An Identity Based Key Exchange Scheme with Perfect Forward Security

Project Submitted in Partial Fulfilment

of the Requirements for the Degree of

Bachelor of Technology

in

Computer Science & Engineering

By

Sonalin Subhadarshini(111CS0446)

Under the Guidance of

Dr. Sujata Mohanty

Department of Computer Science & Engineering,
National Institute of Technology,
Rourkela-769008.

National Institute of Technology, Rourkela.

CERTIFICATE

This is to certify that the thesis entitled, *"An Identity Based Key Exchange Scheme Based upon One Round Identity Based Key Exchange with Perfect Forward Security"* submitted by *Sonalin Subhadarshini(111CS0446)* in the partial fulfilment of the requirements for the award of Bachelor of Technology Degree in Computer Science and Engineering at National Institute of Technology, Rourkela is an authentic work carried out by her under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Date:

Dr. Sujata Mohanty
Department of Computer Science & Engineering
National Institute of Technology, Rourkela.

National Institute of Technology, Rourkela

DECLARATION

I, Sonalin Subhadarshini, hereby declare that this thesis is my own and has been generated by me as the result of my own original research. I confirm that, wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research. The text of the whole thesis is generated by me and my supervisor is not responsible if any dispute may arise. Furthermore, this thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree.

Sonalin Subhadarshini
(111CS0446)
Bachelor of technology,
Dept. of Computer Science & Engineering,
National Institute of Technology, Rourkela.

National Institute of Technology, Rourkela

ACKNOWLEDGEMENT

I am indebted to my guide Dr. Sujata Mohanty for giving me an opportunity to work under her guidance. Like a true mentor, she motivated and inspired me through the entire duration of my work.

I also express my sincere gratitude to Dr. Santanu Kumar Rath, Head of the Department, Computer Science and Engineering, for providing valuable departmental facilities.

Sonalin Subhadarshini
(111CS0446)
Bachelor of technology,
Dept. of Computer Science & Engineering,
NIT Rourkela.

ABSTRACT

Identity-based authenticated key exchange protocol(IBAKE) with perfect forward security(PFS) is one of the major advancement in the field of cryptography. This protocol is used to establish secure communication between two parties who are provided with their own unique identities, by establishing their common secret keys without the need of sending and verifying their public key certificates. This scheme involves a key generation centre(KGC) which would provide the two parties involved, with their static key that can be authenticated by the parties.

Our protocol can be viewed as a variant of the protocol proposed by Xie et al. in 2012 [8]. Our protocol does not rely on bilinear pairings. We have made a comparative study of the existing protocol and the proposed protocol and proved that our protocol is more efficient. We have also provided enough proofs to verfy that our protocol is secure under attacks and is not forgeable.

Keywords: IBAKE (Identity-based authenticated key exchange protocol), PFS (perfect forward security), KGC (key generation center), static key.

CONTENTS

CHAPTER 1

INTRODUCTION

Communication in a public network needs a secure connection between the parties involved. One of the most important cryptographic methodologies that aid such a communication is the Authenticated Key Exchange (AKE) protocol. The parties communicate by establishing a secure common session key. Identity-Based authenticated key exchange (IBAKE) protocol enables to have a secure communication where parties use their identities, such as, phone numbers, e-mail addresses to establish their common secret keys. The major advantage of IBAKE is that the parties need not send their public key certificates.

Research work done so far on IBAKE [1,2,3,4,5] protocol are computationally efficient but most of the protocols do not provide perfect forward security (PFS) [6] which is one of the most desirable and important property of the AKE protocol. Some protocols provide PFS but are restricted to work on much larger groups.

In this thesis, we propose an IBAKE protocol without bilinear pairings which is secure with PFS under the Computational Diffie-Hellman (CDH) assumption in the random oracle model. We have chosen the Schnorr-like signature scheme as the base for the construction of the protocol. The protocol can be applied on any cyclic group in which the Diffie-Hellman problem is considered to be hard. This protocol is a combination of a secure IBAKE protocol having a secure ID-based signature scheme.

This thesis is organized as follows. In chapter 2, we discussed the literature covering all the significant research in this area. The construction of the protocol is discussed in chapter 3. The proposed scheme is discussed in Chapter 4. The implementation details of the above scheme are given in Chapter 5. The results and observations are given in chapter 6.The future scope and conclusion is presented in Chapter 7.

CHAPTER 2

LITERATURE SURVEY

2.1. Cryptography

Cryptography is the science of secret writing. Encryption and Decryption are the two processes involved in cryptography. The ordinary text is known as the plain text. The plain text is encrypted is called cipher text. The process of converting plain text into cipher text is known as encryption. The process of converting a cipher text back to the original plain text is known as decryption. There are three techniques of encryption and decryption- Symmetric cryptography, Asymmetric cryptography and Hashing [9, 10].

2.1.1. Symmetric key cryptography:

In this type, the secret key is common for encryption and decryption. The signer and requester use the same key by sharing it via a communication channel .The sender locks the plain text using the shared secret key, the plain text is converted into the cipher text and is sent to the receiver. The receiver receives the cipher text, converts it to the plain text using the common secret key [9, 10].

2.1.2. Asymmetric key cryptography:

In this, there are two keys involved- private key and public key. The public key is known to all the other communicating parties. The private key is secret to one party only. The sender sends the information by encrypting it using the public key of the receiver and the receiver uses its own private key to decrypt it and vice versa. This type of encryption is very widely used since it has a lot of advantages [9, 10].

2.1.3. Hashing:

Hashing is a way of converting a variable length input into a fixed length output. It uses a cryptographic hash function to perform its function. Every hash function should have three properties- pre image resistance, second pre image resistance and collision resistance [9,10]

2.2. Diffie-Hellman Key Exchange

The method of exchanging keys over an insecure channel (public channel) is known as Diffie-Hellman key exchange. It is a very old and popular method of public key exchange that was implemented and used. The Diffie–Hellman key exchange helps two communicating parties to communicate via a common secret key that is shared between them through an insecure channel. Whitfield Diffie and Martin Hellman invented this scheme in 1976. Although Diffie–Hellman key agreement helps in authenticated key exchange though it is itself non-authenticated, and it provides perfect forward secrecy in the transport layer using ephemeral keys (referred to as EDH or DHE depending on the cipher suite) [14].

Cryptographic explanation:

This protocol is implemented over a multiplicative group of integers, where *p* is prime, and *g* is a primitive root modulo *p*. The steps are:

1) Alice and Bob are supposed to choose a prime number *p* and a base generator *g* .
2) Alice chooses a secret integer a and computes A= $g^a modp$ and sends A to Bob.
3) Bob chooses a secret integer b and computes B= $g^b modp$ and sends B to Alice.
4) Alice computes s= $B^a modp$
5) Bob computes s'= $A^b modp$

s and s' have the same computed output since-

$g^{ab} modp = g^{ba} modp.$

All parameters except a and b are public. Once the common secret key is established Alice and Bob use this shared key for encryption and can send and receive messages over any insecure channel.

2.3. One Round Identity-Based Key Exchange

One round Identity based key exchange uses the identity parameters of a user, such as email addresses or phone numbers, for encryption and signature verification. This feature reduces the complexity since there is no need of sending and verifying public key certificates [9, 10].

It involves a private key generator or the key generation centre which generates the master public key and master private key before any operation.

The process of encryption and decryption proceeds as follows:

1) Alice generates the plaintext P. She uses Bob's identity IDBob and the KGC's public key pkKGC to encrypt P, generating cipher text C. Alice sends C to Bob. IDBob and pkGGC were public parameters and were already known to Alice, even before the starting of the encryption process and so there is no need of prior preparation of Bob to communicate the keys to Alice for communication.
2) Bob receives the cipher text C from Alice. It is assumed in most of the cases that C comes with a set of plaintext instructions to contact the KGC to get the private key required for decryption. Bob sends sufficient proof to KGC to authenticate itself with the KGC that IDBob belongs to him, on receiving which the KGC sends Bob's private key skIDBob to him over a secure channel. If IDBob were based on an email address, for example, the KGC sends a nonce to this email address, the successful return of which might provide an acceptable level of assurance that the owner of IDBob was the one who had contacted the KGC. This nonce can be returned through a SSL hypertext join which gave Bob a protected connection for downloading his private key. For a larger amount of certification, Bob could be obliged to present his certifications in individual and get a conservative circle containing skIDBob.
3) Bob uses his private key skIDBob to decrypt the cipher text C and recover plaintext message P.

2.4. Perfect Forward Security

Perfect Forward Secrecy is a feature in which if by any chance the private key of the server is compromised then also the session key generated is not compromised This is achieved by generating unique session keys for every session that is initiated. The main advantage us that even if the session key of the current session is compromised then the data that is exchanged in that session might be compromised but the subsequent sessions use different keys so that information exchange is still secure.

With Perfect Forward Secrecy, for each session a new set of Diffie-Hellman parameters are generated and both communicating parties create a new shared secret key that is unique and hidden from any attacker. [13].

2.5. Computational Diffie-Hellman Assumption

In a cyclic group, a certain computational problem is hard. This is implid by the computational Diffie-Hellman assumption (CDH assumption).

Considering a cyclic group G of order q. According to the CDH assumption that if we are given (g, g^a, g^b) for a generator g that is randomly choosen and random $a, b \in \{0, \ldots, q-1\}$, it is not possible to compute in polynomial time the value- $\boldsymbol{g^{ab}}$ [11].

CHAPTER 3

PRELIMINARY

The schnorr-like signature scheme is the basis of our protocol. In this section we discuss the schnorr-like signature in detail [7].

3.1. Schnorr-like signature scheme

This includes four algorithms:

1) Generation of global parameters:

   -Input: k (a secure parameter selected randomly)

   -Output:

   a) Master secret key (MSK) = z

   where z is a random element from Z_q

   b) Master public key (MPK) = (G, g, q, g^z, H, H')

   where G = generation algorithm

   H’:$\{0,1\}^*$-> Z_q

   H:$\{0,1\}^*$-> Z_q

2) Key-extraction algorithm:

- Input: MSK, MPK and ID of the user

-Output: a) static key $r_{ID} = g^k \bmod p$ = (y , g^r) where r is a random element from Z_q

b) $Y = r + z.H(g^r, ID)$

3) The signing algorithm:

- Input: MPK, s_{ID}, message m

-Output: signature $S = (g^a, b, g^r)$

where a is a random element from Zq

$b = a + y.H^{'}(ID, g^a, m)$

4) Verification algorithm:

-Input: MPK, S, m, ID

-Output: $g^b = g^a (g^r g^{zc})^d$, the output shows if this equation is true or not

where $C = H(g^r, ID)$

$d = H^{'}(ID, g^a, m)$

This scheme is secure against existential forgery on adaptive chosen message and adaptive identity attacks under the discrete logarithm assumption in the random oracle model [7].

CHAPTER 4

THE PROPOSED SCHEME

4.1. The Proposed Scheme:

The following are the major changes that in the existing scheme [8] in order to propose our new scheme:

1) Use of a single hash function.
2) The computation of s_{ID}.
3) Verification of the received static key by the KGC.
4) In the protocol running, the computation of the signature and the verification has also been modified.
5) The number of exponentiation has been reduced in case of the verification algorithm.

The proposed scheme consists of three participants, namely, KGC, signer and requester. We used five modules: Mater public key generation, public key generation, static key generation, verification of static key, key exchange. Each module is described as below.

CHAPTER 5

IMPLEMENTATION OF PROPOSED SCHEME

5.1. Details regarding Implementation:

- Language used: Java
- Hash Function: SHA-1
- Number of users communicating: 2
- Static key is generated by the module “static key generation”
- Running environment: my eclipse
- Operating system: windows 7(64 bit)
- RAM specification: 4GB

Master public key generation:

Values of Parameters: *g=2, w=31, p=97*

Public key Generation:

In this module we generate the private keys for A an B (the two users who are communicating).

Values of Parameters: x_a (private key of A) = 54, x_b (private key of B) = 65, g=2 , p=97

Static key generation:

The static key $t_{ID} = (s_{ID}, r_{ID})$ is calculated using the following parameters as input:

k_a =*23*, Identity of A=12

k_b =67, Identity of B=14

Static Key Verification:

The user can authenticate the static key generated by the KGC. The value of authentication is calculated using equation- $S_{ID} = H(ID,(r_{ID}.W^{r_{ID}})^{x_{user}} \bmod p)$ and output is shown in the snapshot of implementation.

Key Exchange protocol:

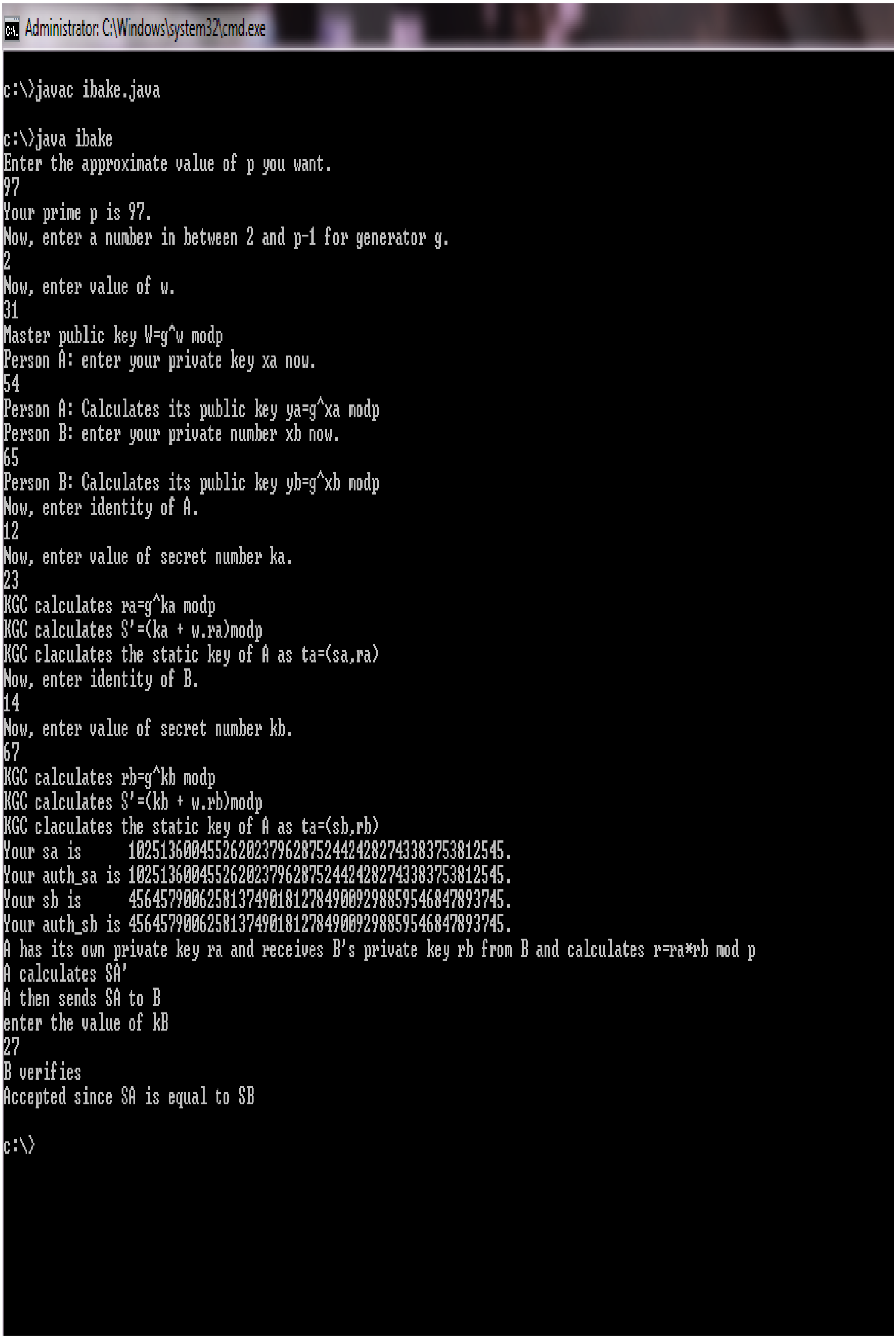
Value of parameters: $k_B = 27$

S_A' and S_B' are calculated using -

$$S_A^{'} = H(ID, (r_B . y_b^{'r})^{x_a} \bmod p)$$
$$S^{"} = (k_B + r.x_B \bmod p) \bmod p$$
$$S_B^{'} = H(ID, y_a^{S^{"}} \bmod p)$$

respectively.

5.2. Snapshot of the implementation:



```
c:\>javac ibake.java

c:\>java ibake
Enter the approximate value of p you want.
97
Your prime p is 97.
Now, enter a number in between 2 and p-1 for generator g.
2
Now, enter value of w.
31
Master public key W=g^w modp
Person A: enter your private key xa now.
54
Person A: Calculates its public key ya=g^xa modp
Person B: enter your private number xb now.
65
Person B: Calculates its public key yb=g^xb modp
Now, enter identity of A.
12
Now, enter value of secret number ka.
23
KGC calculates ra=g^ka modp
KGC calculates S'=(ka + w.ra)modp
KGC claculates the static key of A as ta=(sa,ra)
Now, enter identity of B.
14
Now, enter value of secret number kb.
67
KGC calculates rb=g^kb modp
KGC calculates S'=(kb + w.rb)modp
KGC claculates the static key of A as ta=(sb,rb)
Your sa is      102513600455262023796287524424282743383753812545.
Your auth_sa is 102513600455262023796287524424282743383753812545.
Your sb is      456457900625813749018127849009298859546847893745.
Your auth_sb is 456457900625813749018127849009298859546847893745.
A has its own private key ra and receives B's private key rb from B and calculates r=ra*rb mod p
A calculates SA'
A then sends SA to B
enter the value of kB
27
B verifies
Accepted since SA is equal to SB

c:\>
```

CHAPTER 6

RESULTS AND OBSERVATIONS

6.1. SECURITY ANALYSIS

In this section, we first show the security assumptions of the proposed scheme. Then we discuss the correctness of the scheme.

1) The proposed scheme satisfies unforgeable property

   In order to forge the static key, the adversary has to know the values of k_a, $k_{b.}$ To forge the signature, x_a and x_b are required. All these values are protected under the DLP assumption.

2) The authenticity of the static key obtained by a communicating party can be verified using $S_{ID} = H(ID, (r_{ID}.W^{r_{ID}})^{x_{user}} \bmod p)$

   Proof:

$$s_{ID} = H(ID, y_{user}{}^{s'} \bmod p)$$
$$y_{user}{}^{s'} = g^{x_{user}}$$
$$= y_{user}{}^{(k+w.r_{ID})}$$
$$= g^{x_{user}{}^{(k+w.r_{ID})}}$$
$$= g^{kx_{user}}.g^{w.r_{ID}x_{user}}$$
$$= r_{ID}{}^{x_{user}}.w^{r^{ID}x_{user}}$$
$$= (r_{ID}.w^{r_{ID}})^{x_{user}}$$

3) The signature can be verified using (8) which is indeed correct.

   Proof:

$$y_a{}^{s"} = y_a{}^{k_B+rx_B}$$
$$y_a{}^{s"} = g^{x_A(k_B+rx_B)}$$
$$= g^{x_Ak_B+rx_Ax_B}$$
$$= (g^{k_B})^{x_A}.(g^{x_B})^{rx_A}$$
$$= (r_B)^{x_A}.(y_B)^{rx_A}$$
$$= (r_B.y_B{}^{r})^{x_A}$$

4) Security of the protocol under PFS(Perfect Forward Security):

Case 1: When r_a and r_b are known to the adversary:

Proof: If an adversary gets to know the r_A and r_B of one session, for the next session new values of r_A and r_B are calculated as-

$$r_A = g^{k_a} \bmod p$$

$$r_B = g^{k_b} \bmod p$$

Where k_a and k_b are randomly selected and both belong to Z_q^*.

Thus, the adversary can never use the known values of r_A and r_B in the next session and thus all the sessions are secure.

Case 2: When x_a or x_b is known to the adversary:

Proof: In our protocol we calculate S_A' as-

$$S_A' = H(ID, (r_B . y_b{}^r)^{x_a} \bmod p)$$

Even if the private keys are known to the adversary, the adversary would still need the value of r to get S_A'. For each session r is calculated as-

$$r = r_A . r_B \bmod p$$

Thus knowing x_a would not reveal the session-key.

We observe from case 1 and case 2 that the shared secret key (S_A', S_B') is not transmitted and both are calculated at each end separately. We can thus say that our protocol provides perfect forward security.

6.2. PERFORMANCE EVALUATION

We compared the performance of the proposed scheme with the existing one [8] considering the number of exponentiations and the result is shown in the following table-

	#E-e	#E-sk	#E-S	#E-V	PFS
MIBE protocol	1	2	1	3	YES
Proposed protocol	1	2	2	1	YES

#E-e and #E-sk stand for the number of exponentiations done for ephemeral key and static key respectively. #E-S and #E-V stand for the number of exponentiations done for the signing and verifying.

It is observed from the table that in case of MIBE protocol the number of exponentiations done for verification is 3 while that of the proposed protocol is only 1.

CHAPTER 7

FUTURE WORK AND CONCLUSION

In this thesis, we proposed a modified IBAKE scheme that allows two parties to communicate without sending and verifying their public key certificates. The proposed protocol has been proved to be unforgeable and is also secure against attacks such as chosen cipher text attack. The scheme also provides perfect forward security which makes this protocol secure and efficient. The proposed protocol is computationally cost effective.

The proposed protocol can be successfully implemented in an electronic communication which involves two parties. This protocol can be implemented in a micro-payment scheme where the protocol would aid to authenticate the customer and the merchant before any transaction occurs.

BIBLIOGRAPHY

[1] Liqun Chen, Caroline Kudla, Identity based authenticated key agreement protocols from pairings, in: Proc. 16th IEEE Security Foundation Workshop, IEEE Computer Society Press, 2002, pp. 219-233.

[2] Colin Boyd, Wenbo Mao, Kenneth G. Paterson, Key agreement using statically keyed authenticators, in: Markus Jakobsson, Moti Yung, Jianying Zhou (Eds.), ACNS 04: 2nd International Conference on Applied Cryptography and Network Security, Yellow Mountain, China, June 8-11, 2004, in: Lecture Notes in Computer Science, vol. 3089, Springer, Berlin, Germany, 2004, pp. 248-262.

[3] Colin Boyd, Kim Choo, Security of two-party idebtity-based key agreement, in: E Dawson, S. Vaudenay (Eds.), Progress in Cryptology- Mycrypt 2005, Malaysia, Kuala Lumpur, Springer, Berlin, Heidelberg, 2005, pp. 229-243.

[4] Liqun Chen, Zhaohui Cheng, Nigel P. Smart, Identity-based key agreement protocols from pairings, International Journal of Information Security 6 (4) (2007) 213-241.

[5] Shengbao Wang, Zhenfu Cao, Feng Cao, Efficient identity vbased authenticated key agreement protocol with PKG forward secrecy, International Journal of Network Security 7 (2) (2008) 181-186.

[6] Dario Fiore, Rosario Gennaro, Making the Diffie-Hellman protocol identity-based, in: Joseph Pieprzyk (Ed.), Topics in Cryptography – CT-RSA 2010, San Francisco, CA, USA, March 1-5, 2010, in: Lecture Notes in Computer Science, vol. 5985, Springer, Berlin, Germany, 2010, pp. 165-178.

[7] David Galindo, Flavio D. Garcia, A Schnorr-like light weight identity-based signature scheme, in:Bart Preneel (Ed.), Topics in Cryptography- CT-RSA 2010, San Francisco, CA, USA, March 1-5, 2010, in:Lecture Notes in Computer Science, vol. 5985, Springer, Berlin, Germany, 2010, pp.. 165-178.

[8] Min Xie, Libin Wang, One-round identity-based key exchange with Perfect Forward Security, School of Computer, South China Normal University, 510631, PR China, 2012.

[9] William Stallings, "Cryptography and Network Security: Principles and Practice", Prentice Hall, 4th Edition, 2005.

[10] Behrouz A. Fourazan, Debdeep Mukhopadhyay, "Cryptography and Network Security", Tata McGraw Hill, 2nd Edition, 2010.

[11] W. Diffie and M.E. Hellman, " New Directions in Cryptography." IEEE Transactions on Information Theory , 22(5):644-654, 1976.

[12] Bellare, M., Rogaway, P., One round identity based key exchange, Advances in

Cryptology,- CRYPTO’93, Lecture Notes in Computer Science , Springer-Verlag, Vol 773 (1994) 232-249

[13]. A. Menezes, P. vanOorschot, and S. Vanstone, Handbook of Applied Cryptography, chapter 12, CRC Press, 1996.

[14] W. Diffie, M. Hellman: New directions in cryptography, IEEE Trans. Info. Theory IT-22 November (1976)644-654