p$.
2. **Sign on Alice's side**: For a message $m =<A||T_A> \in \{0, 1\}^*$, chooses T_A as a time stamp and identity ID_A, computes $t_A= a^e \bmod N$ and $S_A= a.K_{IDA}.^{H1(tA,\ m)} \bmod N$. Alice sends A, $<S_A, t_A>$, T_A and ID_A to Bob.
3. Similarly, Bob chooses a random integer $b < p$ and $b \in Z_N$, Calculates $B=g^b \bmod p$.
4. **Sign on Bob's side**: For a message $m =<B||T_B> \in \{0, 1\}^*$, chooses T_B as a timestamp and identity ID_B, computes $t_B=b^e \bmod N$ and $S_B=b.K_{IDB}^{H1(tB,m)} \bmod N$. Bob sends B, $<S_B, \mathbf{t_B}>$, T_B and ID_B to Alice.
5. **Verification on Alice's side:** By verification equation, Computes $((SA)^e)'= t_B H_2(ID_B)^{H(tB,m)} \bmod N$ with his private key (K_{IDA}), Bob identity (ID_B), t_B, Bob's public key (B) and N as input parameter. Check if $(S_B)^e$ and $((S_A)^e)'$ are equal?
6. **Verification on Bob's side:** By verification equation, Computes $((SB)^e)'= t_A H_2(ID_A)^{H1(tA,m)} \bmod N$ with his private key (K_{IDB}), Alice's identity (ID_A), t_A, Alice's public key (A) and N as input parameter. Check if $(S_A)^e$ and $((S_B)^e)'$ are equal?
7. If signature verified on Alice's side, Alice calculates the secret key, $X_A= B^{\alpha} \bmod p$.
8. And then, Alice sends a confirmation message $H_2(T_A')$ and T_A' to Bob.
9. If the confirmation is Ok, then he calculates the secret key $X_B=A^b \bmod p$.

For the correctness of DH exchange key scheme, we say, both X_A and X_B are same and shared between two users. In steps 1 and2, Alice computes his public key and generates the signature and sends to Bob. In Steps 3 and 4, Bob computes his public key and generates the signature and sends to Alice. Alice and Bob verify the corresponding signature in step 5 and 6 respectively. On successfully verification, both compute their shared secret key in steps 7 and 9.

Example: Suppose p=1259, g=187 (primitive root of prime number p=1259 has 576 primitive root they are 2, 6, 8, 10, 11, 13,……187,…), ID_A=1033, ID_B=2161. These Parameters are the public variable and known to everyone who joins the network also Alice and Bob agree on this input parameter. Now Algorithm works as follows:

Setup: PKG first runs setup algorithm and generate N which is the product of two large prime say 29 and 43 i.e. N=1247, choose e say 89 and compute d such that equation ed= 1 mod φ(N) , where d is the PKG's private-key and (N,e) is public key publically available to everyone. PKG also choose two hash functions H_1 and H_2 available to all.

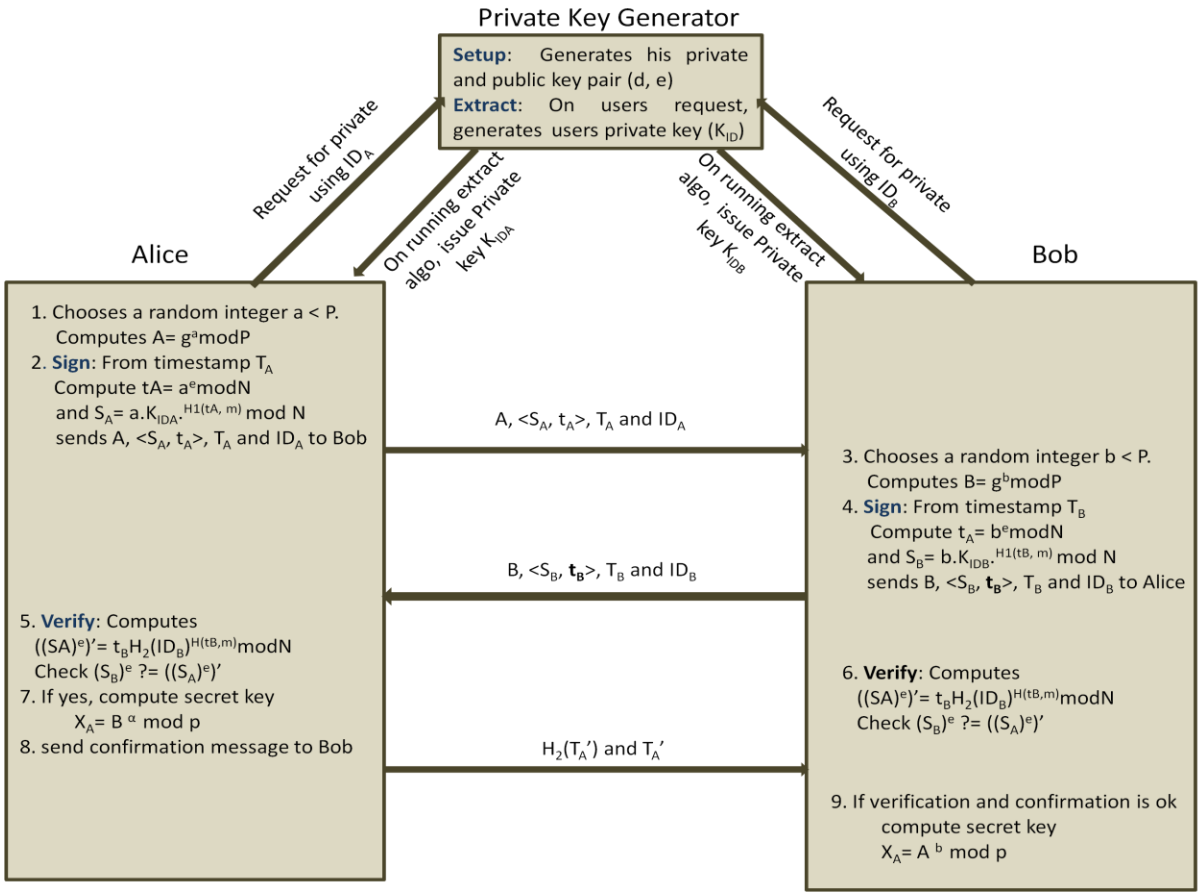

Figure 6 An ID based authenticated key exchange scheme.

Extract: In this algorithm, PKG generates private key for Alice and Bob request, Let ID_A= 1033 and ID_B=2161. By using equation 14, PKG compute Alice private key K_{IDA} = 345, 1592, 2839, 4086, 5333, 6580, 7827,… Alice may choose any from them say K_{IDA} = 2839. Similarly, PKG also compute Bob private key K_{IDB} = 772, 2019, 3266, 4513, 5760, 7007, 8254, 9501,… using equation 15. Bob may choose any from them say K_{IDB} = 4513.

Rest of the example will going as per the Algorithm 3.

Suppose, Alice chooses random integer a=983 and calculates A=1138 and computes signature <S_A,t_A =1112, 308> with time stamp, say T_A = 311 and sends the values of A, <S_A, t_A>, T_A ID_A to Bob. Similarly, Bob chooses a random integer b= 2557 and calculates B = 133 and computes signature <S_B,t_B =906, 370> with time stamp, say T_B = 6967 and send the values of B, <S_B, t_B>, T_B ID_B to Alice. On receiving the public parameter, now, Alice checks $((S_A)^e)'$= $(S_B)^e$=1060 using ID_B and B. Similarly, Bob checks $((S_B)^e)'$= $(S_A)^e$=1168 with ID_A and A. Thus, verification is done on both side, Alice and Bob can now compute shared key X_A = X_B = 412.

VI. SECURITY OF OUR SCHEME

A. Correctness

In this section, we explain the correctness of following schemes.

1). IBS Scheme Verification

In our scheme, steps 2 to 6 take the responsibility of signature generation on one side and signature verification on the corresponding side. In this section, we present the correctness of the scheme using verification condition. Let Alice generate the signature σ = <s_A,t_A> and send to Bob where, S_A=a.$K_{IDA}^{e.H1(tA, m)}$mod N and t_A=a^emod N. Bob

verify the signature using verification equation $S'^{e}=t_A H_2(ID_A)^{H1(tA,m)} \bmod N$. Because, $(K_{ID})^e = ID \bmod N$ and $t_A = a^e \bmod N$.

Therefore,

$$S'^{e} = a^e.(K_{IDA})^{e.H1(tA,m)} \bmod N$$
$$=(a.(K_{IDA})^{H1(tA,m)} \bmod N)^e$$

Because e is relatively prime to φ(n). So, e can be cancelled from exponent on both sides. Therefore,

$$S' = a.(K_{IDA})^{H1(tA,m)} \bmod N$$

Therefore, $S' == S_A$.

2). DH Key Exchange

Steps 7 and 9 in our scheme compute the secret key Sec_{AB} and Sec_{BA} with their private key and public key where, Sec_{AB} is the secret key calculated by Alice and Sec_{BA} is the secret key calculated by Bob. In this section, we mean to prove that both keys are same:

$$\begin{aligned} Sec_{AB} &= (Pub_B)^{PrA} \bmod p \\ &= (\alpha^{PrB} \bmod p)^{PrA} \bmod p \\ &= (\alpha^{Pr_B})^{PrA} \bmod p \\ &= (\alpha)^{Pr_B Pr_A} \bmod p \\ &= (\alpha^{PrA})^{PrB} \bmod p \\ &= (\alpha^{PrA} \bmod p)^{PrB} \bmod p \\ &= (Pub_A)^{PrB} \bmod p \\ &= Sec_{BA} \end{aligned}$$

B. Secure against existential forgery on adaptively chosen message and given ID

From Lemma 1 in [19], we require to prove that our scheme is secure against existential forgery on adaptively chosen message and given ID attacks. In the proposed scheme in [22], Bennain Dou, Hong Zhang, Chungen Xu, and Mu Han give the theorem which says that there is Algorithm $\mathcal{A}$ which solves RSA problem with negligible probability. Theorem 1 can be proven by using the Theorem 1 in [22]. And in rest of the theorem, we show that an adversary A cannot impersonate the second user to communicate with the first user.

Theorem 1: *For given input (t, q_H, q_E, q_S, ε), a Forger $\mathcal{F}$ break our proposed scheme under adaptively chosen message and given ID attacks in random oracle model, using algorithm $\mathcal{A}$(t', ε') and solves RSAP, where*

$$\text{adv } \varepsilon' \geq \frac{\varepsilon^2}{qH+qS} + \frac{(qH+qE+1)^4}{(qH+qS)2^{2l1}} + \frac{(qH+qS+1)^2}{2^{2k+l1+l2}} - \frac{1}{2^{l0}}$$

and

$$t' = 2t + (q_H+q_S)t_{exp} + O((q_S+q_H+q_E+1)^2).$$

Where, t_{exp} is the time to run queries.

Proof: The idea of the proof is to obtain two forgeries signature $\sigma_F = (s_F, t)$ and $\sigma'_F = (s'_F, t')$ with identity ID^A and ID^B respectively from forger F using *Forking Lemma* in [21] that satisfies $s^e_F = t_i(H_2(ID_i^A)^{ci}) \bmod N$ and $s'^e_F = t'_i(H_2(ID_i^B)^{c'i}) \bmod N$. Such that $c_i = c'_i$ if ID_i is the original identity ID*, otherwise $c_i \neq c'_i$.

Now, we are going to present the following proof: Consider a Forger $\mathcal{F}$ has an Algorithm $\mathcal{A}$ to break our proposed scheme. Given input N=p*q, $e \in Z^*_{\varphi(N)}$, $y \in Z^*_N$, $h_1,\ldots..h_{qh+qs} \in \{0,1\}^{l1}$, $\mathcal{A}$ choose an identity ID^A ,and let $H_2(ID^A)=z^e y \bmod N$ where $z \in Z^*_N$. $\mathcal{A}$ returns (N, e) and ID^A to $\mathcal{F}$. Algorithm $\mathcal{A}$ makes Table $T_1[.;.;.]$,$T_2[.]$, $T_3[.;.;.]$, $T_4[.;.;.]$ and $T_5[.]$. Where, T_1 and T_2 are capable to simulate the value of timestamp and private value r respectively, such that, $t = r^e \bmod N$. To simulate random oracle H_1 and H_2, Table T_3 and T_4 respectively are used, while T_5 assign a unique index $1 \leq i \leq q_H+q_S$ to each identity ID occurring as identity ID_i in $\mathcal{F}$'s signature query. Algorithm $\mathcal{A}$ assign index 0 to original identity ID* by setting $T_5[ID^*] \leftarrow 0$. $\mathcal{A}$ response $\mathcal{F}$'s queries as follow:

Hash function query: Denoting ID_i is the i-th queries. When $\mathcal{F}$ queries (t_i, ID_i, m_i) to H_1, $\mathcal{A}$ output the hash value of $H_1(t_i, ID_i, m_i)$, stored in Table $T3[t_i, ID_i, m_i]$ and return to F. If F queries ID_i to H_2, $\mathcal{A}$ chooses a random number $z_i \in Z^*_N$, and return $z_i^e \bmod N$ as the output of $H_2(ID_i)$. If $\mathcal{F}$ queries ID^A to H_2, $\mathcal{A}$ returns $z^e y \bmod N$ as the output of H_2 (ID^A), stored in Table $T_4[ID_i, z_i, z_i^e]$.

Extract query: Given an identity ID_i, if ID_i has been in Table T_4, $\mathcal{A}$ output z_i. Otherwise $\mathcal{A}$ runs hash query again, and then outputs z_i.

Sign query: Let ID^A be the ID if H_2 list has ID^A, and ID_i be the ID if H_2 list has no ID^A. $\mathcal{A}$ runs Extract query again. In first case, for a given ID_i, a

message m_i, and signature σ', $\mathcal{A}$ returns a signature σ. In second case, For identity ID^A, a message m_i, and signature σ', if $\mathcal{F}$ runs sign queries, $\mathcal{A}$ choose a random number $k \in \{0,1\}^l$, then compute $v= (y^{-1})^k$ mod N, $\mathcal{A}$ outputs the signature σ on message m_i by identity ID^A.

$\mathcal{A}$ receives v from $\mathcal{F}$ and searches in Table T_4 for values ID_i so that $v_i = T_4[ID_i]$. If two or more than two such values are found in $T_4[ID_i]$, then it sets $case_1 \leftarrow$ true, abort the $\mathcal{F}$'s execution and halt output (0, ε). Otherwise, A computes T and r, check whether $T_1[m_i,Pub_i]$ and $T_2[t_i]$ respectively have already been defined. If so, it checks in $T_3[t, m]$ and it set $case_2 \leftarrow$ true, aborts the execution of $\mathcal{F}$ and halts with output (0, ε).

Suppose, Prob[$case_i$] denotes the probability of the event that $case_i$ is set to true. We define the probability of accepting acc of $\mathcal{A}$ with input parameter which defines in Lemma 1 in [21] as follows:

$$\begin{aligned} acc &\geq \varepsilon - Prob[case_1] - Prob[case_2] - Prob[case_3] \\ &\geq \varepsilon - \frac{(qH+qE+1)^2}{2^{l1+1}} - \frac{1}{2^{l2}}\frac{1}{2^k}\frac{(qH+qS)}{2^k} - \frac{1}{2^{l1}} \\ &\geq \varepsilon - \frac{(qH+qE+1)^2}{2^{l1}} - \frac{(qH+qS)}{2^{2k+l2}} \end{aligned}$$

Now, we simplified the definition in the second equation. At any point in the execution of $\mathcal{F}$ two values $ID_i \neq ID'_i$ are found such that $H_1(ID_i) = H_1(ID'_i)$, so there must be at least one collision occur in H_1. All output of H_1 are uniformly taken at random from $\{0, 1\}^{l0}$, and there are at most q_H+q_S queries to H_1, the probability that least one collision occur is at most $((q_H+q_E)\ (q_H+q_E+1)/2)/2^{l1} \leq (q_H+q_E+1)/2^{l1+1}$. During i-th query, $case_2$ can be set to true, algorithm $\mathcal{A}$ first search T in T_1 with probability $1/2^{l2}$such that $T\in \{0,1\}^{l2}$, if found, search r in T_2 with probability $1/p \leq 1/2^k$, and then run H1 queries with probability $(q_H+q_S)/p \leq (q_H+q_S)/2^k$. In order to set case3=true, $\mathcal{F}$ must have predicted the private key $r \in \{0, 1\}^{l1}$ with probability $1/ 2^{l1}$. By assuming $l_0, l_1, l_2, q_H, q_E, q_S > 0$, simply rearranging the second inequality we can obtain the third inequality.

Let $\mathcal{A}$ can perfectively response $\mathcal{F}$'s queries; On $ID_i=ID^A$ $\mathcal{F}$ can give a fraud signature $\sigma_F = (s_F, t)$ with probability ε in time t', where, $t' \leq t$. Suppose there is an another algorithm B which on input ID* runs the forking algorithm FA(ID*), with probability frk return (1, (t, h, s) , (t', h', s')) where $h \neq h'$. when $\mathcal{F}$ replay, algorithm $\mathcal{B}$ uses another random oracle with identity ID^B, $\mathcal{F}$ may also have another fraud signature $\sigma'_F=(s'_F,t')$ on the same pair (ID, m) with probability ε′, such that, $t_i'= t_i$

Thus,

$$H_1(t_i,ID_i,m_i) = H_1(t_i',ID_i,m_i)$$

But,

$$H_1(ti,ID^A,m) \neq H_1(t',ID^B,m')$$

As $\sigma_F =(s_F, t)$ and $\sigma'_F =(s'_F, t')$ are valid signature, then both signature are equals.

$$s^e_F(H_2(ID^A)^{H1(t,ID,\ m)})^{-1} = s'^e_F(H_2(ID^B)^{H1(t',ID,\ m)})^{-1} = 1 \bmod N$$

Let

$$H_1(t,ID^A,m) = c$$

$$H_1(t',ID^B,m') =c'$$

$$s^e_F(H_2(ID^A)^c)^{-1}= s'^e_F(H_2(ID^B)^{c'})^{-1}= 1 \bmod N$$

$$s^e_F((z^e y)^c)^{-1} = s'^e_F((z^e y)^{c'})^{-1}= 1 \bmod N$$

$$s^e_F((z^e y)^c)^{-1}= s'^e_F((z^e y)^{c'})^{-1} \bmod N$$

$$(s_F z^{c'}(s'_F z^c)^{-1})^e=y^{c-c'} \bmod N$$

Such that $|e| > |c-c'|$, and $gcd(e, (c-c')) =1$. There exist two integer say a and b such that $ae+b(c-c')=1 \bmod N$,
We have

$$y=y^{ae+b(c-c')} = y^{ae}\, y^{b(c-c')} \bmod N$$

$$=y^{ae}\,(s_F z^{c'}(s_F z^{c'})^{-1})^{eb} \bmod N$$

$$=(y^a\,(s_F z^{c'}(s_F z^{c'})^{-1})^b)^e \bmod N$$

From the general *Forking Lemma* in [21], given $N=p*q$, $e \in Z^*_{\varphi(N)}$, and $y \in Z^*_N$, $\mathcal{B}$ can find

$$x= y^a\,(s_F z^{c'}(s'_F z^c)^{-1})^b \bmod N \qquad (3)$$

Such that $x^e=y \bmod N$ with probability

$$\varepsilon' \geq frk$$

$$\geq \frac{acc^2}{qH+qS} - \frac{1}{2^{l0}}$$

$$\geq \frac{\varepsilon^2}{qH+qS} + \frac{(qH+qE+1)^4}{(qH+qS).2^{2l1}} + \frac{(qH+qS)}{2^{2(2k+l2)}} + \frac{2(qH+qS+1)^2}{2^{2k+l1+l2}} - \frac{1}{2^{l0}}$$

$$\geq \frac{\varepsilon^2}{qH+qS} + \frac{(qH+qE+1)^4}{(qH+qS)2^{2l1}} + \frac{(qH+qS+1)^2}{2^{2k+l1+l2}} - \frac{1}{2^{l0}}$$

Where, l_0, l_1, l_2, q_H, q_E, $q_S > 0$

Now, we are ready to compute the running time of t' of the algorithm $\mathcal{B}$. First, we compute the running time t of $\mathcal{A}$. $\mathcal{A}$'s running time t is the running time of F plus time required to response (q_H+q_E+q_S) random oracle queries and (q_H+q_S) queries. Assume that t_{exp} is the time takes in exponentiation in G, and unit time takes all other operation. Each hash query and key extraction queries takes at most one exponentiation time. The $\mathcal{B}$'s running time t' is twice of the $\mathcal{A}$ plus time requires extracting x from equation no (3).Therefore, we have

$$t' = 2t + (q_S+q_H)t_{exp} + O((q_S+q_H+q_E+1)^2).$$

C. Passive attacks

In the following theorem, we show that the proposed scheme is secure against impersonate attack, replay attack and clogging attack.

Theorem 2: If an adversary $\mathcal{A}$ can guess the con b involved in the Test query with a non-negligible advantage ε', then there exist a challenger $\mathcal{C}$ to solve the CDH problem in the random oracle model.

Proof: By theorem 1, we have shown that for a given input (t, q_H, q_E, q_S, ε), a Forger $\mathcal{F}$ break our proposed scheme under adaptively chosen message and given ID attacks in the random oracle model, using algorithm $\mathcal{B}$ (t', ε'), where adv ε' $\geq \frac{\varepsilon^2}{qH+qS} + \frac{(qH+qE+1)^4}{(qH+qS)2^{2l1}} + \frac{(qH+qS+1)^2}{2^{2k+l1+l2}} - \frac{1}{2^{l0}}$, with non negligible advantage, which is a contradiction. Thus, the proposed scheme is secure against the man-in-middle attack and impersonates attack.

Theorem 3: The proposed scheme secure against replay attacks and clogging attack under the CDH problem and in the random oracle model.

Proof: A key exchange scheme is secure against replay attack and clogging attack if data transmission is not frequently delayed or repeated and recipient assures that there is no traffic in the network, respectively. On receiving the signature and timestamp, recipient confirms that the timestamp is within a limit of acceptance; otherwise dismiss the message which contains no timestamp or delivering reporting too late. After confirmation message $H_2(T_A')$ and T_A' received from Alice, Bob assures that there is no traffic in the network. Therefore, our proposed scheme is secure against replay attacks and clogging attack.

D. Other security attacks

Theorem 4: The proposed scheme provides the implicit key confirmation under the CDH problem and in the random oracle model.

Proof: A key exchange scheme offers implicit key confirmation if the second user is convinced that the first user is able to compute the sharable secret key and no one other than the two users can compute it. By theorem 1, we have shown that Alice and Bob can authenticate each other with their private key (K_{ID}=H_2(ID)) in the random oracle model under CDH assumption. By theorem 2, we have shown that no other Alice and Bob can compute the sharable secret key. Therefore, our proposed scheme provides implicit key confirmation.

Theorem 5: The proposed scheme offers forward secrecy under the CDH problem and in the random oracle model.

Proof: Key agreement scheme offers forward secrecy if any key from the long-term previous key is weaken in the future, then sharable secret key recovered from a set of long-term keys cannot be compromised. If the users random value a is compromised, then all previous secret keys cannot be compromised from the public parameter, because the adversary cannot compute $t=a^e \bmod N$, $S=a.K_{ID}^{H1(t,A\|T)} \bmod N$ and $X_A = B^{\alpha} \bmod p$. Similarly, corruption of the user itself cannot help to recover the previous sharable secret key. When the adversary $\mathcal{A}$ makes a corrupt query on $H_2(ID_C)$, the challenger returns the K_{IDC}. Theorem 2 holds under corrupt query to the adversary. Therefore, our proposed scheme offers forward secret.

Theorem 6: The proposed scheme offers non-repudiation under the CDH problem and in the random oracle model.

Proof: A protocol offers Non-repudiation, if recipient ensure that a sender cannot deny the authenticity of their signature on a message that they generate. By theorem 1, we have shown that Alice and Bob can authenticate each other with their private key ($K_{ID}=H_2(ID)$) in the random oracle model under CDH assumption. So, they can never deny the authenticity of the signature on the transcript. Therefore, our proposed scheme offers Non-repudiation.

Theorem 7: The proposed scheme offers key authentication under the CDH problem and in the random oracle model.

Proof: A key exchange protocol offers key authentication if one user is convinced that other than identified second user, no one may access to the secrete key. By theorem 1, we have shown that sender sign the message with his in the random oracle model under CDH assumption. So, the recipient can verify the message, if the message could really sign for him. Therefore, our proposed scheme offers key-authentication.

E. Dependent on Authentication Server

As identity-based signature scheme [20] is used, authentication is provided in the signature itself in terms of identity of the receiver and the private key of the sender and signature is verified by the private key of the receiver and identity of the sender. Now, the two parties need not required the service of third party. Hence, the requirement of authentication is eliminated.

Table 1 shows the comparison of Diffie-Hellman [1], Elgamal key exchange [26], NanLi [3] and our scheme with respect to some security attacking parameters.

F. Analysis of Security

In this section, we discuss the security analysis of our scheme.

- The security is based on factorizing the large integer N (product of two similar size prime, N=p*q). Peter Shor in [10] realizes that a quantum computer has a polynomial-time algorithm for factoring integers. But architect such quantum computer is very difficult, so this is safe for now.

Security Parameters	Diffie-Hellman[1]	ElGamal [26]	NanLi [3]	Our scheme
User Authentication	No	No	Yes	Yes
Entity used to authenticate the user	------	------	Authentication Server	User identity
Impersonate attack	Not secured	Not secured	Secured	Secured
Man-in-middle-attack	Not secured	Not secured	Secured	Secured
Replay attack	Not secured	Not secured	Secured	Secured
Clogging attack	Not secured	Not secured	Secured	Secured
Non-repudiation	Repudiation	Repudiation	Repudiation	Offers
Security proved by standard model	No	No	Not mention in [3]	Yes
Perfect Forward Secrecy	Offers	Not Offers	Not Offers	Offers
Implicit key confirmation	Refuse	---------	Refuse	Offers
Key authentication	Refuse	Offers	Offers	Offers
Explicit key confirmation	Refuse	Refuse	Refuse	Offers

Table 1 Comparison between our scheme and recent proposed scheme

- Security also depends on computing the e-th root modulo N. Thomas S. Messerges, Ezzy A. Dabbish, Robert H. Sloan in [11] noted that it is very difficult for adversary to find the eth root modulo N, and is not being computable in any reasonable amount of time. Thus, no one can extract e-th roots modN except the KGC and also the factorization is known only to the KGC. So far, this has been a safe and secure bet. Therefore, it is difficult for Eve to extraction of K_{ID} i.e. e-th roots modulo N by analyzing a large number of valid signatures of message of his choice. By theorem 1, we have shown that there is an algorithm which solve RSA problem with negligible advantage.
- If e is relatively primes to H_1, it is impossible to extract the private key (K_{ID}) by manipulating the verification condition. So, it is requiring making value e as large prime and H_1 a sufficiently strong one-way function.
- The value r in equation $t=r^e \text{mod} N$ should never be reused more than one or never revealed, it keeps secret to users. Unless it makes the scheme vulnerable to attack.

VII. CONCLUSION AND OPEN PROBLEM

In this paper, we propose and implement an authenticated Key exchange scheme derived from the model of Nan Li proposed in [3]. Our scheme provides the mutual authentication between two parties and proves its security in the standard model. To prove the security of our scheme, we use the *Forking Lemma* [21]. For user authentication, ID-based signature is used. Unlike of previous scheme [3], our scheme has the trusted third party (PKG) which generates the key pair (Public/Private) for every user once when the users join the network. The Public key is publicly known to everyone and a private key is known only to the owner and PKG. Thus, all user's private keys are stored at PKG. So, with users private key PKG may impersonate with other user. This is knows as Key Escrow Problem. How to construct an ID-based key exchange scheme free from key escrow problem is an open problem.

The proposed work has been a conspicuous approach towards the security aspects of secret sharing. The scheme can be further implemented by bilinear pairing. To provide the communication security over internet, transport layer security and secure socket layer are designed; our protocol can be used in transport layer security and secure socket layer.

ACKNLOWLEGE

The authors would like to thank the readers for their useful feedback, fellow honors students for their supportive nature, friends for their fruitful discussion, and my loving family.

REFERENCES

[1] W. Diffie and M. Hellman (1976), "New directions in cryptography", IEEE Transactions on Information Theory, IT-22(6), pp 644-654.

[2] A. Shamir (1984), "Identity-based cryptosystem and signature scheme", proc. Crypto 84, pp 47-53.

[3] Nan Li (2010), "Research on Diffie – Hellman Key Exchange Protocol", IEEE 2nd International Conference on Computer Engineering and Technology, Vol. No 4, pp 634 – 637

[4] Baek, J Newmarch, R Safavi-Naini and W. Susilo (2004), "A Survey of Identity-Based cryptography", in Proc. AUUG 2004, pp. 95-102.

[5] J. Menezes (2007), P. C. van Oorshot and S. A Vanstone (1997). Handbook of Applied Cryptography. CRC Press, New York, USA.

[6] Kenneth G. Paterson and Geraint Price (2003), "A comparison between traditional Public Key Infrastructures and Identity-Based Cryptography", Information Security Technical Report, Vol. 8, No. 3, pp 57-72.

[7] D. Boneh and M. Franklin (2001), "Identity-based encryption from the Weil pairing", Advances in Cryptology – CRYPTO 2001, Springer-Verlag, Vol. No 2139, pp 213-229.

[8] R.L. Rivest, A. Shamir, and L. Adleman (1978), "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". Communications of the A.C.M., Vol. No 21, issue No 2, pp 120-126

[9] U. Maurer and Y. Yacobi (1992), Non-interective public-key cryptography, Proc. Of Eurocrypto '91, Lecture Nores in Computer Sciences, Springer-Verlag, Vol. No 547, pp 498-507.

[10] P. Shor (1997), "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", SICOMP, Vol. No 26, Issue 5, pp 1484–1509.

[11] Thomas S. Messerges, Ezzy A. Dabbish, Robert H. Sloan (1999), Power Analysis Attacks of Modular Exponentiation in Smartcards", CHES'99, LNCS 1717, pp 144−157.

[12] Y. Desmedt and J. Quisquater (1987), "Public-key Systems based on the Difficulty of Tampering", Proc. of Crypto '86, Springer-Verlag, Lecture Notes in Computer Sciences, Vol. No 263, pp 111-117.

[13] U. Maurer and Y. Yacobi (1992), "Non-interective public-key cryptography", Proc. Of Eurocrypto '91,

Lecture Nores in Computer Sciences, Springer-Verlag , Vol. No 547, pp 498-507.

[14] H. Tanaka, "A realization scheme for the identity-based cryptosystem", Proc. of Crypto '87, Springer-Verlag, Lecture Nores in Computer Sciences, Vol. No 293, pp 341-349.

[15] R. Sakai, K. Ohgishi, and M. Kasahara (2001), Cryptosystems based on pairing, Proc. of SCIS '00, Okinawa, Japan, Jan. pp 26-28.

[16] Ik Rae Jeong, Jeong Ok Kwon, Dong Hoon Lee (2007) , "Strong Diffie-Hellman-DSA Key Exchange", IEEE Journals and magazines , pp. 432 - 433

[17] Harn, L., and Lin, H.-Y. (1998), "An authenticated key agreement without using one-way hash functions". Proc. 8th Nat. Conf. on Information Security, Kaohsiung, Taiwan, pp 155–160

[18] L. Harn, W. J. Hsin and M. Mehta (2005), "Autheticated Diffie-Hellman Key exchange protocol assumption", IEEE Journal and Magazines, pp 432-433.

[19] J. Cha and J. Cheon (2003),"An identity-based signature from gap Diffie- Hellman groups", In: Proc. PKC'2003, Lecture Notes in Computer Science, vol. 2567, pp 18-30.

[20] L. Guillou and J. Quisquater (1990), "A paradoxical indentity-based signature scheme resulting from zero knowledge", In: Proc. CRYPTO'88, Lecture Notes in Computer Science, vol. 403, pp 216-231.

[21] M. Bellare and G. Neven (2006), "Multi-signatures in the plain public-key model and a general forking lemma", in: Proc. ACM CCS'06, pp 390-399.

[22] Bennain Dou, Hong Zhang, Chungen Xu, and Mu Han, (2009), "Identity based sequential aggregation signature from RSA", International Journal of Innovative Computing, Information and Control, Vol. 8, pp 6401–6413.

[23] M. Bellare and G. Neven (2006),"New multi-signatures and a general forking lemma", In: preceding of the13th conference on computer and communication security-ACM CCS 2006.

[24] Yuh-Min Tseng, et al., (2007), "A mutual authentication and key exchange scheme from bilinear pairings for low power computing devices", IEEE, Computer Software and Applications Conference, COMPSAC 2007, Vol. No. 2, pp 700-710.

[25] Chen Hao and Guo Yajun (2009), "A Key Agreement Scheme Based on Bilinear Pairing for Wireless Sensor Network", Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, doi:10.1109/DASC.2009.9, pp 384-388.

[26] Taher ElGamal, *A public key cryptosystem and a signature scheme based on discrete logarithms*, IEEE Trans. Inform. Theory **3**1 (1985), no. 4, pp 469–472.