ZERO KNOWLEDGE PASSWORD AUTHENTICATION PROTOCOL

NIVEDITA DATTA

Supercomputer Education & Research Centre, NIISc, Bangalore, Bangalore, India
E-mail : nivedita.mtech@gmail.com

Abstract – In many applications, the password is sent as cleartext to the server to be authenticated thus providing the eavesdropper with opportunity to steal valuable data. This paper presents a simple protocol based on zero knowledge proof by which the user can prove to the authentication server that he has the password without having to send the password to the server as either cleartext or in encrypted format. Thus the user can authenticate himself without having to actually reveal the password to the server. Also, another version of this protocol has been proposed which makes use of public key cryptography thus adding one more level of security to the protocol and enabling mutual authentication between the client & server.

Keywords- computer network, computer security, authentication protocol, zero-knowledge proof , password.

I. INTRODUCTION

a) Motivation

In today's world of Internet, most of the people have mail accounts or accounts with social networking sites etc. where they need to authenticate themselves before logging in and being able to access their resources. However, very few people are actually aware of the fact that many of such applications make use of PAP(Password Authentication Protocol) in order to authenticate the users which is not very secure.

In case of PAP, though the password is stored in hashed format on the server along with its corresponding username making it less vulnerable to attacks, still the fact that the username-password pair travels in clear on the wire makes it vulnerable to attacks like eavesdropping & packet sniffing which will easily reveal the sensitive data to the intruder.

Here it is assumed that only the user knows the sensitive data(password) which is a secret information for him.

b) Contribution

This paper presents a protocol using which the users can be authenticated by the authentication server without having to reveal the password. This protocol, based on zero knowledge proof[6], thus protects the sensitive data from being revealed to the server or any intruder listening to the communication channel. It is meant to to be basically used in distributed system or peer to peer networks.

This paper first presents a simple version of the ZK-PAP in which the user can authenticate himself to the server without revealing the password[8]. The protocol uses a challenge-response mechanism (between the server and client) based on nonce. A nonce is a randomly generated number to be used only once throughout the session in order to avoid replay attacks.

The simple version of this protocol supports only one way authentication i.e. only the clients can authenticate themselves to the server. However, the other way round authentication is not possible.

The other version of this protocol i.e. ZK-PAP with PKE incorporates the concept of public key cryptography[4] thus adding a second level of security to the protocol and also enabling two-way authentication, i.e. the client can authenticate the server and vice versa.

c) Organization of paper

The paper has been briefly divided into four sections. The first section introduces the readers to the basic notations and concept such as zero-knowledge proof [6,7,10] and PAP [11] which one needs to understand before he can understand the protocol proposed. The second section gives a basic idea about the CHAP authentication protocol which is a relevant work in this area.

The third section gives some brief idea about the basic primitives or building block of the protocol followed by description of working of the protocol proposed in this paper.

II. NOTATIONS AND DEFINITIONS

a) Notations

In this section, we shall be discussing some of the basic notations which we will encounter in the paper later. Key $k \in K$ is symmetric session key which will be established between the user and client in every session to carry out the further communication. H is a collision resistant hash function used to generate the hash value of any data. As discussed already, nonce is a randomly generated data denoted by Ni (N1, N2 etc) and transformation function is any simple mathematical function which can be applied on integer data (assuming that nonce here is integer in nature).

Also we have encryption & decryption functions which are denoted by E & D respectively. In case of asymmetric (public key) cryptography, $E_{PR\text{-}A}$ & $E_{PU\text{-}A}$ represent encryption using private key & public key

of A respectively. Similarly, D_{PR-A} & D_{PU-A} represent decryption using private key & public key of A respectively. In case of symmetric (private key) cryptography, as we have no concept of public key hence E_{PR-A} & D_{PR-A} represent encryption and decryption respectively using the secret key of A.

b) Definitions

Here we shall be discussing the concepts of zero-knowledge proofs and PAP in brief.

Zero-knowledge Proof :

Let us first discuss the concept of zero knowledge proofs. The concept of zero-knowledge can be explained with the help of a classical example of two identical balls[9]. Suppose a person, say 'A' has two identical billiards ball of different colors, say red and blue. Now he want to convince his friend, say 'B' that the two balls are of different colors.

The basic approach will be to give the two balls to B so that he can see them and confirm the fact that the two balls are of different colors or not. However, in this scheme B gains knowledge about the colors of the balls.

Using the zero-knowledge approach, however A can convince his friend B that he has balls of different colors without having B see the balls actually. To do this, A blindfolds B and then places a ball on each of B's hand. Though B has no idea about which ball is of which color but A can see the color of the two balls.

Now A asks B to take his hands at the back and either swap the arrangement of the two balls or keep the arrangement same as original and show him the balls again. A sees the new arrangement of the balls and lets B know whether the balls were swapped or not. Thus A can prove to B that he has given him balls of different colors without revealing anything about color of the balls.

Let us say they play this game 't' times, where the value of t is large. If A tries to cheat B by giving him both the balls of same color, then the probability that A will still be able to answer correctly in each game is 2^{-t} which is negligible for large value of t.

This is a zero knowledge approach since A convinced B that he has two balls of different colors but at end of all games, B does not gain any knowledge about the colors of the two balls or any knowledge on how to distinguish the two balls.

Another classic example to understand zero-knowledge proof is given in [15] which uses the example of magic cave to explain the same concept.

Password Authentication Protocol:

Let us now discuss about the Password Authentication Protocol(PAP)[12,13]. PAP is an authentication protocol which is being used by point-to-point protocol to validate and authenticate users before they can access resources. This protocol requires the user to send the username and password to the authenticating server in cleartext thus making it vulnerable to packet sniffing & eavesdropping.

After the server receives the username & password, it generates hash of the password using the same algorithm which was used to hash the password before storing it into the password file. Then the generated hash is matched against the stored password hash corresponding to the entered user name. If a match is found, then the user is allowed to login else access is denied.

Here, though the password is stored in encrypted format on the server thus making it less vulnerable to attacks but sending the unencrypted ASCII password over the network makes the protocol insecure.

III. RELEVANCE TO PRIOR WORK

One of the relevant work done in this field is the CHAP(challenge handshake authentication protocol)[1,2,3,5,12]. This protocol is based on challenge-response model and makes use of single-use keys to provide more security. However this system does not completely eliminate the need to send data over wire in plain text format.

This protocol works in the following manner : when a user types his user name, the server generates a random key and sends it to the client machine(user) in unencrypted format. The user then encrypts his password using the received key and sends it to the server. The authenticator program on server encrypts the password corresponding to the received username using the generated key & matches it against the data received from the client machine.

The user is allowed to login and access his resources if the match occurs else access is denied.

Also, CHAP keeps sending various challenges to the client (user) throughout the session to verify that only an authorized person is logged in.

The main advantages of the scheme are as follows:

- It solves the problem of logged in but unattended systems.
- Also, the password no more travels in clear but in encrypted form thus solving the problem of packet sniffing or eavesdropping.

However, this scheme poses the following disadvantages :

- As the randomly generated key is sent to the user in clear, an intruder can get the key by packet-sniffing.
- The password on the server is stored in unencrypted format thus making it more vulnerable to attacks.
- Also, on continuously sniffing a line, the intruder will be able collect many key-ciphertext pairs for a user's password thus gaining some knowledge about the user's password.

IV. CRYPTOGRAPHIC PRIMITIVES

The algorithms which are designed to perform any cryptographic operation are known as cryptographic primitives. The primitives are the

building blocks which are used to create more complex cryptographic protocols to achieve various security goals. The primitives can be classified into two major groups : symmetric (or private key) & asymmetric (or public key). We will now define some of the primitives used in the proposed protocol:

a) *Collision Resistant Hash Function[16] :* A collision resistant hash function is a function which takes a variable length input and produces a fixed length output with the property that even slightest change in the input will reflect change in the output(hash value). The input to hash function is called a message and the output is known as hash value.

The collision resistant hash function exhibits the following four properties:

- It should be easy to compute hash value for any message.
- It should be infeasible to deduce the message from the hash value. This is known as one way property.
- It should be infeasible to find two different messages say m1 & m2 with same hash value. This property is known as collision resistance.

b) It should be infeasible to change a message without reflecting any change in it's hash value.

This kind of hash function has many applications such as digital signature, MAC etc.Block Cipher : It is one of the most important primitives of various cryptographic algorithms & protocols like MAC & various hash functions. It is used mainly to provide confidentiality of data. Block cipher works on fixed length inputs known as blocks. These ciphers encrypt or decrypt one block of data at a time. Some of the most widely used block ciphers are DES (Data Encryption Standard), 3DES (Triple DES) and AES (Advanced Encryption Standard).Stream Cipher : It is another most important & common primitives for various cryptographic algorithms. In this case, the encryption or decryption of data takes place one bit at a time. Thus it can be treated as a block cipher with block size of 1 bit. Some of the most commonly used stream cipher algorithms are ORYX, SEAL & RC4.Transformation Function : Transformation function is any simple mathematical function which can be applied to an integer. Here the transformation function is applied on the nonces to avoid replay attacks.

V. THE PROPOSED SECURITY PROTOCOL : ZERO KNOWLEDGE PASSWORD AUTHENTICATION PROTOCOL (ZK-PAP)

As in general scenario, every user has a username & password used to login to a system to access various resources. The password is secret to the user which only he can change when logged in to the application and the same change is registered with the server.

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

The protocol is initiated by the prover by sending his username and a challenge(nonce) N1 to the verifier in clear. 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 received user name. This encrypted data is then sent to the prover.

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. 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. The prover then applies the transformation function on the nonce N2, encrypts it with the received session key and sends it to the verifier.

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 is denied. As in CHAP, in ZK-PAP, a series of challenges can be exchanged between the prover & verifier through out the session to verify that only an authorized person is logged in.

The main advantages of this protocol are as follows:

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

Here it is assumed that the security of the server is not compromised else the protocol becomes vulnerable to attacks. In this protocol, we can also use time stamp instead of nonce, however that will incur an overhead of keeping all the communicating systems synchronized in time

.

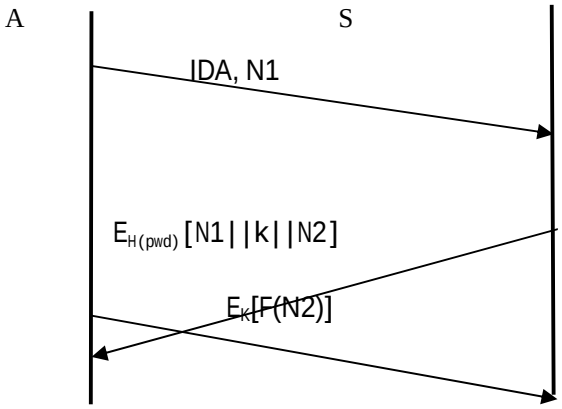

Fig 1 : Zero Knowledge Password Authentication Protocol

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

VI. ZK-PAP WITH PKE

This section briefs about the other version of the ZK-PAP protocol described above. This version of the protocol makes use of public key encryption[4] in order to give an added level of security and also enable two-way authentication ie. the verifier(server) can authenticate the prover(client) and vice versa.

Here it is assumed that all the users have (or can get) the public key of the server and the server has or can receive public keys of all the users. The protocol works as follows:

- The user,say A sends his username and a nonce to the server after encrypting it with server's public key.
- The server decrypts the message with his private key and extracts the value of the nonce N1.
- 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 user A, then with public key of the user A and sends the encrypted data to A.
- 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.
- 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.
- The server decrypts the received value with its private key & then with the shared session key.
- The user A is allowed to login if the server receives the expected value else access is denied.

As it can be seen from the above steps, only server will be able to extract the correct value of nonce N1 as it was encrypted with server's public key. Thus, if the client receives correct value of the nonce N1 from the server, it knows that the message was sent by the server itself and not by some intruder. Thus, use of public key encryption also allows the client to authenticate the server thus enabling mutual authentication.

Also, a series of challenge can be exchanged between the server and client to ensure that only an authorized person is logged in. This will also solve the problem of logged-in but unattended systems or workstations.

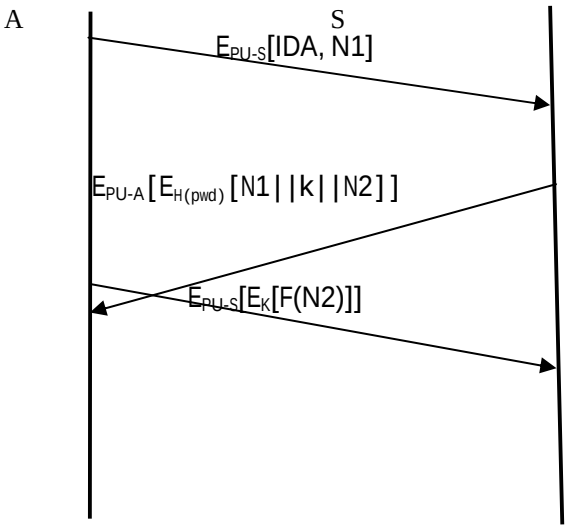

Fig 2 : ZK-PAP with PKE

Notations used:
IDA : User name 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

VII. CONCLUSION

This paper illustrates 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. As the protocol uses the hash of the password as key, using a strong encryption cipher (in which

key-recovery is hard) will strengthen the security of this protocol.

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. Both protocol proposed here are simple & efficient, thus enabling their practical use.

ACKNOWLEDGMENT

The author extends thanks to Indian Institute of Science at Bangalore, India for introducing me to this fascinating field in cryptography and giving me the opportunity to study it for my own interest.

REFERENCES

[1] W. Simpson, Request for Comments 1994, PPP Challenge Handshake Authentication Protocol (CHAP), Network Working Group, California, 1996.

[2] Securing Authentication of TCP/IP Layer Two by Modifying Challenge-Handshake Authentication Protocol, M. W. Youssef and Hazem El-Gendy, Advanced Computing: An International Journal (ACIJ), Vol.3, No.2, March 2012

[3] G. Zorn, Request for Comments: 2759: Microsoft PPP CHAP Extensions- Version 2, Network Working Group, Microsoft Corporation, 2000.

[4] Dolev, and A. Yao. On the Security of Public Key Protocols. IEEE Transactions on Information Theory, 29(2):198-208, March 1983.

[5] Verification of two versions of the Challenge Handshake Authentication Protocol (CHAP), Guy Leduc, Research Unit in Networking (RUN)

[6] Oded Goldreich. Zero-knowledge twenty years after its invention. Un-published manuscript. 2002.

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

[8] "Zero-knowledge password proof" Wikepedia, The Free Encyclopedia. (http://en.wikipedia.org/ wiki/Zero-knowledge_password_proof)

[9] 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

[10] A Survey of Zero-Knowledge Proofs with Applications to Cryptography, Austin Mohr, Southern Illinois University at Carbondale

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

[12] Microsoft TechNet, "Authentication Methods" (http://technet.microsoft.com/en-us/library/cc958013.aspx)

[13] Microsoft Technet "Password Authentication Protocol" (http://technet.microsoft.com/enus/library/cc737807%28v=ws.10%29)

[14] B. Lloyd & W. Simpson, Request for Comments 1334, PPP AUTHENTICATION PROTOCOLS Network Working Group , October 1992

[15] Jean-Jacques Quisquater, Louis C. Guillou, Thomas A. Berson. How to Explain Zero-Knowledge Protocols to Your Children (http://www.cs.wisc.edu/□mkowalcz/628.pdf). Advances in Cryptology- CRYPTO '89: Proceedings, v.435 p.628-631, 1990.

[16] "Cryptographic Hash Function" Wikipedia, the free encyclopedia (http://en.wikipedia.org/wiki/ Cryptographic_hash_function)

◈◈◈