ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Implementation of Two-Server Password-Based Authentication

Jincy Sebastian, Sindhu Jose

PG Scholar, Dept. of CSE, VJCET, M.G University, Muvattupuzha, Kerala, India

Assistant professor, Dept. of CSE, VJCET, M.G University, Muvattupuzha, Kerala, India

ABSTRACT: There are many cross-domain communication scenarios, where the information being communicated may need to be protected against both passive and active attackers. In these scenarios, a user is typically registered to some kind of domain servers; two communicating parties from different domains very often neither share a password nor possess a public key certificate. 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. Authenticated Key Exchange protocols enable several parties to establish a shared cryptographically strong key over an insecure network using various authentication means. The password-based mechanism allows users to be authenticated by remote computer systems via easily memorable passwords. Designing a secure password-based system is a precise task. In this work the users are not required to have public key certificates; they share their login passwords with their respective domain servers. The domain servers, assumed to be part of a standard certificate authority that certify some key materials that the users can subsequently use to exchange and agree on as a session key. The authentication system implemented in a disconnected model provides a token for further communication after the client authentication process.

KEYWORDS: Password-based protocol, key exchange, cross-domain, Performance, technique, framework.

I. INTRODUCTION

Nowadays, passwords controls access to protected computer resources. A computer user may require passwords for logging in to computer accounts, retrieving e-mail, accessing programs, databases, web sites and so on. Earlier password-based authentication systems transmitted a cryptographic value of the password which makes the value accessible to attackers. When this is done, the attacker can work offline, rapidly testing possible passwords against the true password's cryptographic value. Studies have shown that a large fraction of user-chosen passwords are readily guessed.

In password-based authentication an end-user and an authentication server mutually authenticate with a password. This authentication mechanism establishes a cryptographic key for secure communications. This mechanism is useful for authentication in network systems. It allows users to be authenticated by remote systems via passwords. Since people like to select easily memorable strings as their passwords, many authentication systems are vulnerable. Current solutions [4] for password-based authentication follow two models. The first model, called PKI-based model, here the client keeps the server's public key in addition to share a password with the server. In this, the client can send the password to the server by public key encryption. The second model is called password-only model, where the password is used as a secret key to encrypt random numbers for the purpose of key exchange.

Key exchange protocols provide communication sessions over a public channel, with a secure session key, which allows establishment of virtual secure channels over insecure networks. Password Authenticated Key Exchange (PAKE) protocols have been played an essential role in providing secure communications. PAKE protocols permits an end-user and a server to authenticate each other and generate a strong session key using a password which is already shared between them. Designing a secure PAKE is non-trivial thing due to the fact that the password is picked up from a small space so that the protocol is vulnerable to attacks.

There are many cross-domain communication scenarios, such as email communication, mobile phone communication, and instant messaging, where the information being communicated may need to be protected from passive and active attackers. In these scenarios, a user is registered to domain server, such as email exchange server or home location register (in the cases of email and mobile phone communications, respectively). Moreover, the communicating parties from different domain very often neither share a password nor possess a public key certificate. Here the users are required to share their login passwords with their respective domain authentication servers. The authentication servers, assumed to be part of a standard certificate authority that certify some key materials that the users can subsequently use to exchange and agree on as a session key. The authentication system implemented in a disconnected model provides a token for further communication after the client authentication process.

II. EXISTING SYSTEM

The most important problem of cryptography is to provide secure communication between clients in a public communication channel. This problem is reduced to the problem of generating a secure session-key. There are many ways to establish secure session keys with the existence of Public Key Infrastructure. If two parties are allowed to obtain such a strong cryptographic session key without relying on the PKI, but with only a pre-shared memorable password is more convenient. The solution for this problem is known as Password Authenticated Key Exchange. Authentication relying on passwords is a popular method for user authentication in the client-server model because of its easy-to-memorize property. There are several password-based authenticated key exchange methods.

In single-server C2C-PAKE protocol [1], a single universal trusted third party is involved. All clients share their passwords with trusted third party rather than share a password between every pair of clients. So in the single-server C2C-PAKE protocol, every client only needs to remember his own password which enables him to establish a session key with another client through the trusted third party. Compared with normal PAKE protocol, single-server C2C-PAKE protocol provides an efficient solution for peer communication in a group of large number of users. But due to the single universal trusted third party assumption, this application is limited, which introduces secure-realm C2C-PAKE protocol. When all clients belong to the same organization or the same university, the single universal trusted third party may be available. However, when clients come from different organizations, it is not desire to put the whole system's security on a single trusted third party. Every organization would have their own trusted servers to manage clients' passwords and provide service for them. If a client shares his password with a server, he is in the realm of the server and every server stores the passwords of all clients belonging to his realm.

PAMKE1 [2] consists of three building blocks; two-party PAKE in which each user of a group and the server exchange a secret key, detection of on-line dictionary attacks in which each user and the server check whether there are malicious attempts or not to make use of they as an oracle for on-line dictionary attacks, and key distribution in which the server distributes randomly selected a secret key to each user using the secret key resulted in the two party PAKE. It has fault-tolerance. If some clients of a group are disconnected by network failures, the other clients who execute the protocol correctly can successfully share a session key without any additional message sending and delay. PAMKE1 is secure against off-line dictionary attacks since an attacker's capability to perform the off-line attacks is limited by its computational power, while the attacker can only test one password per a message. PAMKE2 [2] is designed to resist the curious servers. To achieve this goal, another approach called MAC key distribution and MAC-authenticated multi-party key exchange are used. Through this, the session key is determined not by the server but by all honest group users together. This protocol is still requires a constant number of rounds. It is not fully fault-tolerant. If someone among clients of a group receiving the broadcast messages from server is disconnected by network failures, the session key computation would be failed. Because the session key is correctly shared between clients, if and only if the clients involved in the MAC key distribution and MAC-authenticated multi-party key exchange phase are linked in a cyclic structure. Until the cyclic structures completed, the multi-party key exchange may be delayed.

III.PROPOSED SYSTEM

In order to establish a secure communication channel over an insecure public channel, ensure that secret session keys are exchanged securely. The session key shared used to achieve confidentiality or data integrity. The communicating client has to keep long random secret keys in the public-key based and symmetric-key based key exchange protocols. The client uses an additional storage device to keep the random string, since it is difficult for a human to memorize a long random string. The password-authenticated key exchange (PAKE) protocols allow sharing a secret session key between communicating clients using only a human-memorable password. Thus, PAKE protocols avoid the use of additional storage device. Hence, PAKE provides mobility and convenience. PAKE Protocols especially used in networks where a security infrastructure like Public-Key Infrastructure is not deployed. PAKE has received significant attention because PAKE protocols provide a unique and new way to authenticate parties and derive high-quality cryptographic keys from low-grade passwords.

There are many special scenarios in cross domain communication such as communication in private domains. In scenarios like file accessing applications clients save their files in domain servers in a special folder. The client can access the file if the domain server is up regardless of the client owning the file is up or not. The file owner is actually not involved in the communication. So this client is not involving in the process of session establishment. In this scenario, an authenticated key exchange protocol is used to establish a session key such that two users can securely communicate information from one domain to another. Each domain has at least a trusted domain server which acting as an authentication server serving a group of users. Each user within the domain does not own a public key certificate. The user shares a password with the server. The domain servers are connected to a certificate server. In this setting, the focus is on enabling a user from one domain to securely access a file of another user from a different domain through their respective domain servers. This system uses password-based techniques for authentication purpose and public-key cryptographic techniques for key exchange purpose.

The protocol run has two phases; in the first phase, the authentication servers corresponding to two communicating users provide some key materials that are associated with the users. In the second phase, the users can exchange the key materials and agree on a session key. The user trusts the authentication server to certify key materials submitted by an authenticated user. Through a 2PAKE protocol between the user and his authentication server the key material is obtained. The user needs to remember only a password to authenticate with the server and obtains a key material. Once both the sides achieved the certified key materials from their corresponding servers, they exchange the key materials following a two-party AKE protocol.

This framework (Two-Server Password-Based Authentication) consists of the following phases:

a) Local authentication: A runs a 2PAKE protocol with domain (authentication) server to establish a temporary session key for communication in the next step.
b) Token acquirement: Subsequently, A obtains an authentication token issued by domain server via the temporary session key established in the previous step. Certificate authority acquires a token for domain server B.
c) Verification: Domain server B verifies the token to approve the session key.
d) Session key generation: Finally the authentication tokens are used to establish a valid session.

In a protocol run, there is no interaction between servers. This can avoid overloading the servers with high communication cost in an open and distributed environment should they need to exchange messages in the protocol. This saves the communication bandwidth also. Generic framework for Two-Server Password-Based Authentication is shown in Figure 1.

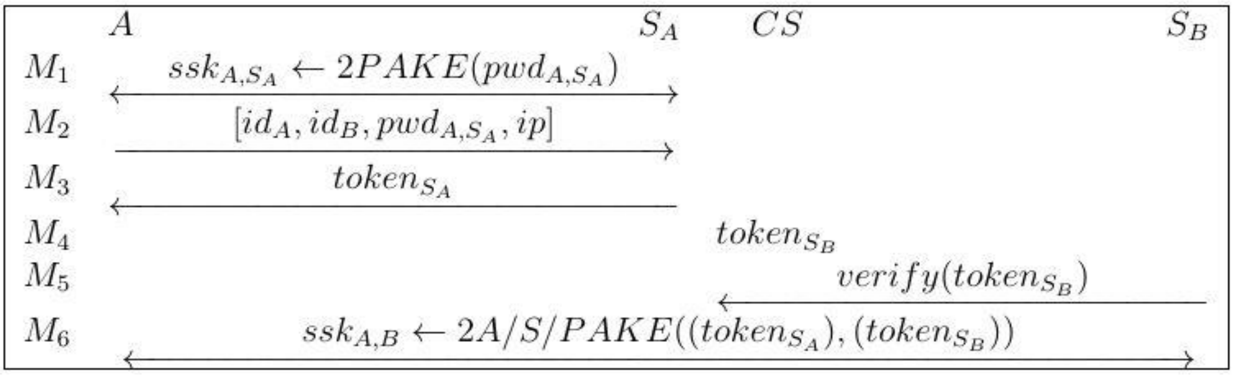
A
S_A
CS
S_B
M_1
$ssk_{A,S_A} \leftarrow 2PAKE(pwd_{A,S_A})$
M_2
$[id_A, id_B, pwd_{A,S_A}, ip]$
M_3
$token_{S_A}$
M_4
$token_{S_B}$
M_5
$verify(token_{S_B})$
M_6
$ssk_{A,B} \leftarrow 2A/S/PAKE((token_{S_A}),(token_{S_B}))$

Fig 1: Generic framework for Two-Server Password-Based Authentication

In the disconnected model of the proposed system, when the communication starts between two clients, first the source contacts with the certificate server and gets a public key certificate. Through that trusted certificate, the communication is performed between the two parties. The proposed model allows reusing the certificate for the entire communication between the two parties for one or more specified sessions. After getting the certificate, both the source and the destination applications are disconnected from the certificate server. The same certificate can be reused for the entire communication or for the specified time period. The main advantage of using such a model is that every time both the parties need not get different certificate for each communication. This leads to make the communication speedier. This increases the overall efficiency of the communication system. The system is similar to issuing a gate pass for a day, and for the entire day the party can use the same pass for authentication. This will reduce the time for authentication and communication bandwidth consumption. This will conserves system resources and provides maximum security for resources and also has less impact on system performance. This framework has following features:

a) Communication overhead: There is no interaction between servers S_A and S_B during a protocol run. This seems to be a very attractive property since overloading the servers with high communication cost in an open, distributed environment should they needs to exchange messages in the protocol can be avoided. The savings in terms of communication bandwidth is significant. This scheme achieves protocol interoperability by reusing existing two-party protocols and this protocol is easier to analyse.

b) Computational overhead: Computational complexity of this protocol is evaluated by counting the computationally expensive operations in a protocol run. The client-side computational overhead in this protocol is expected to be slightly higher, while the server-side overhead is expected to be slightly lower.

c) Key privacy: Each client possesses a copy of its server's credential in order to verify the authentication token issued by the server. This is not necessarily required to be done in advance, the servers can distribute their credentials to their clients during the execution of 2PAKE, or alternatively, by using a MAC algorithm. In that case, secure and authenticated key exchange can be achieved if neither the client nor the server is corrupted. However, to prevent a stronger password compromise impersonation attack, the clients must obtain their respective servers' credentials through out-of-band mechanisms.

IV.PERFORMANCE ANALYSIS

The performances of connected and disconnected model are evaluated based on the execution time of the protocol and execution delay for each request. The analysis conducted over three user sessions and total of twenty file accesses. The first, second and third user accessed eight, three and nine files respectively.

The execution delay analysis is shown in Table 1 and Figure 2. In Table 1 and Figure 2, the execution delay analysis of connected model shown that whenever a user session started the execution delay of the protocol is high, this happens due to the communication connection establishment time and it is 3.029, 1.88 and 1.764 seconds for first, nine-nth and twelfth file in Figure 2. Afterwards the delay is almost constant.

Table 1: Execution delay analysis

	Connected Model	Disconnected Model
1	3.029	1.718
2	0.306	0.117
3	0.257	0.16
4	0.278	0.123
5	0.226	0.12
6	0.391	0.177
7	0.692	0.112
8	0.392	0.167
9	1.88	1.84

DOI: 10.15680/IJIRCCE.2015. 0311074

10	0.552	0.395
11	0.335	0.175
12	1.764	1.83
13	0.161	0.234
14	0.153	0.216
15	0.847	0.169
16	0.975	0.193
17	0.4	0.28
18	0.254	0.165
19	0.157	0.145
20	0.463	0.124

In Table 1 and Figure 2, the execution delay analysis of disconnected model shown that the execution delay is high whenever a new user session started or a new session key is established, and almost constant for further communication. This hike in delay happens due to the communication connection establishment time or checking whether a valid session exists and new session key generation. The hike in delay due to new user session is shown at first, nine-nth and twelfth file in Figure 2 and due to new session key generation is shown at sixth and seventeenth file in Figure 2.

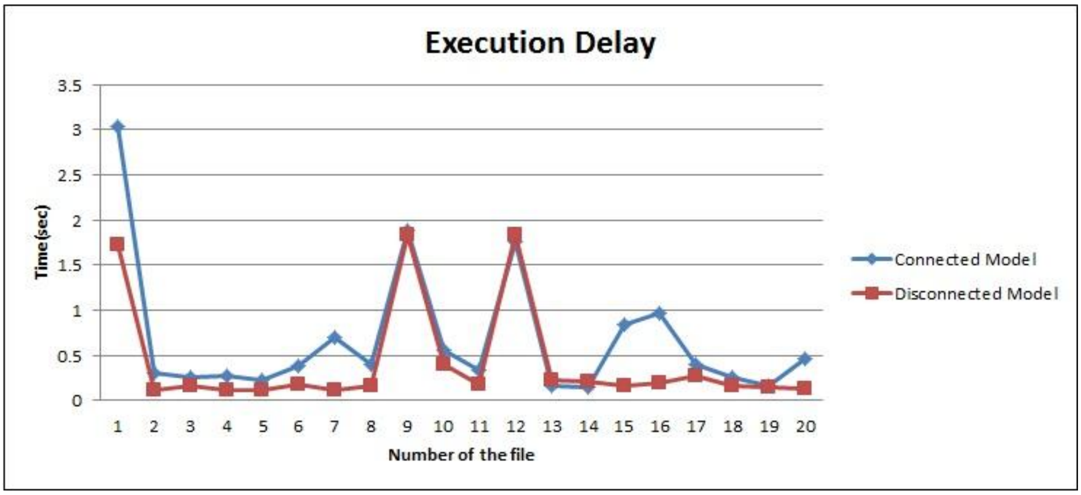

Fig 2: Execution delay analysis

Both the models are compared using the total execution time for each request, and it is clear that the execution time of disconnected model (8.46 sec) is always less than that of connected model (13.512 sec). The execution time analysis is shown in Table 2 and Figure 3.

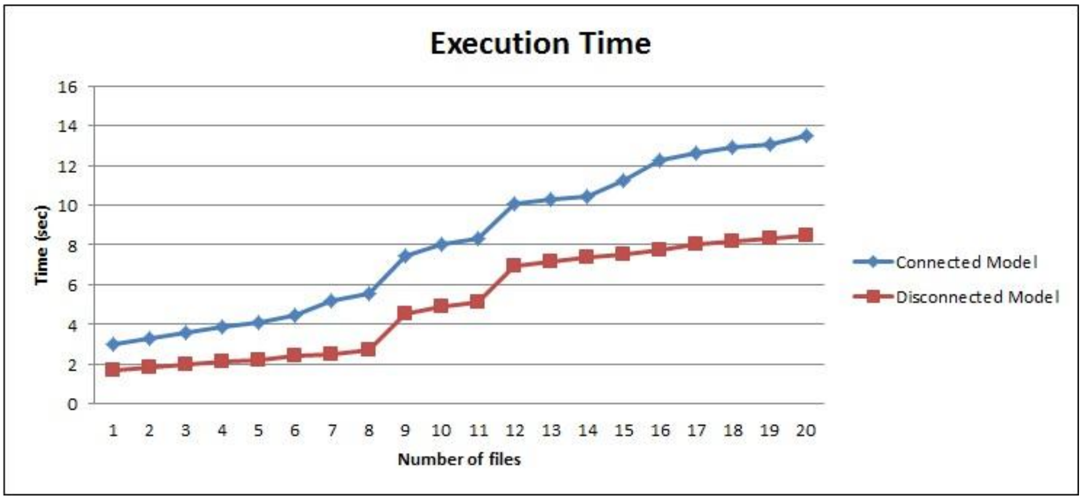

Fig 3: Execution time analysis

In a protocol run, there is no interaction between authentication servers. This can avoid overloading the servers with high communication cost in an open and distributed environment should they need to exchange messages in the protocol. This saves the communication bandwidth also. In disconnected model, the servers are disconnected from the certificate server, further saves the communication bandwidth.

Table 2: Execution time analysis

	Connected Model	Disconnected Model
1	3.029	1.718
2	3.335	1.835
3	3.592	1.995
4	3.87	2.118
5	4.096	2.238
6	4.487	2.415
7	5.179	2.527
8	5.571	2.694
9	7.451	4.534
10	8.003	4.929
11	8.338	5.104
12	10.102	6.934
13	10.263	7.168
14	10.416	7.384
15	11.263	7.553

DOI: 10.15680/IJIRCCE.2015. 0311074

16	12.238	7.746
17	12.638	8.026
18	12.892	8.191
19	13.049	8.336
20	13.512	8.46

V. CONCLUSION

The most important problem of cryptography is to provide secure communication between clients in a public communication channel. This problem is reduced to the problem of generating a secure session-key. Authentication relying on passwords is a popular method for user authentication in the client-server model because of its easy-to-memorize property. There are several password-based authenticated key exchange methods. In the connected model, the token acquired during authentication process establishes the secure session keys for a single file manipulation. In the disconnected model, the token acquired during authentication process establishes the secure session keys for a particular period of time. This will reduce the time for authentication and communication bandwidth consumption. This will conserves system resources and provides maximum security for resources and also has less impact on system performance. The performance analysis also shows that the time required to execute the protocol in disconnected model is less compared to that of connected model and the execution delay of the protocol in disconnected model is less.

ACKNOWLEDGMENTS

I would like to thank my co-author for her valuable comments on this quite active and interesting area.

REFERENCES

[1] Yin Yin, and Li Bao, “Secure Cross-Realm C2C-PAKE Protocol”, IEEE Conference on Information Security and Privacy, 2008.

[2] Jeong Ok Kwon, Ik Rae Jeong, Kouichi Sakurai, and Dong Hoon Lee, “Password-Authenticated Multi-Party Key Exchange with Different Passwords”, IEEE conference on Information security and cryptology, 2009.

[3] Liqun Chen, Hoon Wei Lim, and Guomin Yang, “Cross-Domain Password-Based Authenticated Key Exchange Revisited”, ACM Transactions on Information and System Security, Vol. 16, No. 4, April 2014.

[4] Xun Yi, San Ling, and Huaxiong Wang, “Efficient Two-Server Password-Only Authenticated Key Exchange”, IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 9, September 2013.