A Secure Key Exchange for Symmetric Peer Server against Passive Attack

K.ARUN PRASAD[1], DIVYA.G[2], SWETHA MURALI[3]
[1]Research Scholar, St.Peters University, [2, 3] Agni College of Technology, Chennai, India
arunprasad.it@act.edu.in[1], divyaganesan09@gmail.com[2],muraliswetha93@gmail.com[3]

International Conference on Green Technologies for Power Generation and Communications-ICGPC 2014

Abstract—Password-authenticated key exchange (PAKE) is where a client and a server, who share a password, authenticate each other and meanwhile establish a cryptographic key by exchange of messages. The main aim of this project is to authenticate a client, in the scenario where two Peer servers co-operate for Authentication and if one server is compromised, the attacker still cannot pretend to be the client with the information from the compromised server. Because no Password Information will be stored and keeps providing services instead of any Crash Report. Our Two-Server PAKE protocol is Symmetric, and runs in parallel in authenticating a client by Encrypted key exchange (EKE), providing efficient services to user.

Keywords: *Password-authenticated key exchange, dictionary attack, Diffie-Hellman key exchange, ElGamal encryption, Active and Passive attack.*

1. INTRODUCTION

Earlier password-based authentication systems transmitted a cryptographic hash of the password over a public channel which makes the hash value accessible to an attacker. When this is done, and it is very common, the attacker can work offline, rapidly testing possible passwords against the true password's hash value. Studies have consistently shown that a large fraction of user-chosen passwords are readily guessed automatically. For example, according to Bruce Schneier, examining data from a 2006 phishing attack, 55 percent of MySpace passwords would be crackable in 8 hours using a commercially available Password Recovery Toolkit capable of testing200, 000 passwords per second in 2006 [2].Recent research advances in password-based authentication have allowed a client and a server mutually to authenticate with a password and meanwhile to establish a cryptographic key for secure communications after authentication. In general, current solutions for password based authentication follow two models. The first model, called PKI-based model, assumes that the client keeps the server's public key in addition to share a password with the server. In this setting, the client can send the password to the server by public key encryption. Gong [6], [7] were the first to present this kind of authentication protocols with heuristic resistant to offline dictionary attacks, and Halevi and Krawczyk [6] were the first to provide formal definitions and rigorous proofs of security for PKI-based model. The second model is called password-only model. Bellovin and Merritt [4] were the first to consider authentication based on password only, and introduced a set of so-called "encrypted key exchange" protocols, where the password is used as a secret key to encrypt random numbers for key exchange purpose. Formal models of security for the password-only authentication were first given independently by Bellare et al. [3] and Boyko et al. [8]. Katz et al. [9] were the first to give a password-only authentication protocol.

2. RELATED WORKS

The seminal work in the area of password-based key exchange is the encrypted key exchange (EKE) protocol of Bellovin and Merritt [7]. In their protocol, two users execute an encrypted version of the Diffie Hellman key exchange protocol, in which each row is encrypted using the password shared between these two users as the symmetric key. Due to the simplicity of their protocol, several other protocols were proposed in the literature based on it [6], each with its own instantiation of the encryption function. Our protocol is also a variation of their EKE protocol. Minimizing the use of random oracles. One of our main goals is to provide schemes that are simple and efficient, but relying as little as possible on random oracles.

Ideally, one would want to completely eliminate the need of random oracles as done in the KOY protocol [6]. However, such protocols tend to be less efficient than those based on the EKE protocol of Bellovin and Merritt [7]. To understand the difficulties involved in the design of protocols with few random oracles, let us consider the extreme case of the protocol in Figure 1 in which no random oracles are used. Despite being secure against passive attacks, this protocol can be easily broken by an active adversary performing a man-in-the-middle attack. Such an adversary can easily create two different sessions whose session keys are related in a predictable manner. For instance, an adversary can do so by multiplying X? by gr for a known value r. The relation between the underlying session keys SK A and SKB is SKB = SKA _ Y r. Hence, if the adversary learns the value of these two keys, it can perform an o_-line dictionary attack using Y = (SK B=SKA)r and Y ? to recover the password. Moreover, since the adversary can use arbitrary values for r, we cannot detect such attacks.

3. PRELIMINARIES

3.1 Diffie-Hellman Key Exchange Protocol

The Diffie-Hellman key exchange protocol [10] was invented by Diffie and Hellman in 1976. It was the first practical method for two users to establish a shared secret key over an unprotected communications channel. Although it is a non authenticated key exchange protocol, it provides the basis for a variety of authenticated protocols. Diffie-Hellmankey exchange protocol was followed shortly afterward by RSA [5], the first practical public key cryptosystem. Consider two users Alice and Bob, who know nothing about each other, but wish to establish secure communications between them, Diffie-Hellman key exchange protocol can be used as follows:

1. Alice and Bob agree on a cyclic group GG of large prime order q with a generator g.
2. Alice randomly chooses an integer a from ZZ_ q and computes X ¼ ga, while Bob randomly chooses an integer b from ZZ_ q and computes Y ¼ gb. Then Alice and Bob exchange X and Y .
3. Alice computes the secret key k1 ¼ Y a ¼ gba, while Bob computes the secret key k2 ¼ Xb ¼ gab. It is obvious that k1 ¼ k2 and thus Alice and Bob have agreed on the same secret key, by which the subsequent communications between them can be protected.

Diffie-Hellman key exchange protocol is secure against any passive adversary, who cannot interact with Alice and Bob, attempting to determine the secret key solely based upon observed data. The security is built on the well-known computational Diffie-Hellman (CDH) and decisional Diffie- Hellman (DDH) assumptions as follows: CDH assumption. Consider a cyclic group GG of large prime order q with a generator g. The CDH assumption states that, given (GG; g; ga; gb) for randomly chosen a; b from ZZ_ q ,it is computationally intractable to compute the value gab.DDH assumption. Consider a cyclic group GG of large prime order q with a generator g. The DDH assumption states that, given (GG; g; ga; gb) for randomly chosen a; b from ZZ_q , the value gab looks like a random element in GG. This intuitive notion is formally stated by saying that no probabilistic polynomial time (PPT) algorithm can distinguish the following two probability distributions with a probability more than 1/2 plus a non negligible value.. (ga; gb; gab), where a and b are randomly and independently chosen from ZZ_q .. (ga; gb; gc), where a; b; c are randomly and independently chosen from ZZ_q .

3.2 ElGamal Encryption Scheme

The ElGamal encryption scheme was invented by ElGamal in 1985 [1] on the basis of Diffie-Hellman key exchange protocol. It consists of key generation, encryption, and decryption algorithms as follows:
1. Key generation. On input a security parameter k, it publishes a cyclic group GG of large prime order q with a generator g. Then it chooses a decryption key x randomly from ZZ_q and computes an encryption key y ¼ gx.
2. Encryption. On inputs a message m 2 GG and the encryption key y, it chooses an integer r randomly from ZZ_q and outputs a ciphertext C ¼ Eðm; yÞ ¼ ðA;BÞ ¼ ðgr;m _ yrÞ.
3. Decryption. On inputs a ciphertext (A;B), and the decryption key x, it outputs the plaintext m ¼ DOC;xÞ ¼ B=Ax. ElGamal encryption scheme is a probabilistic encryption scheme. If encrypting the same message with

ElGamal encryption scheme several times, it will, in general, yield different ciphertext. Tsiounis and Yung [2] proved ElGamal encryption scheme to be semantically secure under the DDH assumption. ElGamal encryption scheme has useful homomorphic properties as follows:
Given an encryption of m, Eðm; yÞ ¼ ðA; BÞ, one can compute ðA; _BÞ ¼ Eð_m; yÞ for any _ in GG, an encryption of _m, and one can also compute ðA_;B_Þ ¼ Eðm_; yÞ for any _ in ZZ_ q , an encryption of m_. Given encryptions of m1 and m2, Eðm1; yÞ ¼ ðA1;B1Þ and Eðm2; yÞ ¼ ðA2;B2Þ, one can compute ðA1A2;B1B2Þ ¼ Eðm1m2; yÞ, an encryption of m1m2. solutions for two-server PAKE, we assume the two servers never collude to reveal the password of the client. When the two servers cooperate to authenticate a client C, we assume that the client C can broadcast a message to both of S1 and S2 simultaneously, but stress that we do not assume a broadcast channel and, in particular, an attacker can deliver different messages to the two servers or refuse to deliver a message to a server. In our protocol, the client and the two servers communicate through a public channel which may be eavesdropped, delayed, replayed, and even tampered by an attacker. Our protocol is symmetric if two peer servers equally contribute to the authentication in terms of computation and communication.

Definition 1.
Proposed protocol is correct if each server establishes a secret session key with the client in the end.
An adversary in our system is either passive or active. We consider both online dictionary attack, where an attacker attempts to login repeatedly, trying each possible password, and offline dictionary attack, where an adversary derives information about the password from observed transcripts of log sessions. The online dictionary attack cannot be prevented by cryptographic means but can be easily detected and suspended once the authentication fails several times. We assume that an adversary can compromise one server only and obtain all information stored in the server. A passive adversary is able to monitor the communications among the client and two servers. An active adversary is able to pretend to be both one server and the client to communicate with the honest server or pretend to be both two servers to communicate with the legal client, deviate in an arbitrary way from the actions prescribed by the protocol. In our protocol, the adversary attempts to learn the secret session key established between the client and the honest server. In an active attack, an adversary can learn the secret session key between the client and the honest server if the adversary can determine the password of the client. In general, we say that our protocol is secure if no adversary can succeed in any passive and active attacks in case that one server is compromised.

4. PROPOSED PROTOCOL

4.1 Secret Key Establishment:

The J2EE Environment is setup and Two Peer Servers are initialized and release the public parameters for the user by exchange of keys using Diffe-Helman key Exchange. The Secret key is established between Two Servers for further secure communication for Peer Servers. The Secret Session key will ensure that the two servers are genuinely involved in the process of User Registration and Authentication to provide Peer Services for the Genuine User. Proposed protocol runs in three phases - initialization, registration and authentication. Of Which Initialization comes Under Secret Key Establishment which uses Diffie-Helman key Exchange.

4.2 System Initialization:

The two peer servers S1 and S2 jointly choose a cyclic group G of large prime order q with a generator g1 .Next, S1 randomly chooses an integer s1 from Z* q and S2 randomly chooses an integer s2 from Z * q , and S1 and S2 exchange g1^s1 and g1^s2 . After that, S1 and S2 jointly publish public system parameters G, q, g1, g2 , where g2 = g^s1s2 .

4.3 User Registration and Authentication:

The public parameters released by Peer Servers will be used by client Registration Process, while a user registering to the Peer Services for Encryption of Password Shares and providing Authentication Information to Server1 and server2 Respectively. Registration and Authenticated Key Exchange are the Next Two Phases of our Protocol.

4.4 Registration:

We refer to the concept of public key cryptosystem, the encryption key of one server should be unknown to another server and the client needs to remember a password only after registration. Prior to authentication, each client C is required to register both S1 and S2 through different secure channels. First of all, the client C generates decryption and encryption key pairs (xi, yi) where yi = g1^xi 1 for the server Si (i=1,2) using the public parameters published by the two servers. Next, the client C chooses a password pwC and encrypts the password using the encryption key y, according to El-Gamal encryption. At last, the client C delivers the password authentication information Auth(1) C = {x1, a1, b1, E(g2^pwC , y2)} to S1 through a secure channel, and the password authentication information Auth(2) C = {x2, a2, b2, E(g2^pwC , y1)} to S2 through another secure channel. After that, the client C remembers the password pwC only.

4.5 Authentication and Key Exchange:

The two servers S1 and S2 have received the password authentication information of a client C during the registration. The following steps are involved in the process of Authentication.

1.The client C randomly chooses an integer r from Z * q , computes R = g1 ^ r * g 2 ^ −pwC and then broadcasts a request message M1 = {C,Req,R} to the two servers S1 and S2.The Diffie-Helman Key Exchange Occurs between Peer Servers Which run in parallel and establishes a secret session to fetch the password authentication Information and the two servers mutually generate two values and send Hash functions to Client based on their Password Authentication Information.

2. The client computes the Hash functions sent and Ex-or ing the Hashes will produce the Hash of his own Password. If the Password Hash and computed Hash are same the cli8ent can ensure that he is connected with Genuine Servers and can continue enjoying Services from Peer Servers without worry.

3.So the user needs to Remember the Password Only. Not anything else. He is safe and secure under our Proposed Model.

5. CONCLUSION

Multiple Services are embedded into a single gateway that prevents the flow of our personal data over a public channel during third party transactions. The clients can enjoy the service even a server is disclosed by any kind of attacks even if it is shut down manually for new Deployment purpose. Our Session Hand Over Mechanism will hand over all the user session at the time of attack to another server. So user Session will retain safe and the requests will be redirected to active server from them.

REFERENCES

[1] M. Abdalla and D. Pointcheval, “Simple Password-Based Encrypted Key Exchange Protocols,” Proc. Int’l Conf. Topics in Cryptology (CT-RSA), pp. 191-208, 2005.

[2] M. Abdalla, O. Chevassut, and D. Pointcheval, “One-Time Verifier-Based Encrypted Key Exchange,” Proc. Eighth Int’l Conf. Theory and Practice in Public Key Cryptography (PKC ’05), pp. 47-64,2005.

[3] M. Bellare, D. Pointcheval, and P. Rogaway, “Authenticated Key Exchange Secure against Dictionary Attacks,” Proc. 19th Int’l Conf.Theory and Application of Cryptographic Techniques (Eurocrypt ’00), pp. 139-155, 2000.

[4] S. Bellovin and M. Merritt, “Encrypted Key Exchange: Password-Based Protocol Secure against Dictionary Attack,” Proc. IEEE Symp. Research in Security and Privacy, pp. 72-84, 1992.

[5] D. Boneh and M. Franklin, “Identity Based Encryption from the Weil Pairing,” Proc. 21st Ann. Int’l Cryptology Conf. (Crypto ’01),pp. 213-229, 2001.

[6] Y. Yang, R.H. Deng, and F. Bao, “A Practical Password-Based Two-Server Authentication and key Exchange System,” IEEE Trans. Dependable and Secure Computing, vol. 3, no. 2, pp. 105-114,Apr.-June 2006.

[7] Y. Yang, R.H. Deng, and F. Bao, “Fortifying Password Authentication in Integrated Healthcare Delivery Systems,” Proc. ACM Symp.Information, Computer and Comm. Security (ASIACCS ’06), pp. 255-265, 2006.

[8] X. Yi, R. Tso, and E. Okamoto, “ID-Based Group Password- Authenticated Key Exchange,” Proc. Fourth Int’l Workshop Security:Advances in Information and Computer Security (IWSEC ’09), pp. 192-211, 2009.

[9] J. Katz and M. Yung, “Scalable Protocols for Authenticated GroupKey Exchange,” Proc. Advances in Cryptology Conf. (Crypto ’03), pp. 110-125, 2003