Dual server authentication with key exchange protocol

[1]Ms.Annaram Mounika ; [2]Mr.K.Raghavendra Rao ; [3]Mr.G.Vishnu Murthy
[1]M.Tech Student, Department of Computer Science and Engineering, Anurag Group of Institutions, Telangana, India.
[2]Assistant Professor, Department of Computer Science and Engineering, Anurag Group of Institutions, Telangana, India.
[3]Professor and Head of the Department, Department of Computer Science and Engineering, Anurag Group of Institutions, Telangana, India.
[1]Mail id: annarammounika07@gmail.com, [2]Mail id: raghava514.k@gmail.com,
[3]Mail id: hodcse@cvsr.ac.in

***Abstract**:*

In a two-server password-authenticated key exchange (PAKE) protocol, a client splits its password and stores two shares of its password in the two servers, respectively, and the two servers then cooperate to authenticate the client without knowing the password of the client. In case one server is compromised by an adversary, the password of the client is required to remain secure. There are two compilers that transform any two-party PAKE protocol to a two-server PAKE protocol on the basis of the identity-based cryptography, called ID2S PAKE protocol. By the compilers, one can construct ID2S PAKE protocols which achieve implicit authentication. As long as the underlying two-party PAKE protocol and identity-based encryption or signature scheme have provable security without random oracles, the ID2S PAKE protocols constructed by the compilers can be proven to be secure without random oracles. Compared with the Katz et al.'s two-server PAKE protocol with provable security without random oracles, our ID2S PAKE protocol can save from 22 to 66 percent of computation in each server. .

Keywords

PAKE Protocol, Cryptography, Security, Encryption, Password, Authentication.

1. Introduction

To secure communications between two parties, an authenticated encryption key is required to agree on in advance. So far, two models have existed for authenticated key exchange. One model assumes that two parties already share some cryptographically-strong information: either a secret key which can be used for encryption/authentication of messages, or a public key which can be used for encryption/ signing of messages. These keys are random and hard to remember.

In practice, a user often keeps his keys in a personal device protected by a password/PIN. Another model assumes that users, without help of personal devices, are only capable of storing “human-memorable” passwords. Bellovin and Merritt were the first to introduce password-based authenticated key exchange (PAKE), where two parties, based only on their knowledge of a password, establish a cryptographic key by exchange of messages. A PAKE protocol has to be immune to on-line and off-line dictionary attacks.

In an off-line dictionary attack, an adversary exhaustively tries all possible passwords in a dictionary in order to determine the password of the client on the basis of the exchanged messages. In on-line dictionary attack, an adversary simply attempts to login repeatedly, trying each possible password. By cryptographic means only, none of PAKE protocols can prevent on-line dictionary attacks. But on-line attacks can be stopped simply by setting a threshold to the number of login failures. PAKE, numerous PAKE protocols have been proposed. In general, there exist two kinds of PAKE settings, one assumes that the password of the client is stored in a single server and another assumes that the password of the client is distributed in multiple servers. PAKE protocols in the single-server setting can be classified into three categories as follows.

Password-only PAKE: Typical examples are the “encrypted key exchange” (EKE) protocols given by Bellovin and Merritt, where two parties, who share a password, exchange messages encrypted by the password, and establish a common secret key.

PKI-based PAKE: PKI-based PAKE protocol was first given by Gong et al., where the client stores the server's public key in addition to share a password with the server. Halevi and Krawczyk were the first to provide formal definitions and rigorous proofs of security for PKI-based PAKE.

ID-based PAKE: ID-based PAKE protocols were proposed by Yi et al., where the client needs to remember a password in addition to the identity of the server, whereas the server keeps the password in addition to a private key related to its identity. ID-based PAKE can be thought as a trade-off between password-only and PKI-based PAKE. In the single-server setting, all the passwords necessary to authenticate clients are stored in a single server. If the server is compromised, due to, for example, hacking or even insider attacks, passwords stored in the server are all disclosed. This is also true to Kerberos, where a user authenticates against the authentication server with his username and password and obtains a token to authenticate against the service server. To address this problem, the multi-server setting for PAKE was first suggested in, where the password of the client is distributed in n servers. PAKE protocols in the multi-server setting can be classified into two categories as follows.

Threshold PAKE: The first PKI-based threshold PAKE protocol was given by Ford and Kaliski, where n severs, sharing the password of the client, cooperate to authenticate the client and establish independent session keys with the client. As long as n^{-1} or fewer servers are compromised, their protocol remains secure. Jablon gave a protocol with similar functionality in the password-only setting. MacKenzie et al. proposed a PKI-based threshold PAKE protocol which requires only t out of n servers to cooperate in order to authenticate the client. Their protocol remains secure as long as t^{-1} or fewer servers are compromised. Di Raimondo and Gennaro suggested a password-only threshold PAKE protocol which requires less than 1/3 of the servers to be compromised.

Two-server PAKE: Two-server PKI-based PAKE was first given by Brainerd, where two servers cooperate to authenticate the client and the password remains secure if one server is compromised. A two-server password-only PAKE protocol was given by Katz et al., in which two servers symmetrically contribute to the authentication of the client. The protocol in the server side can run in parallel. Efficient protocols were later proposed, where the front-end server authenticates the client with the help of the back-end server and only the front-end server establishes a session key with the client. These protocols are asymmetric in the server side and have to run in sequence. Yi et al. gave a symmetric solution which is even more efficient than asymmetric protocols .Recently, Yi et al. constructed an ID2S PAKE protocol with the identity-based encryption scheme (IBE) .To address this problem, the multi-server setting for PAKE was first suggested, where the password of the client is distributed in n servers.

The objective of this paper is to propose a new compiler for ID2S PAKE protocol based on any identity-based signature scheme (IBS).

2. Literature Survey

ID2S PAKE Based on IBS

We need an identity-based signature scheme (IBS) as our cryptographic building block. A high-level description of our compiler is given in which the client and two servers A and B establish two authenticated keys, respectively. If we remove authentication elements from our compiler, our key exchange protocol is essentially the Diffie-Hellman key exchange protocol [1].

New Directions in Cryptography

The development of computer controlled communication networks promises effortless and inexpensive contact between people or computers on opposite sides of the world, replacing most mail and many excursions with telecommunications. For many applications these contacts must be made secure against both eavesdropping and the injection of illegitimate messages. At present, however, the solution of security problems lags well behind other areas of communications technology. Contemporary cryptography is unable to meet the requirements, in that its use would impose such severe inconveniences on the system users, as to eliminate many of the benefits of teleprocessing.

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel order to use cryptography to insure privacy, however, it currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such a private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Public key distribution systems offer a different approach to eliminating the need for a secure key distribution channel. In such a system, two users who wish to exchange a key communicate back and forth until they arrive a key in common. A third party eavesdropping on this exchange must find it computationally infeasible to compute the key from the information overheard.

A second problem, amenable to cryptographic solution which stands in the way of replacing contemporary business communications by teleprocessing systems is authentication. In current business, the validity of contracts guaranteed by signatures. A signed contract serves as gal evidence of an agreement which the holder can present in court if necessary. The use of signatures, however, requires the transmission and storage of written contracts. In order to have a purely digital replacement for his paper instrument, each user must be able to produce message paper instrument, each user must be able to produce message not have been produced by anyone else, even the recipient. Since only one person can originate messages but many people can receive messages, this can be viewed as broadcast cipher Current electronic authentication techniques cannot meet this need.

Authenticated Key Exchange Secure Against Dictionary Attacks

Password-based protocols for authenticated key exchange (AKE) are designed to work despite the use of passwords drawn from a space so small that an adversary might well enumerate, off line, all possible passwords. While several such protocols have been suggested, the underlying theory has been lagging. We begin by defining a model for this problem, one rich enough to deal with password guessing, forward secrecy, server compromise, and loss of session keys.

Id-based two-server password authenticated key exchange

Secret key authenticated key exchange (PAKE) protocols are intended to be secure in addition to when the secrete key utilized for validation is a human-paramount password. In this paper, we consider PAKE protocols in the partite situation, in which a partite of clients, each of them imparts a password to a "legitimate yet inquisitive" server, mean to set up a typical secrete key (i.e., a partite key) with the assistance of the server. In this setting, the key set up is known to the clients just and nobody else, including the server. Every client needs to recollect passwords just while the server keeps passwords in addition to private keys identified with his identity.

Efficient two-server password-only authenticated key exchange

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. In this setting, all the passwords necessary to authenticate clients are stored in a single server. If the server is compromised, due to, for example, hacking or even insider attack, passwords stored in the server is all disclosed. In this paper, we consider a scenario where two servers cooperate to authenticate a client and if one server is compromised, the attacker still cannot pretend to be the client with the information from the compromised server.

Current solutions for two-server PAKE are either symmetric in the sense that two peer servers equally contribute to the authentication or asymmetric in the sense that one server authenticates the client with the help of another server. This paper presents a symmetric solution for two-server PAKE, where the client can establish different cryptographic keys with the two servers, respectively. Our protocol runs in parallel and is more efficient than existing symmetric two-server PAKE protocol, and even more efficient than existing asymmetric two-server PAKE protocols in terms of parallel computation

An efficient password-only two server authenticated key exchange system

One of the prominent advantages of secret key just two server authenticated key exchange is that the client password will stay secure against disconnected lexicon attacks even after one of the servers has been bargained. The principal arrangement of this sort was proposed by Yang, Deng and Bao in 2006. The framework is proficient with a sum of eight communication adjusts in one protocol run. Nonetheless, the security suppositions are solid. It accepts that one specific server can't be exchanged off by a dynamic foe. It likewise accepts that there exists a protected communication channel between the two servers.

3. System Analysis

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

A. Existing System

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.

Disadvantage:

1. The hash value accessible to an attacker.

2. The attacker can work offline, rapidly testing possible passwords against the true password's hash value.

3. An adversary can always succeed by trying all passwords one-by-one in an on-line impersonation attack. A protocol is secure if this is the best an adversary can do. The on-line attacks correspond to send queries.

B. Proposed System

Typical examples are the "encrypted key exchange" (EKE) protocols given by Bellovin and Merritt, where two parties, who share a password, exchange messages encrypted by the password, and establish a common secret key. The formal model of security for PAKE was firstly based on the security model; PAKE protocols have been proposed and proved to be secure.

A security model for ID2S PAKE protocol was given and a compiler that transforms any two-party PAKE protocol to an ID2S PAKE protocol was proposed on the basis of the Cramer-Shoup public key encryption scheme and any identity-based encryption scheme, such as the Waters' scheme.

The second model is called password-only model. Bellovin and Merritt 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. and Boyko et al.. Katz et al. were the first to give a password-only authentication protocol which is both practical and provably secure under standard cryptographic assumption.

Advantages:

1. Establish a cryptographic key for secure communications after authentication.

2. The sense that an adversary attacking the system cannot determine session keys with advantage non-negligibly greater than that of an online dictionary attack.

4. Design and Implementation

A. Modules

1. User Registration Module

2. Cipher key Generation Module

3. Private key Generation Module

4. Key Requisition Module

Module Description:

1. User Registration Module:

In this Module, users get registered first where they need to fill the details like Name, Email, Phone No, Address etc.,. After they get registered, they need to login into the application using their username, password.

2. *Cipher key Generation Module:*

In this module, a 16-bit random key called cipher key is generated by the servers in which first 8-bit is generated by server1 and another 8-bit is generated by server2 in order to give access for downloading a file.

3. Private key Generation Module:

In this module, a 4-bit random key called private key is generated by the servers in which first 2-bit is generated by server1 and another 2-bit is generated by server2 in order to give access for downloading a file.

5. Conclusion

By storing the keys in two servers, it will be very difficult for the attacker to attack the data present in it. They may not be aware of multiple servers and splitting of the data.

In this way by storing data in multiple servers helps to provide more security to data.

6. References

1. W. Diffie and M. Hellman. New directions in cryptography. IEEE Transactions on information Theory, 32(2): 644-654, 1976.

2. M. Abdalla and D. Pointcheval. Simple password-based encrypted key exchange protocols. In Proc. CT RSA 2005, pages 191-208, 2005.

3. M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated key exchange secure against dictionary attacks.In Proc. Eurocrypt'00, pages 139-155, 2000.

4. E. Bresson, O. Chevassut, and D. Pointcheval. Security proofs for an efficient password-based key exchange. In Proc. CCS'03, pages241-250, 2003.

5. J. Brainard, A. Juels, B. Kaliski, and M. Szydlo. Nightingale: A new two server approach for authentication with short secrets. In Proc. 12th USENIX Security Symp., pages 201-213, 2003.

6. B. Clifford Neuman and Theodore Ts'o. Kerberos: An authentication service for computer networks. IEEE Communications, 32 (9):33-38, 1994.

7. O. Goldreich and Y. Lindell. Session-key generation using human passwords only. In Proc. Crypto'01, pages 408-432, 2001.

8. L. Gong, T. M. A. Lomas, R. M. Needham, and J. H. Saltzer.Protecting poorly-chosen secret from guessing attacks. IEEE J. on Selected Areas in Communications, 11(5):648-656, 1993.

9. H. Jin, D. S. Wong, and Y. Xu. An efficient password-only two-server authenticated key exchange system. In Proc. ICICS'07, pages 44-56, 2007.

10. X. Yi, R. Tso and E. Okamoto. Identity-based password authenticated key exchange for client/server model. In SE-CRYPT'12, pages 45-54, 2012.

11. X. Yi, F. Hao and E. Bertino. ID-based two-server password-authenticated-key exchange. In ESORICS'14, pages 257-276, 2014.