International Journal of Smart Home
Vol. 9, No. 9 (2015), pp. 177-184
http://dx.doi.org/10.14257/ijsh.2015.9.9.19

Efficient Anonym Smart Card Based Authentication Scheme for Multi-Server Architecture

An Braeken

Industrial Engineering INDI, Vrije Universiteit Brussel VUB, Nijverheidskaai 170, 1070 Brussel
an.braeken@vub.ac.be

Abstract

Multi-server authentication schemes are very practical from a user point of view, since they allow a user to get access to different services on different servers with one single registration. Smart card based approaches lead to more secure systems because they offer two-factor authentication, based on the strict combination of user's password and the possession of the smart card. In this paper, we first show that a previously proposed scheme does not satisfy perfect forward secrecy and is not resistant against insider attacks. Next, we propose a very efficient smart card based authentication scheme, solely using xor and hash operations, which is resistant against dishonest users and servers. Also anonymity and untraceability of user's behaviour is avoided.

Keywords: *Anonymity, Authentication, Multi-server architecture, Smart card*

1. Introduction

An enormous amount of servers, providing an even larger number of services, are currently available. Most of these services can only be accessed after successful authentication of the user in order to verify its identity. A classical authentication method is to send to each of these services an identity and hashed password, which is then compared with the stored verification table at server side. Such method encompasses several security problems and requires huge storage capacities. In 2000, Hwang et al. [1] introduced the smart card based authentication. Here, a registration center (RC) activates the card with some secret information, which is (partly) shared with the registered servers. During the last decade, a lot of research has been performed on developing a practical and secure authentication protocol for this type of setting.

The goal of this paper is to show an important weakness in a recently proposed protocol [2]. Besides the fact that the protocol offers resistance against the most common security attacks, it turns out that it is completely broken once there is a malicious user or a dishonest server. Moreover, any server can follow the communications with other servers and the system does not satisfy perfect forward secrecy as it claims to do. Our improved version is based on the ideas of [3], with the inclusion of untraceability of the user.

The rest of the paper is structured as follows. Section 2 presents related work. In Section 3, we discuss the security weaknesses from [2]. Section 4 describes our protocol. In Section 5, we analyze the security and performance of the protocol. Finally, we conclude the paper in Section 6.

2. Related Work

In the survey [4], an overview is given of the most important security attacks and the proposed schemes in literature for smart card based authentication on single-server and multi-server architectures. We here focus on the last type of schemes,

ISSN: 1975-4094 IJSH

since they are more general and offer additional user friendliness since the user does not need to collect different identities and passwords for each service.

We can basically distinguish two types of approaches in the authentication schemes, public key based and symmetric key based schemes. Most recent examples of the first category are [5-7]. However, as public key based operations are computationally much more demanding and do not offer additional benefits, we will focus on the symmetric key based approach. Recently, Banerjee et al. [2] showed weaknesses in a previous authentication scheme and proposed the first scheme, solely based on xoring and hash operations, which offers protection against stolen smart card attack, replay attack, user impersonation attack, insider attack, and satisfies perfect forward secrecy. However, as we will show in the next section, they consider the adversary of the insider attacks only as an insider of the RC. A more general version of this attack is to consider the adversary also as a malicious user or dishonest server. We also show that the scheme does not satisfy perfect forward secrecy.

On the other hand, also three-factor authentication, by including biometrics into the authentication schemes, has been proposed in literature [3,8-14]. In [3,14] it is shown that the systems from [8-13] do not offer resistance against the well-known security attacks. In [14], a new system is proposed, but requires a communication between server and RC for each login of the user, which is not very practical. The proposed system from [3] does not require such additional communication and is shown to be resistant against insider attacks, also for insiders as malicious users or dishonest servers. However, the system is said to satisfy anonymity, but still allows traceability of a user.

To summarize, our scheme will use ideas from [3], but restrict to a two-factor authentication since we believe that the solution with biometrics is much more costly and can in many cases still be circumvented. However, we also include the property of user untraceability into the system.

3. Authentication Scheme from [2]

The authentication scheme from [2] consists of five phases, namely registration, user login, verification, password change, and smart card revocation phase. We will restrict the explanation to the first three phases, as we will use them to explain the weaknesses of the scheme. There are three types of participants into the scheme, being the user U_i with the smart card, the RC, and the server S_j. The RC chooses a master secret key x and a secret number y, and computes $h(x||y)$ and $h(y)$. These numbers are shared with the servers S_j.

The same notations as in [2] are used and are summarized in Table I. After the explanation of the different phases, we show the security weaknesses of the scheme.

Table I. Notations and Definitions in this Paper

Notation	Definitions
U_i	i-th user
S_j	j-th server
RC	Registration center
ID_i	Unique identification of U_i
PW_i	Password of U_i
N	Number of registration request by same ID_i
SID_j	Unique identification of S_j
x,y	Master secret key and secret number of RC
N_i, N_j	Nonce determined by U_i and S_j resp.
$h(.)$	Hash operation

$\oplus$	Bitwise xor operation
$\|\|$	Concatenation

3.1. Different Phases

3.1.1. Registration Phase: When a user U_i wants to access the services of one of the remote servers, controlled by the RC, the user needs to register himself with the RC. The following actions are then performed.

- U_i : The user U_i chooses his identity ID_i, password PW_i and computes $RPW_i=H(b \oplus PW_i)$, where b is a random number generated by U_i. Then, U_i sends ID_i and RPW_i to the RC for registration through a secure channel.
- RC: After receiving this request, the RC verifies the validity of ID_i. Denote by N be the number of registrations performed by the user. Then, the RC computes the following steps

$$A_i = h(x \,||\, ID_i \,||N)$$
$$B_i = h(ID_i \,||h(y)||RPW_i) \oplus A_i$$
$$V_i = h(A_i \,||h(y)||RPW_i)$$
$$D_i = h(A_i \oplus h(x||y))$$
$$E_i = A_i \oplus h(x||y)$$

- RC: Finally, RC stores $\{B_i, V_i, D_i, E_i, h(y), h(.)\}$ to the memory of U_i's smart card and sends it to the user through a secure channel.
- U_i : Upon receiving the smart card, U_i securely stores b. Consequently, the smart card contains $\{B_i, V_i, D_i, E_i, h(y), h(.), b\}$.

3.1.2. User Login Phase: In this phase, the user wants to obtain access to the services. He first inserts his smart card into the reader and inputs ID_i and PW_i. Then, the smart card does the following actions.

$$RPW_i=H(b \oplus PW_i)$$
$$A_i = h(ID_i||h(y)||RPW_i) \oplus B_i$$
$$V_i^* = h(A_i \,||h(y)||RPW_i)$$

If V_i^* equals to the stored V_i, then the user is verified and the process can be continued.

$$P_{ij} = E_i \oplus h(SID_j||h(y)||N_i)$$
$$CID_i = RPW_i \oplus h(D_i||SID_j||N_i)$$
$$C_1 = h(A_i||D_i||CID_i||N_i)$$
$$C_2 = h(SID_j||h(y)) \oplus N_i$$

Here N_i is a random nonce generated by the smart card. The login request $\{CID_i, P_{ij}, C_1, C_2\}$ is send to the server S_j.

3.1.3. Authentication Phase: In this phase, server S_j verifies the authentication of the request and a shared secret key between S_j and U_i is negotiated. S_j first executes the following steps.

$$N_i = h(SID_j||h(y)) \oplus C_2$$
$$E_i = P_{ij} \oplus h(SID_j||h(y)||N_i)$$
$$A_i = E_i \oplus h(x||y)$$
$$D_i = h(A_i \oplus h(x||y))$$
$$RPW_i = CID_i \oplus h(D_i||SID_j||N_i)$$
$$C_1^* = h(A_i||D_i||CIDi||N_i)$$

If $C_1^* = C_1$, then the user is authenticated and the process can continue. A new nonce N_j is determined and the following steps are executed.

$$C_3 = h(SID_j||D_i||RPW_i||N_j)$$
$$C_4 = RPW_i \oplus N_i \oplus N_j$$

The message $\{C_3, C_4\}$ is send to U_i. A shared secret key is derived from the nonces Ni, N_j.

3.2. Security Weaknesses

The main weakness follows from the fact that all servers share the same security material, $h(y)$, $h(x||y)$, with the RC. Moreover, also a malicious card user that is able to retrieve the storage material on the card can derive these values. The value $h(y)$ is directly stored on the card and $h(x||y)$ can be derived through $E_i = A_i \oplus h(x||y)$. As shown in the user login phase, the user is able to derive A_i and E_i is stored on the card.

Once $h(y)$, $h(x||y)$ are leaked or abused by a malicious entity (user or server), all existing smart cards should be revoked. Let us discuss several actions that become possible for an attacker.

3.2.1. Conflict with Perfect Forward Secrecy: The malicious entity can generate its own authentication material for any server S_j without passing through the RC. He simply needs to randomly choose the values C_2, P_{ij}, CD_i. The other values N_i, E_i, A_i, D_i, RPW_i and C_1 should just satisfy the pre-determined equations. Let us go a little bit more into detail.

- First, a random value for C_2 is chosen, and N_i is derived as $N_i = h(SID_j||h(y)) \oplus C_2$
- Next, a random value is chosen for P_{ij} and E_i is derived as $E_i = P_{ij} \oplus h(SID_j||h(y)||N_i)$
- Then, we find a value for A_i and D_i as $A_i = E_i \oplus h(x||y)$ and $D_i = h(A_i \oplus h(x||y))$
- Finally, we choose a value for CID_i and derive a value for RPW_i as follows $RPW_i = CID_i \oplus h(D_i||SID_j||N_i)$

- Next, we choose C_1 as $C_1 = h(A_i||D_i||CID_i||N_i)$
- The message $\{CID_i, P_{ij}, C_1, C_2\}$ is send to the server S_j.

The server will authenticate this message, since the computed C_1 will match with the transmitted value, as it is derived by the constructions explained above. Next, the message $\{C_3, C_4\}$ is computed, based on the material derived from the adversary. Consequently, the adversary can also derive the nonce N_j and determine the shared secret key.

3.2.2. Impersonation Attacks: The adversary with knowledge of $h(y)$, $h(x||y)$, is also able to impersonate the two other participants of the scheme, being the server and the user.

- He can impersonate a server by performing a man in the middle attack after receiving the message $\{CID_i, P_{ij}, C_1, C_2\}$.
- He can impersonate a user since he can derive RPW_i from a transmitted message $\{CID_i, P_{ij}, C_1, C_2\}$.
- He can derive the session keys from any user by intercepting the messages send from user to server and server to user.

Consequently, the system described above is completely broken for an insider attack with a server as inside attacker or a smart card stolen attack combined with an insider attack with a user as inside attacker. Moreover, it clearly does not satisfy perfect forward secrecy as claimed in [2].

4. Proposed Scheme

We will use similar notations and setting as the scheme described in the previous section. Again, the RC chooses a master secret key x and a secret number y. However, now the RC computes $h(x||y)$ and also $h(SID_j||h(x))$ for each server S_j. These values are send to each server individually through a secure channel.

We also distinguish the similar five phases as in the previous scheme and now describe each of them into more detail.

4.1. Registration Phase

In order to get access to the services, the user first registers with the RC as follows.

- U_i : The user U_i chooses his identity ID_i, password PW_i and computes $RPW_i=h(b \oplus PW_i)$, where b is a random number generated by U_i. Then, U_i sends ID_i and RPW_i to the RC for registration through a secure channel.
- RC: After receiving this request, the RC verifies the validity of ID_i. Denote by N the number of registrations performed by the user. Then, the RC computes the following steps:
  $A_i = h(ID_i \| N \| y)$
  $B_i = h(x\|y) \oplus A_i$
  $C_i = h(RPW_i \| ID_i) \oplus h(A_i)$
  $D_i = h(x) \oplus h(ID_i)$
  $E_i = RPW_i \oplus ID_i$
- RC: Finally, RC stores $\{B_i, C_i, D_i, E_i, h(.)\}$ to the memory of U_i's smart card and sends it to the user through a secure channel.
- U_i : Upon receiving the smart card, U_i securely stores b. Consequently, the smart card contains $\{B_i, C_i, D_i, E_i, h(.), b\}$.

4.2. User Login Phase

Now, the user wants to obtain access to the services. He first inserts his smart card into the reader and inputs ID_i and PW_i. Then, the smart card does the following actions.

$RPW_i = H(b \oplus PW_i)$
$ID_i^* = RPW_i \oplus E_i$

If ID_i^* equals to the transmitted ID_i, then the smart card authenticates the user and the process can be further continued. Again, let N_i be a random nonce generated by the smart card.

$h(x) = h(ID_i) \oplus D_i$
$h(A_i) = C_i \oplus h(RPW_i \| ID_i)$
$C_1 = h(SID_j \| h(x)) \oplus h(IDi \| N_i)$
$C_2 = h(A_i) \oplus N_i$
$V_1 = h(N_i \oplus B_i)$
$CID_i = B_i \oplus h(h(SID_j \| h(x)) \| h(IDi \| N_i))$

The login request $\{CID_i, V_1, C_1, C_2\}$ is send to the server S_j.

4.3. Authentication Phase

In this phase, a secret shared key is determined by mutual authentication between server and user. The server S_j starts with the following steps, using its stored secret material $h(x\|y)$ and $h(SID_j \| h(x))$.

$h(ID_i \| N_i) = h(SID_j \| h(x)) \oplus C_1$
$B_i = CID_i \oplus h(h(SID_j \| h(x)) \| h(IDi \| N_i))$
$A_i = B_i \oplus h(x\|y)$
$N_i = C_2 \oplus h(A_i)$
$V_1^* = h(N_i \oplus B_i)$

If V_1^* equals to the transmitted V_1, then the user is authenticated and the process can continue. A new nonce N_j is determined and the following steps are executed.

$SK_{ij} = h(h(ID_i \| N_i) \| SID_j \| B_i \| N_j)$
$C_3 = N_j \oplus h(ID_i \| N_i)$
$V_2 = N_i \oplus h(SK_{ij} \| N_j)$

The message $\{C_3, V_2\}$ is send to the user U_i. Now U_i performs the following steps

$N_j = C_3 \oplus h(ID_i \| N_i)$
$SK_{ij} = h(h(ID_i \| N_i) \| SID_j \| B_i \| N_j)$

$N_i^* = V_2 \oplus h(SK_{ij} || N_j)$

If N_i^* equals to the N_i of the login request, the value SK_{ij} is considered as a secret shared session key. A confirmation message can eventually be send from user to server.

4.4. Password Change Phase

Changing a password is possible without interaction with the RC. The user inserts the card in the smart reader, inputs its password and identity, and selects the option to change the password. If the card authenticates the user, it prompts the user for a new password PW_i^* and computes:

$RPW_i^* = H(b \oplus PW_i^*)$

$C_i^* = h(RPW_i^* || ID_i) \oplus h(A_i)$

$E_i^* = RPW_i^* \oplus ID_i$

The values of C_i and E_i are updated and stored on the card.

4.5. Revocation of User's Lost or Stolen Card

This phase is exactly the same as in [2] and follows from the fact that the RC stores the value N, which corresponds with the number of registration requests of a user with a given identity

5. Security and Performance Analysis of Proposed Scheme

5.1. Security Evaluation

The main reason why the attacks from the previous scheme cannot be applied is because each server contains different security material, which is in addition different from the security material of the user. Because of the same reasons as explained in [2,3], the scheme is resistant against stolen smart card attack, replay attack, and user impersonation attack. We now discuss the strongest type of attacks, being the insider attacks with a user or server as adversary. We also explain the forward secrecy and anonymity feature of the scheme.

5.1.1. Impersonation Attacks: We need to discuss two possible scenarios for the adversary, being a malicious inside user or a server. In the first case, the adversary cannot take over the role of server, since it does not know $h(x||y)$, and thus cannot derive A_i, required to determine N_i from the transmitted tuple. This value is essential in the derivation of the secret shared session key. However, it must be said that this adversary has the capability to trace the behaviour of a particular user, since it can derive B_i. On the other hand, it is not possible to link this user with an identity. So, the potential impact of this is negligible.

If the adversary represents a server, he cannot derive any useful information send to other servers since he is not aware of $h(SID_i||h(x))$ for other servers S_j.

5.1.2. Perfect Forward Secrecy: There are two long term secrets $h(x)$ and $h(x||y)$. The first one can be derived by the smart card of the user, the second one is stored on the server. If they are separately leaked and the other is not abused at the same time by an insider attack, the scheme obtains perfect forward secrecy because of the same reasons as explained above. As this combination of leakage and abuse is quite rare to appear, we may consider the scheme to satisfy perfect forward secrecy.

5.1.3. Anonymity: In [3], the scheme was said to satisfy anonymity. However, each message contained the static value B_i. Although, this message is not directly linked with a certain identity, an adversary may still track user's behaviour, which is not always

favourable for the users's privacy. In our scheme, the transmitted message $\{CID_i, V_1, C_1, C_2\}$ contains only dynamic material, since all values are dependent of the random none N_i, which changes in each request. As mentioned before, only malicious inside users and the intended server are able to derive the static information related to a particular user.

5.2. Performance Comparison

From the proposed systems in literature on smart card based authentication for multi-server architectures, including two-factor and three-factor authentication, almost all of them are broken for the classical attacks like impersonation attacks, man-in-the-middle attacks, password guessing attacks, etc. Exceptions are [3, 7, 14].

As mentioned before the system of [7] is using elliptic curve operations and therefore requires much more computational resources. Moreover, the scheme is also very inefficient with respect to storage at the smart card, since separate key material for each different server needs to be stored at the smart card. In our system, we use the trick to have different key material between user $h(x)$ and server $h(x||y)$, $h(SID_i||h(x))$ and where the user is able to build part of the key material (i.e. $h(SID_i||h(x))$) of the server.

Also the system of [14] is very inefficient with respect to our system, since the RC should be involved during the authentication process of the user. Moreover, if the RC becomes off-line for one or other reason, the protocol cannot be continued.

Since our proposed system is building on the same ideas of [3], the performance between both systems is more or less similar. Our system requires in the computation of CID_i, one additional hash and xor operation. However, it is important to note that the required operations are restricted to hash and xor operations, which are both very efficient operations in hardware.

6. Conclusions

In this paper, after showing a weakness in an existing system, we propose a new very efficient scheme for smart card based authentication on multi-server architecture. The scheme is resistant against the well-known security attacks, satisfies perfect forward secrecy and anonymity with untraceability. We believe that it is a very good candidate to be adopted in practice in order to assure user's authentication, obtained by the combination of the user's password and the possession of the smart card.

Acknowledgements

This work is partly supported by the COST Action IC1303.

References

[1] Hwang, M., Li, L., A new remote user authentications scheme using smart cards, IEEE Transactions on Consumer Electronics, 46(1), 28-30 (2000).
[2] Banerjee, S., Dutta, M.P., and Bhunia C.T., An mproved smart card based anonymous multi-server remote user authentication scheme, International Journal of Smart Home, 9(5), 11-22 (2015).
[3] Baruah, K.CH., Banerjee, S., Dutta, M.P., and Bhunia C.T., An Improved Biometric-based Multi server Authentication Scheme using Smart Card, International Journal of Security and Its Application, 9(1), 397-408 (2015).
[4] Tashi, J., J., Comparative analysis of smart card authentication schemes, IOSR Journal of Computer Engineering, 16(1), 91-97 (2014).
[5] Pippal, R.S., and Wu, S., Robust smart card authentication scheme for multi-server architecture, Wireless Personal Communications, 72(1), 729-745 (2013).
[6] Wei, J., Liu, W., and Hu, X. Cryptanalysis and improvement of a robust smart card authentication scheme for multi-server architecture, Wireless Personal Communications, 77(1), 2255-2269 (2014).
[7] Lin, H. Wen, F., and Du, C., A novel and anonymous key agreement multi-server architecture, Journal of Computational Information Systems, 11(8), 3011-3018 (2015).
[8] Li, C.T., and Hwang, M.S., An efficient biometrics based remote user authentication scheme using smart cards, Journal of Network and Computer Applications, 33(1), 1-5. (2010)

[9] Chuang, M.C., Chen, M.C., An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics, Expert Systems with Applications, 41(4), 1411-1418 (2014).
[10] Mishra, D., Das, A.K., and Mukhopadhyay, S., A secure user anonoymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards, Expert Systems with Applications, 41(18), 8129-8143 (2014).
[11] Das A.K., Analysis and improvement on an efficient biometricbased remote user authentication scheme using smart cards, IET Information Security, 5(3), 145–151 (2011).
[12] [An, Y., Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards, Journal of Biomedicine and Biotechnology, 6 pages (2012).
[13] Khan, M.K. and Kumari, S, An improved biometrics-based remote authentication scheme with user anonymity, Journal of Biomedicine and Biotechnology, 9 pages (.2013).
[14] Wen, F., Susilo, W., and Yang, G., Analysis and improvement on a biometric-based user authentication scheme using smart cards, Wireless Personal Communications, 80, 1747-1760 (2015).

Author

An Braeken, she obtained her MSc Degree in Mathematics from the University of Gent in 2002. In 2006, she received her PhD in engineering sciences from the KULeuven at the research group COSIC (Computer Security and Industrial Cryptography). In 2007, she became professor at Erasmushogeschool Brussel (currently since 2013, VUB) in the Industrial Sciences Department. Her current interests include security protocols for sensor networks.

