Volume 5, Issue 1, January 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Survey of Zero-Knowledge Proof for Authentication

Jitendra Kurmi*
Scholar
Department of Computer Science & Engg.
Lovely Professional University
Phagwara, Punjab, India

Ankur Sodhi
Assistant Professor
Department of Computer Science & Engg.
Lovely Professional University
Phagwara, Punjab, India

Abstract - Zero-knowledge proofs are cryptographic protocols which do not disclose the information or secret itself during the protocol. Zero-knowledge proofs plays an important role in the design of cryptographic protocols. The application of Zero-knowledge protocols can be in authentication, identification, key exchange and other basic cryptographic operations. Zero-knowledge proof has been implemented without expose any secret information during the conversation and with smaller computational requirement than using comparable public key protocols. The most cryptographic problems can be solved with the help of zero-knowledge protocols, as well as with cryptography. Zero-knowledge protocols can be a best solution in many occasions. The Zero-knowledge proof protocols are very lightweight, due to which it requires less amount of memory. Thus Zero-knowledge protocols widely used especially in authentication. This paper presents an overview of zero-knowledge protocol used for authentication, identification and key exchange.

Keywords - Proof of knowledge, Zero knowledge, Digital identification, Password Authentication, P2P Identity Authentication, key exchange, RFID, public key encryption, pseudo random number generator.

I. INTRODUCTION

Zero-knowledge proofs and proofs of knowledge are fundamental notions and powerful tools in cryptography. In a zero-knowledge proof system [1], a prover convinces a verifier that some statement is true while leaking nothing but the validity of the assertion. In a proof of knowledge, the prover also convinces the verifier that he indeed knows a satisfying "witness" for the given statement. These proof systems are the building blocks in many cryptographic constructions[2]. A zero-knowledge proof is a proof of some statement which reveals nothing other than the veracity of the statement[10]. The word "proof" here is not used in the traditional mathematical sense. Rather, a "proof", or equivalently a "proof system", is an interactive protocol by which one party (called the prover) wishes to convince another party (called the verifier) that a given argument is true. In zero-knowledge proof, the prover proves that he/she knows a secret without revealing it[5].

Researches in zero-knowledge proof has been prompted by authentication systems where one party wants to prove its identity to a second party via some secret information (such as a password) but doesn't want to disclose anything about this secret to the second party. This is called a "zero-knowledge proof of knowledge". Even if a password is typically too small or insufficiently random to be used in many system for zero-knowledge proofs of knowledge[9].

One of the most absorbing uses of zero-knowledge proofs with in cryptographic protocols is to apply honest behaviour while maintaining secrecy[3]. Roughly, the idea is to enforce a user to prove that its behaviour is correct according to the protocol. Because of soundness error we know that user must really act honestly in order to be able to provide a valid proof[14]. Because of zero-knowledge, we know that the user does not agree with the privacy of its secrets in the process of providing the proof.

A. Definition of Zero Knowledge Proof

ZKP model of computational defined as an interactive proof system (P,V), where P is a prover and V is a verifier. Protocol (P,V) is for proving a language membership statement for a language over { 0,1 }.

Let a L be a language over {0,1}*, for a membership instance x ϵ L, P and V must share the common input x, proof instance is denoted as (P,V)(x).

P and V are linked by a communication channel over which exchange a sequence, called proof transcript a_1, b_1, a_2, b_2…..a_n, b_n. Proof transcript interleaves prover's transcript and verifier transcript. Each element a_i, b_i exchange is bounded by polynomial time in |x| and proof instance (P,V)(x) must terminate in polynomial time in |x|. Upon completing the interaction, the output of the protocol should be of form (P,V)(x)ϵ{Accept, Reject} representing V's acceptance or rejection of P's claim that xϵL.

Three properties are expected from a zero-knowledge proof[1]:

1. *Completeness:* An interactive proof (protocol) is complete if, given an honest prover and an honest verifier (that is one following the protocol), the protocol succeeds with overwhelming probability (i.e. the verifier accepts the prover's claim).

2. *Soundness:* An interactive proof protocol is sound if there exit an excepted polynomial time algorithm M with the following property: if a dishonest prover (impersonating p) can with a probability, successfully execute the protocol with V, then M can be used to extract from this prover knowledge (essentially equivalent to P's secret) which with irresistible probability allows successful resultant protocol executions.
3. *Zero-knowledge:* A protocol has zero-knowledge property if it has the following sense; there exists an expected polynomial time algorithm (simulator) which can produce, upon input of the statement to be proven but without communicating with the real prover, transcripts indistinguishable from those resulting from interaction with the real prover.

II. ZERO-KNOWLEDGE PASSWORD AUTHENTICATION PROTOCOL

The simple version of the algorithm provides only one way authentication, that is, only server can authenticate a client system[8]. Let us designate the server and client as verifier and prover for ease of understanding.

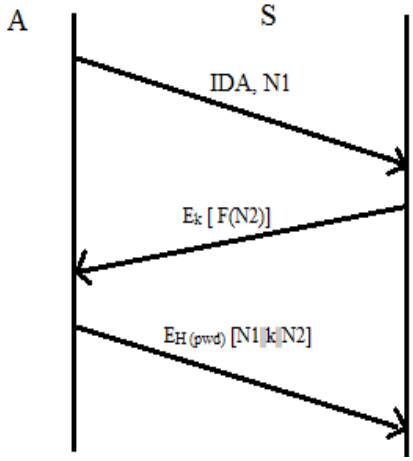

Fig. 1. Zero Knowledge Password Authentication Protocol

Notations used:
IDA: Username of A
N1 & N2: Nonce
K: Shared secret key between A(user) & (server)
F: Transformation function
E_k: Encryption using key k
H[pwd]: Hash of the password

A. Algorithm for Zero Knowledge Password Authentication Protocol

1. The prover sends his username and a challenge (nonce) N1 to the verifier.
2. The verifier responds by generating a random session key, say k and another challenge (nonce) N2. Then it concatenates N1, N2 & k and encrypts them using the hash of the password corresponding to the receiver user name. This encrypted data is then sent to the prover.
3. The prover now decrypts the data using the hash of its password as key, fetches the values of N1, N2 & k and verifies if the value of N1 received is same as the one it had sent to the verifier.
4. The nonce N1 here is used only to avoid any replay attack. If the value of the received & the generated nonce do not match, then the received message is discarded else it retrieves the session key.
5. The prover then applies the transformation function on the nonce N2, encrypts it with the received session key and sends it to the verifier.
6. Once the verifier receives the encrypted message, it then decrypts the message with the generated session key and matches it with the expected value. If match occurs, then the user is allowed to login to his account and access resources else access denied.

The main advantages of this protocol are as follows:

1. The authentication is done without the need of the password to travel across the wire.
2. The password on the server is stored in encrypted format thus making it less vulnerable to attacks.
3. The security of protocol mainly depends on the strength of the encryption algorithm being used. Thus using the standard algorithm like AES, DES etc. will provide high degree of security to the protocol.
4. Use of nonce at each step helps us to prevent replay attacks.

B. Zero knowledge Password Authentication Protocol with Public key encryption

This section deals with the other version of the zero knowledge protocol describe above. The version of the protocol makes use of public key encryption in order to give an added of security and also enable two-way authentication i.e. the verifier (server) can authenticate the prover (client) and vice versa[8].

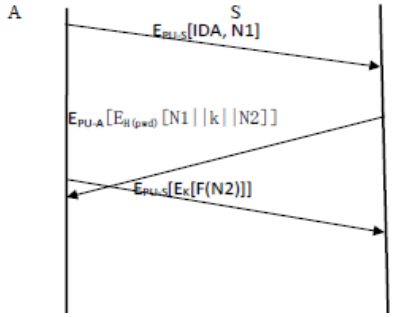

Fig. 2. Zero knowledge password authentication protocol with PKE

Notations used:
IDA: Username of A
N1 & N2: Nonce
K: Shared secret key between A (user) & S(server)
F: Transformation function
E_k: Encryption using key k
H[pwd]: Hash of the password
E_{PU_S} & E_{PU_A}: Encryption using public key of S & A respectively

C. Algorithm for Zero Knowledge Password Authentication Protocol with PKE

1. The user, say A sends his username and a nonce to the server after encrypting it with server's public key.
2. The server decrypts the message with his private key and extracts the value of the nonce N1.
3. The server then generates a nonce N2 and a random session key k, concatenates N1, k & N2 encrypts them with hash of the password of the user A, then with public key of the user A and sends the encrypted data to A.
4. User A then decrypts the received encrypted data with his private key, then with the hash of his password and extracts the values of N1, N2 & k. He then matches the value of received nonce N1 & the generated value of N1.
5. If match occurs, then A extracts the value of k & nonce N2, applies the transformation function F on N2 and encrypts the transformed value first with the session key k, then with public key of the server and sends the encrypted message to the server.
6. The server decrypts the received value with its private key & then with the shared session key.
7. The user A is allowed to login if the server receives the expected value else access is denied.

III. A P2P IDENTITY AUTHENTICATION BASED ON ZERO-KNOWLEDGE

The identity authentication scheme includes two phases[13]: the pre-processing before authentication and the phase of identity authentication.

1. The phase of pre-processing before authentication. During the early stage of establishing the system, each node goes to the management centre for registration and the management centre will allocate to each node a unique identity, public key and private key and so on.
2. The phase of identity authentication. It is the process of identity authentication among nodes by using the new zero-knowledge identity authentication method. During this process, the management centre is not required to be involved.

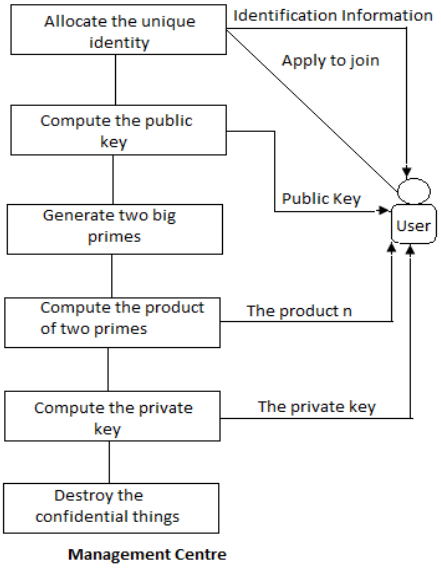

Fig. 3. System Establishment Process

A. Pre-processing before authentication

Each nodes goes to the management centre for registration during which the management centre entities each node with a unique ID used to represent each user[12].

1. The management centre uses a one to one function h(x) to calculate the node's public key, e:
   h(ID) = e
2. The management centre first generates two big primes: p and q, then computes their product according to the formula.
   n = p*q, ϕ(n) =(p-1)*(q-1)
3. Then, it calculate the private key s, according to the following formula: $s^2 = e^{-1} \bmod \phi(n)$
4. Finally, the server sends the ID, the public key (e), the private key (s) and the product (n) of the two big primes to the node.

B. Identity Authentication Process

Assuming that P and V is performing identity authentication with V as the verifier and P as the requester, the process is as in fig 4:

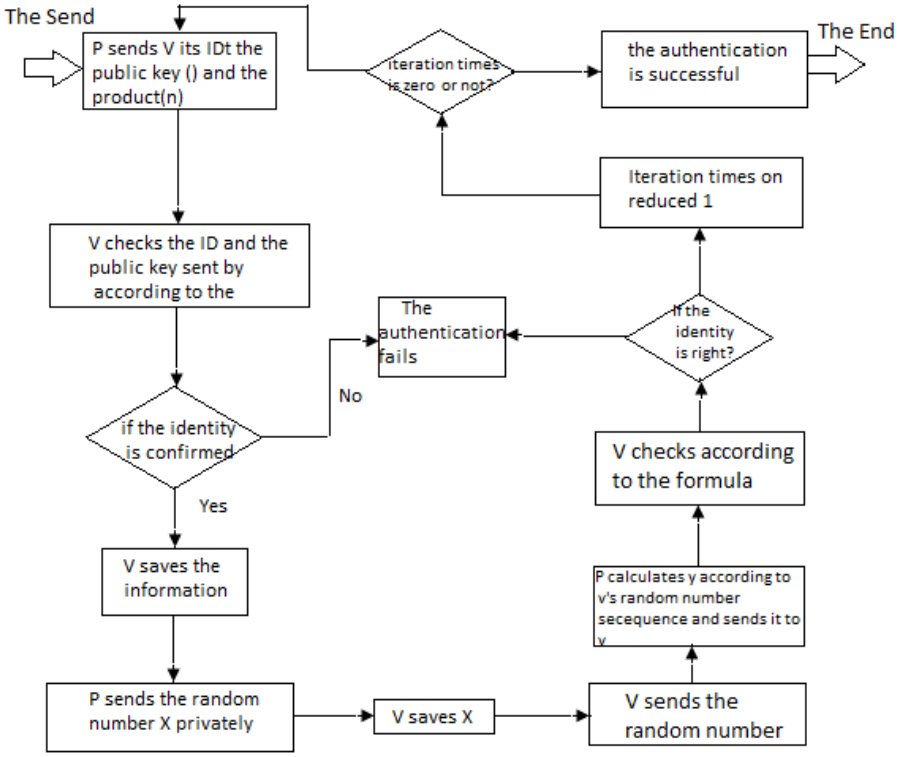

Fig. 4. Identity Authentication process

1. P sends V its ID, the public key (e) and product (n).
2. V checks the ID and the public key (e) sent by P according it the one to one function. If they are one to one correspondence, then save the ID, the public key (e) and the product (n). If the identity conforms, then continue, otherwise go to 9.
3. P randomly chooses an integer (r), 1<r<n, then calculates X according to formula and sends the result to V.
4. V saves X and randomly chooses an integer, b, b $\epsilon\{1,2,\ldots n\}(n\rightarrow\infty)$ and secretly sends it to P.
5. P calculates y according to formula (5) and V's random number sequence, then sends y to V.
   $y=r*b^{s2} \bmod n$
6. V checks whether X and Y means the formula (6). If it means, then continue, otherwise go to 9.
   $X = y^e * b^{-1} \bmod n$
7. Subtract 1 from the number of iterations and if it becomes zero, then the authentication is successful, otherwise continue.
8. Repeat step 1 to 9.
9. The authentication fails and the access is denied.

IV. MODIFICATION OF DIFFIE-HELLMAN KEY EXCHANGE ALGORITHM FOR ZERO-KNOWLEDGE PROTOCOL

The proposed ZKP based on D-H key exchange algorithm in the sense that both parties (the prover and the verifier) exchange non secret information and did not revealing secrets to get one identical secret key[17]. This means that the prover can prove to the verifier that he knows the secret. The basic D-H key exchange algorithm which is vulnerable to man-in-middle-attack. The proposed version has been developed to resists the man-in-middle attack.

A. Proposed ZKP based on Diffie-Hellman Key Exchange algorithm

To protect the proposed algorithm from the man-in-middle attack an encrypted replies (R1 and R2), and mutual authentication between the prover (Alice) and the verifier (Bob) is required[15].

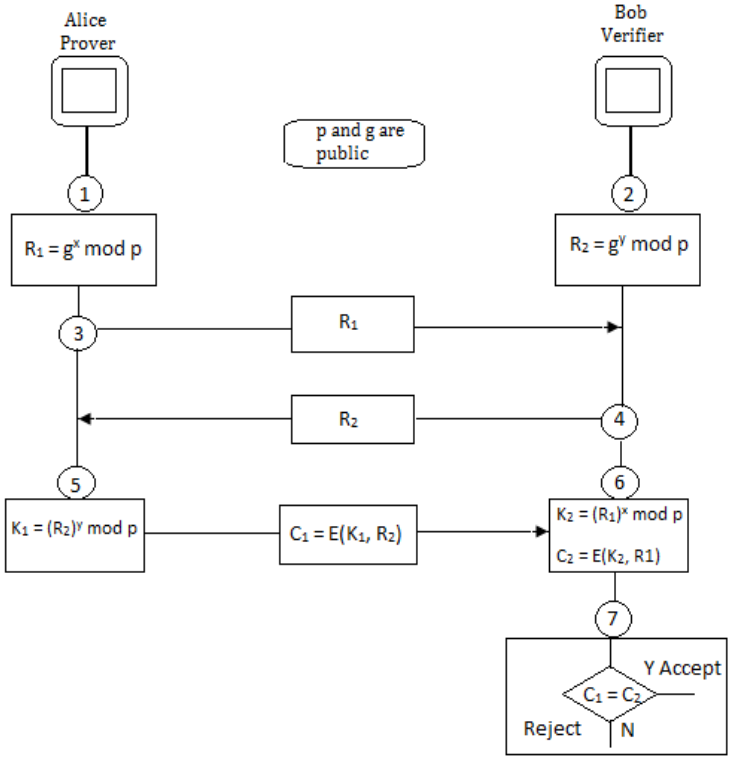

Fig. 5. Basic key exchange algorithm with man in middle attack

The prover (Alice) proves to the verifier (Bob) that she knows a secret by calculating the key (K) and resend Bob's reply (R2) to the verifier (Bob) encrypted with the generated secret key (K). Bob will encrypt his own reply (R2) with the generated secret key (K) and match the two encrypted information, if matched then Alice is verified, otherwise it is rejected. The verifier also needs to prove to the prover that he is honest by sending his reply R1 together with encrypted R1, then the verifier decrypt R1, by his key and match R1 and R1', if they matched then the verifier is honest. Fig shows the procedure of the proposed protocol. The protocol performed as follows:

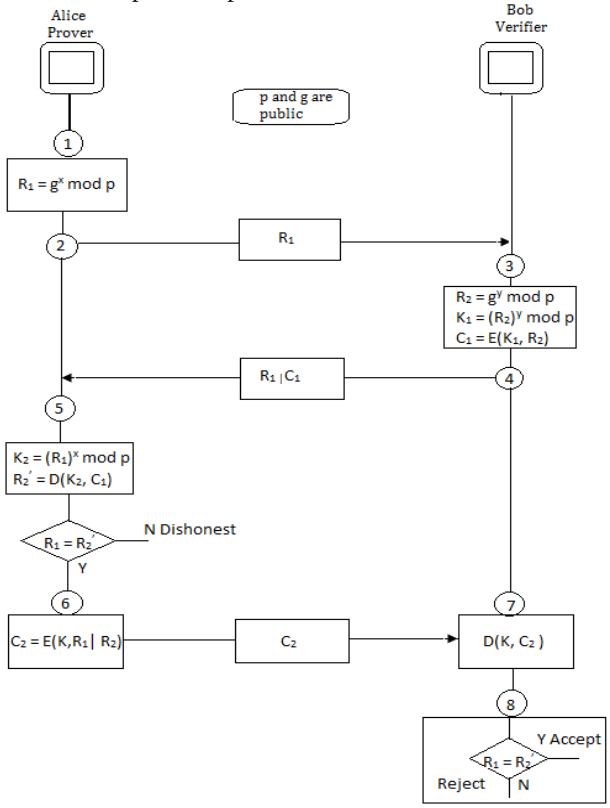

Fig. 6. Proposed Zero Knowledge Protocol for key exchange

1. Alice (the prover) chooses a large random number x, such that 0<x<p and calculate $R1 = g^x \bmod p$
2. Alice sends R1 to Bob.
3. Bob (the verifier) chooses another large random number y, such that 0<y<p and calculate $R2 = g^y \bmod p$, $K_{Bob} = (R1)^y \bmod p$, and $C1 = E(K_{Bob}, R2)$.
4. Bob sends (R2|C1) to Alice.
5. Alice, calculate $K_{Alice} = (R2)^x \bmod p$, decrypt (R2' = $D(K_{Alice}, C1)$) and verify (R2=R2'). If they matched then she proceeds; otherwise the verifier is dishonest.
6. Alice encrypts ($C2=E(K_{Alice},R1|R2)$ and send it to Bob.
7. Bob decrypt C2to get R1' and R2'.
8. Bob verify (R1=R1'); if they are equal then Alice is verified (Accepted), otherwise it is a dishonest prover (rejected).

V. HASH – BASED SECURE ACCESS CONTROL SCHEME FOR MOBILE RFID SYSTEM USING ZERO-KNOWLEDGE

Due to daily increasing popularity of mobile RFID-based application because of the widely use of RFID technology and mobile smart device, which makes its a new hot zone for research and development. Due to portable nature of an RFID mobile device, as well as use of wireless or internet connections for communications. Mobile RFID systems have a specific security concerns that need to be addressed[16]. To deal with above discuss problem an authentication protocol based on zero-knowledge protocol and hashing in mobile RFID is proposed in which readers must be authenticated and hence only valid readers will be allowed to read the tags[18].

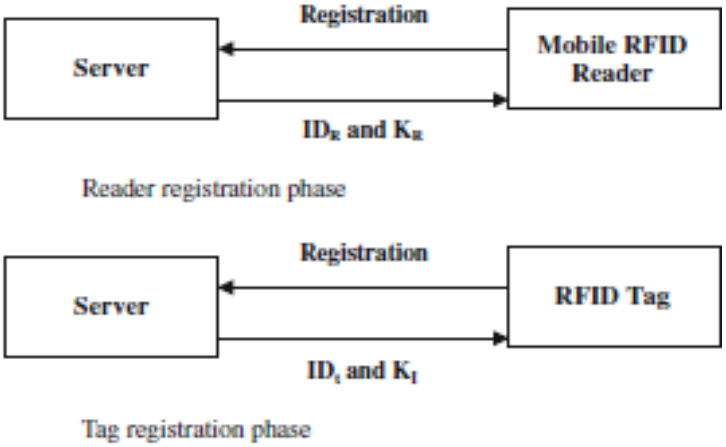

Fig. 7. Query and authentication process

Notations of proposed protocol

Symbol	Meaning
ID_t	Unique Identifier of the tag
ID_R	Unique Identifier of the reader
K_I	Secret key shared between the tag and the server
K_R	Secret key shared between the reader and the server
s	Session key shared between the reader and the server
||	Concatenation
⊕	XOR operation
P	prime number
r1	A random number generated through the use of a Pseudo Random Number Generator (PRNG) with in the server
r2	A random number generated through the use of a Pseudo Random Number Generator (PRNG) with in the reader
K_{I+1}	Updated secret key used in between the tag and the server.
g	A function used by the reader and server.
D	Detailed information about the tag in the database.

A. Registration Phase:-

In this phase, all tags and mobile FID readers must initially register with the server. The mobile RFID reader has to register with the server and obtain a user name and a password shown in fig 7. The tag has to register with the server and the registration phase in between the server and the tag.

B. Authentication Phase:-

This phase provides an information about how to perform the mutual authentication and verify whether user is legal or not[12] shown in fig 8.

1. The reader queries the tag and forwards its identity information to the tag.
2. After receiving the query message the tag computes $H(ID_t \oplus K_i)$ and forwards it to the reader.
3. The reader forwards the message $H(ID_t \oplus K_i)$ concatenated with its identity information ID_R to the server.

4. The server checks whether H($ID_t \oplus K_i$) forwarded by the reader matches with the stored hash code of the tags. If it matches then the database authenticates the tag as a legitimate one. The server verifies whether the reader is an authenticated one using the reader ID and added, it also uses zero-knowledge protocol to avoid fake readers. The server and reader both generate a function $g = (K_R)^2 \bmod p$. K_R is the secret key known between reader and server, p is some number. The server chooses a random number r1 and computes $A = g^{r1} \bmod p$. The server forwards the value A and p to the reader.
5. Similarly the reader chooses a random number r2 and computes $B = g^{r2} \bmod p$. The reader forwards the value B to the server. The server and the reader both generates a session key s and there will not be any transfer of it. The server then computes the session key $s = B^{r1} \bmod p$ and similarly the reader also computes the session key $s = A^{r2} \bmod p$. The session key s generated by both the server and the reader will be the common one.

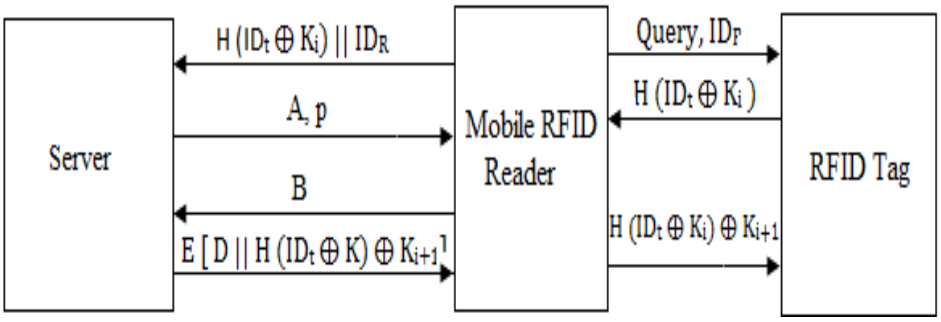

Fig. 8. Query and authentication phase

6. The server updates the confidential information Ki to K_{i+1} using a pseudo random number generator where K_{i+1} =PRNG(Ki). The server computes H(IDt ⊕ Ki) and operates XOR algorithm with K_{i+1}. Thus message along with the detailed information of tag D is forwarded in encrypted from to the reader using the session key s.
7. The reader decrypts the information using the session key s and obtains the information about the tag. The reader forwards the remaining information to the tag. The tag obtains the information and computes K_{i+1} by performing the XOR operation of the information obtained from the reader with $H(ID_t \oplus Ki)$ and then updates the secret key information Ki to K_{i+1}.

VI. CONCLUSION

This paper presents a view of zero knowledge protocol used in authentication, identification and key exchange. Zero-knowledge protocols are probabilistic proofs because there is some small probability (soundness error) that allows a cheating prover to convince the verifier of a false agrument. In section II to V we have discussed ZK-PAP and ZK-PAP with PKE protocols, both of which are based on the concept of zero-knowledge proof. The ability to authenticate oneself without having to reveal one's password will make the system less vulnerable to attacks. Also using the public-key encryption in ZK-PAP with PKE adds a second level of security and enables mutual authentication between the client & server. We have discussed a P2P identity authentication based on Zero-Knowledge which provides the best verifying formula than Feige-Fiat-Shamir algorithm. The protocol fulfils the ZKP properties and protected against discrete logarithm attack and man-in-middle attack. The algorithms serves as key exchange algorithm with the addition to authentication services. With the help of Zero-Knowledge Protocols we know the strength of our protocols, keys and the hardware.

ACKNOWLEDGEMENT

I thank all who have supported for this research.

REFERENCES

[1] Goldwasser, S.; Micali, S.; Rackoff, C.: "The knowledge complexity of interactive proof systems". In: ACM symposium on theory of computing. ACM Press, New York, pp. 291–304 (1985)
[2] Feige, U.; Fiat, A.; Shamir, A.:" Zero-knowledge proofs of identity". J. Cryptol. 1(2), 77–94 (1988)
[3] U. Feige, A. Fiat, AND A. Shmair, "Zero Knowledge proofs of identity", in Proc 19th Annual ACM Symposium on theory of computing, 1987, pp. 210-217.
[4] Krantz, Steven G., (2007), "Zero Knowledge Proofs", AIM Preprint Series, Volume 10-46, July25, 2007.
[5] Back, Amanda, (2009), "The Diffe-Hellman Key Exchange", December 2, 2009, http://129.81.170.14/~erowland/courses/20092/projects/Back.pdf.
[6] Kizza, Joseph M, (2010), "Feige-Fiat-Shamir ZKP Scheme Revisited", International Journal of Computing and ICT Research, Vol. 4, No. 1, June 2010.
[7] Krantz, Steven G., (2007), "Zero Knowledge Proofs", AIM Preprint Series, Volume 10-46, July25, 2007. [1] W. Simpson, Request for Comments 1994, PPP Challenge Handshake Authentication Protocol (CHAP), Network Working Group, California, 1996.
[8] Datta, Nivedita; (2012), "Zero-knowledge Password Authentication Protocol", Supercomputer Education & Research Center, NIISc, Banglore.

[9] "Password Authentication Protocol" Wikipedia, the free Encyclopedia(http://en.wikipedia.org/wiki/Password_authentication_protocol)

[10] "Zero-knowledge proof." Wikipedia, The Free Encyclopedia (http://en.wikipedia.org/wiki/Zeroknowledge_ proof).

[11] WY Lai, CM Chen, B Jeng. Information exchange mechanism based on reputation in mobile P2P networks. Proceedings of the Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing. 2007; 2: 200-204.

[12] Challenging epistemology: Interactive proofs and zero knowledge Justin Bledin Group in Logic and the Methodology of Science, University of California, 910 Evans Hall #3840, Berkeley, CA 94720-3840, USA, Journal of Applied Logic 6 (2008) 490–501

[13] Wang Cheng. The Study of P2P Security Model Based on Identity Authentication and Trust mechanism. Nan Jing Post and Telecommunications. 2007.

[14] Michael Backes and Dominique Unruha, (2009), "Computational Soundness of Symbolic Zero-Knowledge Proofs", Journal of Computer Security, Vol.18, No. 6, pp. 1077-1155, 2010.

[15] Hellman, Martin E., (2002), "An Overview of Public Key Cryptography", IEEE Communications Magazine, May 2002, pp: 42-49.

[16] Sandhya, M; Rangaswamy, R, T; (2013), "Hash-Based Secure Access Control Scheme For Mobile RFID System Using Zero-Knowledge" Arab J Sci Eng 39:1897-1906.

[17] Ibrahem, M.K. (2012), "Modification of Diffie–Hellman key exchange algorithm for zero knowledge proof", In: International Conference on Future Communication Networks, pp. 147–152.

[18] Han, S.; Potdar, V.; Chang, E.: (2007), "Mutual authentication protocol for RFID tags based on synchronized secret information with monitor", In: International Conference on Computational Science and its Applications, pp. 227–238.

