Towards Securing Client-Server Connections against Man-in-the-Middle Attacks

Mihai ORDEAN and Mircea GIURGIU
Communications Department
Technical University of Cluj-Napoca
28 Memorandumului Street, Cluj-Napoca, Romania
Email: {mihai.ordean, mircea.giurgiu}@com.utcluj.ro

***Abstract*—This paper presents the design concept for an authentication string that makes use of the server's public key and provides client's authenticity through its password without the need of a client side certificate or a second channel. Successful strategies for preventing man-in-the middle attacks are currently relying either on two channel/two factor authentication or two-way encryption. Both these strategies have their downsides, the first one requires users to carry a physical device for authentication and the second requires all the devices that connect to the server have encryption certificates.**

***Index Terms*—authentication: man-in-the-middle attack, connection security**

I. INTRODUCTION

The authentication process represents the gateway into any secure system. It is the process that links the real life person to its on-line avatar. Manny attacks target the process of authentication because it is one of the easiest methods of gaining access to a secure system under false pretenses. The most widespread mode of authenticating is the password, but unfortunately it is also the least secure. Because passwords usually have low entropy [1] they are usually relatively easy to discover using exhaustive search attacks.

To increase the security of the authentication process a secure connection is usually established between the connecting parties (i.e. client and server). In addition to providing encryption the process of creating the secure connection also has the ability to confirm the identity of the parties through the use of the Public Key Infrastructure (PKI). PKI however require mutual trusting of the certificate authority by all the parties involved in the communication and are subject to man-in-the-middle (MITM) attacks [1]. Additional methods to prevent MITM attacks use secondary transmission channels (e.g. mobile SMS messages), two-factor authentication (e.g. one time pads) or two-way encryption, but methods like the two-way encryption are difficult to implement and multi-factor authentication often requires possession of physical devices, and even then the credentials can be spoofed [1].

The PAKE protocol [2], [3], [4], [5], [6] represents an alternative to the PKI, but the current standard for the Internet remains the PKI.

This article proposes a method that uses the PKI infrastructure to implement some of the functionality of the PAKE authentication. The method describes the creation of an authentication string that becomes unusable in case of tampering or hijacking by a would-be attacker. By using this authentication string in a client-server authentication process the client is assured that an attacker cannot retrieve its password. The server is also assured that it is granting access to the legitimate client and not to a proxy client represented by the attacker.

The following sections detail the proposed concept. Section II analyzes the current problem of the MITM attack and its methods of deployment. Section III details the creation of the authentication string, while different attacks and their efficiency are being discussed in section IV.

II. THREAT MODEL

A man-in-the-middle attack consists of an attacker interposing between two communicating parties in order to eavesdrop or control the communication. This attack is usually directed at a client-server communications with the purpose of either obtaining authentication credentials or performing tasks in the client's name. Figure 1 describes an attack targeting user credentials. It is assumed that the attacker already knows the server's public key and the password alphabet. The *password alphabet* is referred as such to show applicability to alternate methods of authentication (e.g. visual authentication). The other assumption that is made is that the attacker does not know, nor can obtain the server's private key. The users password is also considered unknown for the attacker because it represents the objective of the attack.

Usually the attacker will proceed using one of the following variants:

1) Attempt to establish two secure connections: one with the server as a client and another one with the client posing as the server. Data can be modified in real-time for this case.
2) Attempt to pose as the server to the client in order to obtain targeted data which can be used at a later time. Data can be processed in a more extensive manner (i.e. exhaustive search attack).

Figure 1 presents the attacker with two simultaneous secure connections established, each with its own pair of keys. In one connection, with the client, the attacker assumes the role of the server and is able to trick the client into believing its legitimacy. In the other connection, with the server, the attacker poses as a normal client. The attacker is able to pose

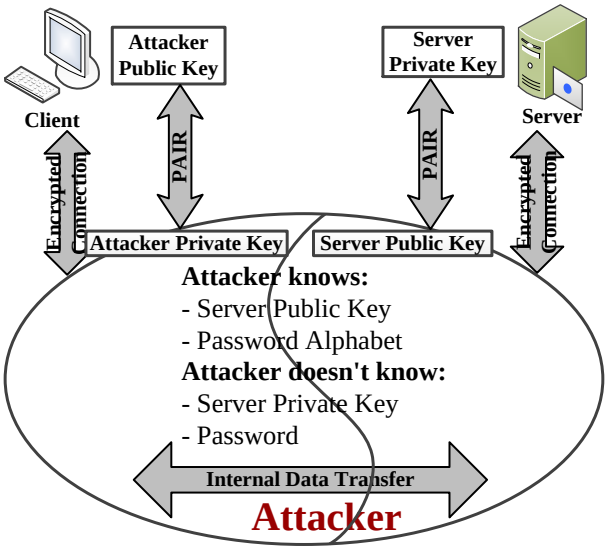

Figure 1: Man-in-the-middle attack diagram.

as the client to the server because the system does not use client identification methods (certificates, tokens).

The *Internal Data Transfer* process describes the link between the two connections. This process can be performed either in real-time mode where the attacker actively relays client's data to the server, or in off-line mode where the attacker records data (i.e. user's credentials) from the client and uses it at a later time.

Because each connection is secure, and the attacker successfully tricked the user into accepting its own certificate, both the client and the server think they are communicating with the intended party. The attacker, however, is able to read all communications. The following section proposes a method for creating an authentication string that becomes unusable even if intercepted by a listening attacker such as the one presented in figure 1.

III. AUTHENTICATION MODEL

The authentication process involves creating an authentication string using: *the server's public key, a random and unique connection ID, and the user's password.*

The full authentication process (figure 2) starts with the client requesting a connection to the server. The server sends a unique connection ID and exchanges keys with the client. Using the obtained public key and the connection ID the password is constructed in the following manner:

$$EncPass = CPbK(Hash(PWD \| CID \| CPbK)) \quad (1)$$

where:

$\|$ represents concatenation,
CPbK(x) is the encryption of the value x with the active connections public key,
Hash(x) is the output of a hash function applied on the value x,
EncPass is the encrypted password,
CPbK is the Base64 encoding of the active connections public key,
PWD is the user's password,
and **CID** is the received connection ID encoded in Base64.

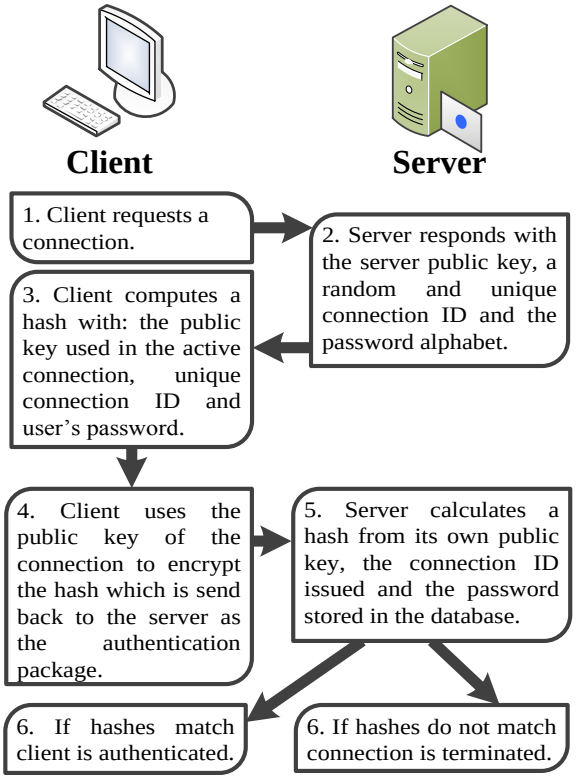

Figure 2: Client-server interaction during the authentication phase.

The encrypted password (i.e. EncPass) created from the constructed hash encrypted with the active connection public key is sent back to the server. The server is able to compute the encrypted hash from its own data, independent of the user, having the self-generated connection ID, its own public key and the password from the user database. Furthermore the server is also able to decrypt the received encrypted password (i.e. EncPass) using its private key and check for a match between the two computed hashes. In the event of a missmatch access the connection is terminated, otherwise the user is successfully authenticated.

IV. ATTACK EVALUATION

Once a successful MITM attack is in place the actual attack can take two forms: either passively eavesdrop on

the communication between the client and the server and save useful data for further use, or actively take over the connection and relay communications between the two. During this relaying real-time modifications of the data transmitted can be performed. The two attacks forms are described in figures 3 and 4.

A. MITM Proxy Attack

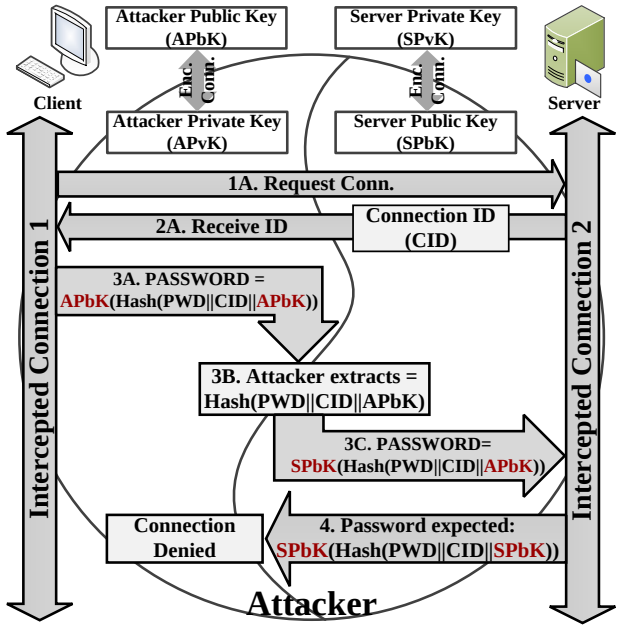

Figure 3: Real-time MITM proxy attack.

Figure 3 describes a MITM attack that actively modifies and relays data from the client to the server. In this scenario the attacker successfully represents itself as a server to the client by sending its own public key enabling the decryption of the data transmitted by the client. In relation with the server the attacker poses as a normal client and hopes to transmit data received from the client, after it has been processed to suit its needs.

Typical data exchange between the client and the server in the proposed attack scenario is described as follows:

1) The client requests the connection to the server, not knowing the server is actually the attacker (i.e. step 1A in figure 3). The attacker relays the request in its name to the real server (i.e. step 1B in figure 3).
2) The server responds with a unique connection ID which is passed on, unmodified, to the client, by the attacker (steps 2A, 2B). The purpose of the attacker is to obtain the authentication string, so relaying the connection ID unmodified is the desired approach for the attacker.
3) The client builds the authentication string from the connection ID received, the user's password and the public key for the active connection (i.e. the connection with the attacker). In order to prevent a possible exploit in which the attacker is able to trick the user into embedding the legitimate server's public key instead of the connection public key, the hash is also encrypted with the connection's public key. This authentication message is then sent back to the attacker posing as the server.
4) The attacker at this point can decrypt the received message and obtain the plain hash, but re-encrypting this hash with the legitimate server's public key would not grant him access (steps 3A, 3B, 3C and 4), because of the public key that is built into the hash.

The attacker is faced with the problem of recreating a hash knowing the connection ID and the public key, and without knowing the user's password, which was the purpose of the attack.

B. Replay Attack

The replay attack can be seen as a special case of a real-time MITM Attack, one that takes place with a significant delay between the data received and the data send by the attacker. This attack consists of a gathering phase and a replay phase. In the data gathering phase the attacker tries to either eavesdrop some of the data exchanged between the client and the server, or recreate part of the server's interface and database in order to trick the user into providing its credentials (i.e. phishing attack). In the replay phase the attacker uses gathered data in order to authenticate. This attack is very successful even if the gathered data is encrypted or masked as the legitimate server has no way of knowing if the connecting entity is the attacker or the legitimate client.

The two phases work as follows:

1) The attacker is able to passively capture the data exchanged on a secure connection between the server and a client (C1, C2 and C3 in figure 4). Once the authentication string is captured the attacker can mount the replay attack.
2) When the attacker tries to authenticate at a later time with the obtained authentication string, will receive a different connection ID (R1, R2, R3 in figure 4).

The challenges for the attacker is to decrypt the string, and recompute the hash for the new connection ID, without knowing the server's private key and the user's password. The only feasible approach is to retry the connection until the received connection ID matches the one in the hash, but this approach can be easily avoided with a large enough connection ID string.

V. CONCLUSIONS

Man-in-the-middle attack still represents a major security concern even if successful ways to prevent it have been developed. Most of the problems arise from the fact that the security measures required for protection against it are hard to deploy or may seem inconvenient, especially when usability comes into question. While the new standards do offer high level of protection they are not yet fully supported by all applications.

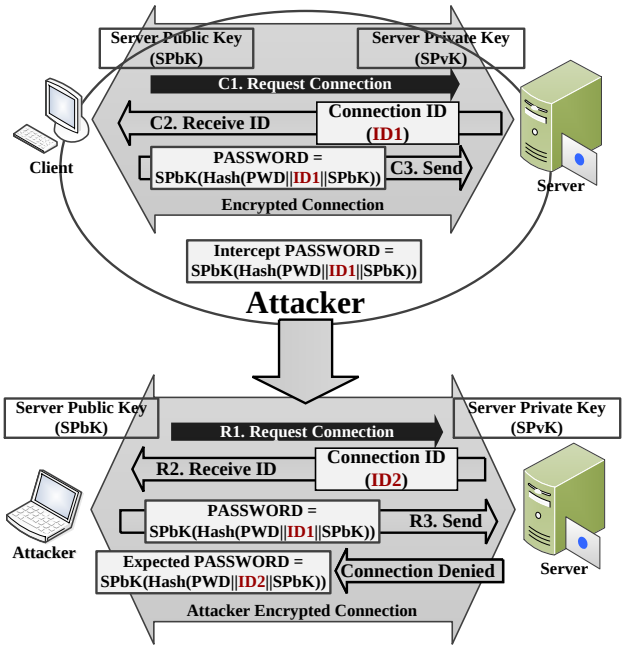

Figure 4: Replay attack.

This paper shows that a method exists that is able to confirm the identity of a client involved in a client-server secure connection using the current Internet security standard (certificate encryption), but without the need for the client to have its own certificate.

Several attacks were illustrated to prove the security of the proposed concept.

ACKNOWLEDGMENTS

This paper was supported by the project "Doctoral studies in engineering sciences for developing the knowledge based society-SIDOC contract no. POSDRU/88/1.5/S/60078, project co-funded from European Social Fund through Sectoral Operational Program Human Resources 2007-2013.

REFERENCES

[1] R. Anderson, *Security Engineering : A Guide to Building Dependable Distributed Systems*. Wiley, 2001.

[2] F. Hao, "J-PAKE: authenticated key exchange without PKI," *Transactions on computational science XI*, 2010.

[3] K. Paterson and D. Stebila, "One-time-password-authenticated key exchange," *Information Security and Privacy*, pp. 1–15, 2010.

[4] J. Katz, "Efficient Cryptographic Protocols Preventing Man-in-the-Middle Attacks," 2002.

[5] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Exchange Organizational Behavior Teaching Journal*, pp. 139–155, Springer, 2000.

[6] D. Jablon, "Strong password-only authenticated key exchange," *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 5, pp. 5–26, 1996.