Cryptanalysis of an Advanced Authentication Scheme

Sattar J Aboud
Department of Information Technology
Iraqi Council of Representatives
Baghdad-Iraq

Abid T. Al Ajeeli
Department of Information Technology
Iraqi Council of Representatives
Baghdad-Iraq

***Abstract*—In this paper we study a scheme for making cryptanalysis and security improvement. This protocol by Song, is a password authentication protocol using smart card. We note that this protocol has been shown to be prone to the offline password guessing attack. We perform an additional cryptanalysis on this scheme and detect that it is vulnerable to the clogging attack, a type of denial-of-service attack. We notice that all smart card typed authentication schemes which lead the scheme by Song, and need the server to find the computationally exhaustive modular exponentiation, similar to the scheme by Xu et al., and it is vulnerable to the clogging attack. Then we propose an enhancement in the scheme to avoid the clogging attack.**

Keywords-authentication protocol; offline password guessing attack; clogging attack.

I. INTRODUCTION

The idea behind improving password authentication protocol is to help authorized users obtain services from an authorized server. When an entity needs a service from a server, it has to identify itself to the server in a certain way. Password authentication has been one of the most suitable techniques for a user ID throughout the years. Currently, millions of providers utilize password authentication schemes to identify authorized users. General cases include private web service, Internet shopping, e-mail service, e-trade service, and other services. Fundamentally, each password authentication method has two steps:

- Registration step: In this step, the user enters a user ID and password in the computer. The password is saved by the server and kept confidential between the entity and the computer.
- Authentication step: In this step, the entity needs a service from the computer. It passes its identity and password to the computer to get a service. The computer then decides if the user is authorized by checking the received information of user ID password with the saved details. The server extends the preferred service to the entity, if found authorized

In authentication processes, the password is sent cross an insecure communication channel. A hacker can intercept the message by listening to the communication. It can imitate the entity by re-using the password acquired from the communication. These issues compromise the entity confidentiality. The service providers normally keep passwords of users in a database for potential verification and authentication. The passwords are kept in a password index in the computer database. It does not provide any security against unprivileged insiders of the computer, and it does not protect the passwords when the computer database is someway hacked. To reduce the problem of password list disclosure, the computer can encrypt the passwords and protect them. Nevertheless, the communication interception remains a threat to the organization security. Another problem is recalling a user identity and password. A compromise in the password for any entity can be like losing a credit card. The hacker can take advantage before the corporation is informed about the loss. Therefore we have to improve efficient password authentication protocol to finish the authentication process in a secured way.

Taking this into consideration, and to produce a more secure scheme, various smart cards typed password authentication schemes have been proposed throughout the last decade [4, 6, 7]. In such a system, the entity is given with a smart card. When the user needs a service, it gives its smart card with a password that remains private. The smart card then employs this password to build a login message that is passed to the computer. The computer then authenticates this message and gives the preferred service if the password is found legitimate.

In this paper, we study such a scheme under smart card by Song [1]. Song presented the scheme as an enhancement of another scheme set is by Xu et al. [3]. The scheme has been shown to be prone to the imitation attack. Song then considered this competitive scheme in the same study, to avoid the imitation attack on the scheme. In this paper, we will show that the Song protocol is prone to the clogging attack, a type of the denial-of-service-attack. We noted that the scheme in [3] was vulnerable to clogging attack and also prone to clogging attack. In this attack, the hacker can easily prevent the computer from giving any service without having any information related to the identity or password of the entity. The hacker wants to make any complex computations to launch a clogging attack on Song scheme. We have come to know a cryptanalysis of the Song scheme by Tapiador et al [2]. However, they did not study the clogging attack in the scheme in their research. It also worth mentioning that the chain of smart card typed authentication schemes that keeps the server computationally exhaustive modular exponentiation [5] are all prone to the clogging attack.

The hacker takes the benefit of the calculation intensiveness of the modular exponentiation computation in initiating this attack. To avoid clogging attack on these schemes key agreement schemes are required in different

types of communication. Keys want to be securely exchanged before a channel can be recognized. There are few security threats to this that are intruder-in-the-middle so that the hacker pretends to be someone else than connecting participants. Replay of old keys is one more attack which is common in this viewpoint. Therefore it is necessary to improve secure key exchanging schemes to create secure channel. The key agreement protocols commonly have two steps:

- Determine the public and private keys: In this step, both the participants compute a pair of keys: the secret key, which is kept private, and the public key, which is made public. In some schemes, a part of this step is completed by a key distribution center that keeps the public keys of the entities in a directory.
- Determine the secret session key: In this step, the users swap their public keys with a certain integer. The secret session key for message is computed using those integers, and the secret and public keys. Several schemes also let a certain participant determine the private key or session key and pass it to the other participant by encrypting it with the message. In this paper we will not study any scheme regarding the key agreement protocols.

The rest of the paper is organized as follows: in section 2, we study the Song scheme including some an enhancement we suggest concerning public and private keys ; then we give a toy example in section 3. Then we have a cryptanalysis of Song scheme .including clogging attack and offline password guessing attack in section 4. We then propose a solution to this attack in section 5. In section 6, we conclude this work.

II. Review of Song Scheme

We summarize here the password authentication protocol of Song [1]. This authentication protocol uses a smart card. In section 3, we consider the achievement and security vulnerability of this protocol. In section 4, we introduce a clogging attack on the scheme. We also consider a possible solution against the attack. Finally, we show that similar protocols of [3], that let the server calculate modular exponentiation, are vulnerable to the same type of attack and has the same achievement dependencies as this protocol. The Song protocol contains three steps: registration, login and authentication. Prior to starting with the registration phase, the server performs the following steps:

1. two primes p and q are chosen where $p = 2*q+1$.
2. an integer $i \in Z_q^*$ is chosen.
3. a hash function h is chosen.
4. an integer e is chosen as encryption key such that $\gcd(e, p-1) = 1$.
5. the decryption key d is computed, where $d = e^{-1} \bmod p-1$.
6. i and d are determined as both are private keys.

A. Registration Phase

This phase comprises the following steps:

1. Entity A selects (id_A, w_A)
2. Entity A passes (id_A, w_A) to the server over a secure way, such that id_A and w_A are the user-id and password of entity A respectively.
3. The server calculates $z_A = h(id_A^i \bmod p) \oplus h(w_A)$.
4. The server saves the parameter (id_A, z_A, h, e) into a smart card.
5. The server sends the smart card to the entity A.

B. Logical Phase

Entity A performs the following steps:

1. Provides its (id_A, w_A).
2. Chooses an arbitrary integer r_A.
3. Sets $T_A \leftarrow$ system present time.
4. Finds $y_A = z_A \oplus h(w_A)$.
5. Computes $x_A = e_{y_A}(r_A \oplus T_A)$.
6. Computes $u_A = h(T_A || r_A || x_A || id_A)$.
7. Sends the message (id_A, u_A, x_A, T_A) to the server.

C. Authentication Phase

The following steps are performed by the server:

1. the server performs the following after receiving the login request from an entity:
   - Verifies that id_A, and T_A. If not, rejects the login message.
   - Finds $y_A = h(id_A^i \bmod p)$.
   - Computes $r_A^{'} = d_{y_A}(x_A) \oplus T_A$
   - Verifies that $u_A \equiv h(T_A || r_A^{'} || x_A || id_A)$. Otherwise rejects the login message.
   - Finds $u_S = h(id_A || r_A^{'} || T_S)$,
   - Passes the message (id_A, u_S, T_S) to an entity A.
2. The following steps are performed by an entity A after the receipt of (id_A, u_S, T_S) from a server:
   - authenticates id_A and T_S
   - Checks that $u_S \equiv h(id_A || r_A || T_S)$. If identical, the server is validated.
3. The entity A and the server then find the session key s as follows:
   - Entity A compute the session key as follows $s = h(id_A || T_S || T_A || r_A)$
   - The server computes the session key as follows $s = h(id_A || T_S || T_A || r_A^{'})$.
   - Finally, a session key s is agreed by the two participants.

D. Comments

When the entity wants to alter the password w_A to a new password $w_A^{'}$, the smart card will first verify the validity of

w_A through interacting with the server, and if it succeeds, it resets w_A to $w_A^{'}$,and substitutes z_A with $z_A^{'} = z_A \oplus w_A \oplus w_A^{'}$.

III. TOY EXAMPLE YOUR

Suppose that the server selects $p = 23$, then $q = 11$. Suppose that $i = 2$, and for simplicity ignore in this phase the hash function $h(.)$. Suppose the encryption key $e = 7$, and then compute the decryption key d as follows $e * d \equiv 1 \bmod p - 1 = 7 * 63 \bmod 22 = 1$.

Registration Phase: Suppose that entity A selects $(id_A = 29, w_A = 24)$, then entity A passes $(id_A = 29, w_A = 24)$ to the server over a secure way so that id_A and w_A are the user-id and password of entity A. Then, the server calculates $z_A = h(id_A^i \bmod p) \oplus h(w_A)$ $= h(29^2 \bmod 23) \oplus h(24) = 21$. The server then saves the parameter $(id_A = 29, z_A = 21, h = (.), e = 7)$ into a smart card. Finally, the server sends the smart card to the entity A.

Login Phase: First, the smart card is attached to the card reader by an entity A and it provides $(id = 29, w_A = 24)$. The smart card then chooses a random integer $(r_A = 12)$ and then sets $(T_A = 4)$. Then entity A computes $y_A = z_A \oplus h(w_A) = 21 \oplus h(24) = 13$. Then compute $x_A = e_{y_A}(r_A \oplus T_A) = 13(12 \oplus 4) = 12$. compute $x_A = 12^7 \bmod 23 = 16$. Then compute $u_A = h(T_A || r_A || x_A || id_A)$ $= h(4 + 12 + 16 + 29) = 61 \bmod 23 = 15$. Then entity A sends the message $(id_A = 29, u_A = 15, x_A = 16, T_A = 4)$ to the server.

Authentication Phase: First, the server verifies that id_A is valid. If not, it rejects the login message. Then, $y_A = h(id_A^i \bmod p) = h(29^2 \bmod 23) = 13$ is computed. Then $r_A^{'} = d_{y_A}((x_A^d \bmod p) \oplus T_A) =$ $13((16^{63}) \bmod 23 = 12) = (12 \oplus 4) = 8 * 13 \bmod 23 = 12$ is computed. Then, the server verifies that $u_A \equiv h(T_A || r_A^{'} || x_A || id_A)$ $= h(4 + 12 + 16 + 29) = 61 \bmod 23 = 15$. If the verification of the previous step succeeds, set T_s as current server time. Suppose that $T_s = 4$, and then compute $u_S = h(id_A || r_A^{'} || T_S) = h(29 + 12 + 4) = 45 \bmod 23 = 22$. Finally, the server passes the message $(id_A = 29, u_S = 22, T_S = 4)$ to entity A. An entity A authenticates id_A and T_S by Checking that $u_S \equiv h(id_A || r_A || T_S)$ $= h(29 + 12 + 4) = 22$. In this example, it is identical; it means that the server is validated. However, the entity A and the server then find the session key s as follows: first, entity A computes the session key as follow $s = h(id_A || T_S || T_A || r_A)$ $= h(29 + 4 + 4 + 12) = 49 \bmod 23 = 3$ .Then, the server computes the session key as follows $s = h(id_A || T_S || T_A || r_A^{'}) = h(29 + 4 + 4 + 12) = 49 \bmod 23 = 3$.

IV. SONG SCHEME CRYPAYLISIS

The clogging attack and offline password guessing attack will be discussed in this section.

A. The Clogging Attack

This scheme has a large dependency on the computer and entity clocks. For a connection-oriented use, this may be unwieldy. The scheme should be designed to care for time synchronization between clocks of different entities and servers. This scheme should be made fault tolerant to deal with complex network faults and also with different types of attacks. Despite that the communication is secure, a chance of an attack can occur and a hacker may intercept a message and alter its timestamp T_A. In this way the hacker successfully repudiates the authorized entity because the server will refuse the login message on the basis of timestamp dissimilarity. Thus this type of attack is probable even if the scheme avoids replay attacks. Also, interacting delays can prepare the timestamp to go beyond the threshold thus making the entire service inherently decelerate. Therefore, we illustrate that Song scheme is vulnerable to the clogging attack. The clogging attack is a type of attack by which the hacker H constantly passes messages to a server and clogs it with those messages [8]. Suppose this could occur with the Song scheme. The following is done by a hacker H :

1. H intercepts the message $(id_A = 29, u_A = 11, x_A = 12, T_A = 4)$ passed by an entity to a server in the login phase.
2. H can alter the timestamp $T_A = 4$ to some $T_u = 10$, because the message is unencrypted. The change satisfies the criterion $T^* - T_u \le g$.
3. H alters $u_A = 11$ to arbitrary nonsense value $u_u = 9$.
4. H passes $(id_A = 29, u_u = 9, x_A = 12, T_u = 10)$ to the server.

The following is performed by the server:

1. Verify if $id_A = 29$ is valid. At this point it is valid.
2. Find $y_A = h(id_A^i \bmod p) = h(29^2 \bmod 23) = 13$.
3. Find $r_A^{'} = d_{y_A}(x_A \oplus T_u) = 13(12 \oplus 10) = 9$
4. verify $u_u = h(T_u || r_A^{'} || x_A || id_A)$ $= h(10 + 9 + 12 + 29) = 60 \bmod 23 = 14$. This does not succeed, thus the message is rejected.

Then, the hacker H will continue repeating the steps many times and let the server calculate the modular exponentiation repeatedly. Essentially, H can potentially alter all the entering login requests from the authorized entity to the server. As modular exponentiation is computationally exhaustive, the victimized server spends large processing resources doing ineffective modular exponentiation rather than any actual work. Therefore the hacker H clog the server with ineffective work and so repudiates any authorized entity. The hacker only wants an id of a valid entity to achieve the clogging attack many times.

B. Offline Password Gussing Attack

In [1] it is claimed that the hacker should not be able to attack and get access to the server by extracting the information kept on the smart card. But, a hacker who gets the result of $z_A = h(id_A^i \bmod p) \oplus h(w_A)$ can simply increase an offline password guessing attack by easily viewing one right authentication session and obtaining access to the results of x_A , and u_A. The attack is performed as follows: for every password w_A^*, the hacker calculates the uncertain encryption key $y_A^* = z_A \oplus h(w_A^*)$. Such a key is then employed to decrypt the nonce result r_A^* by first recovering x_A with y_A^* and then $\oplus$ the value with T_A (both of which are public); namely, $r_A^* = d_{y_A}^*(x_A) \oplus T_A$. Note that when an attempt of password w_A^* is right (i.e., $w_A^* = w_A$), then it is obtained encryption key y_A^* and so, the nonce r_A^* . Now, u_A can be used to verify when that is the case. The hacker finds $u_A^* \equiv h(T_A || r_A^* || x_A || id_A)$ and, when it coincides with u_A, it can conclude that r_A^* is right and thus the password tried. In this reasoning we suppose that h has no collisions. Yet, even when h is not perfect, extra eavesdropping sessions can be employed to exclude false positives and find the right password. Briefly, in addition to what is claimed in [1], the messages exchanged during the scheme certainly decrease the entropy of the password, at least for a hacker with access to the values stored in the card. Also, once the password is guessed, the scheme provides no protection against other attacks.

C. Comments

It can be observed that the attack showed can also be made on the schemes by Xu et al. [3] and by Tsaur et al. [5]. Thus we notice that the clogging attack can be executed on all the smart card typed authentication schemes using computing modular exponentiation.

V. The Possible Solution

We will discuss the possible solution for the problem raised.

A. Prevent The Clogging Attack

At the start of the authentication phase, the server will check if the *IP* address of the entity is valid. It has to identify the *IP* addresses of any registered authorized users. Despite that, hacker *H* might spoof the *IP* address of an authorized entity and replay the login request. To stop it, we may add a cookie exchange step at the start of the login phase of Song scheme. This step has been presented as in the familiar Oakley key exchange scheme [9].

1. The entity *A* selects an arbitrary number m_1 and passes it with the message (id_A, u_A, x_A, T_A) to the server.
2. The server accepts the message and passes its own cookie m_2 to the entity A.

B. Solution Discussion

When the hacker *H* spoofs the entity *IP* address, *H* will not obtain m_2 back from the server. But *H* just succeeds to have the server return an acknowledgement, not to calculate the computationally modular exponentiation. Thus the clogging attack is prevented by these extra steps. We note that this process does not avoid the clogging attack but only frustrates it to a certain extent.

VI. Conclusion

We have studied a scheme in this paper. The scheme is a password authentication protocol; which we have shown to be vulnerable to the clogging attack. We demonstrated that the attack on this scheme could be prevented by using an extra step of exchanging numbers. We demonstrated that it is prone to man-in-the-middle attack. Then we showed how to prevent this attack by using an encryption and decryption algorithm. We indicated a security get-out as Song proposed, which is that the hacker can execute modular exponentiation on both sides of the authentication scheme. In addition, after intercepting the retrieved information, the hacker can start new logon information and successfully log into the server system. Thus, Song proposition cannot give adequate security and it is not appropriate for practical implementation of the proposed scheme.

References

[1] Ronggong Song, "Advanced smart card based password authentication protocol", Computer Standards & Interfaces, Volume 32, Issue 4, pp. 321-325, June 2010.

[2] Juan E. Tapiador, Julio C. Hernandez-Castro, Pedro Peris-Lopez, John A. Clark, "Cryptanalysis of Song's advanced smart card based password authentication protocol", Unpublished manuscript, June 2010.

[3] Jing Xu,Wen-Tao Zhu, and Deng-Guo Feng, "An improved smart card based password authentication scheme with provable security", Computer Standards & Interfaces, Volume 31, Issue 4, pp. 723-728, June 2009.

[4] Chang. C.C., Wu T.C., "Remote password authentication scheme with smart cards", IEEE Proceedings Computers and Digital Techniques, Volume138, Issue 3, pp.165–168, 1991.

[5] Woei-Jiunn Tsaur, Chia-Chun Wu, and Wei-Bin Lee, "A smart card based remote scheme for password authentication in multi-server Internet services", Computer Standards & Interfaces, Volume 27, pp. 39-51, June 2004.

[6] Wen-Shenq Juang, "Efficient password authenticated key agreement using smart cards", Computers & Security, Volume 23, Issue 2, pp. 167-173, March 2004.

[7] Chien H.Y., Jan J.K., and Tseng Y.M., "An efficient and practical solution to remote authentication: smart card", Computers and Security, vol.21, no.4, pp.372–375, 2002.

[8] Tseng Y.M., "An efficient two-party identity-based key exchange protocol", Informatica 18 (1) pp. 125–136, 2007.

[9] Hsi-Chang Shih, "Cryptanalysis on Two Password Authentication Schemes", Master Thesis, Laboratory of Cryptography and Information Security Department of Computer Science and Information Engineering, National Central University, Chung-Li, Taiwan 320, Republic of China, 2006.

[10] Meshram, C. (2010). Modified ID-Based Public key Cryptosystem using Double Discrete Logarithm Problem. IJACSA - International Journal of Advanced Computer Science and Applications, 1(6).

[11] Meshram, C. (2011). Some Modification in ID-Based Public key Cryptosystem using IFP and DDLP. IJACSA - International Journal of Advanced Computer Science and Applications, 2(8).

Authors Profile

Sattar J Aboud is a Professor and advisor for Science and Technology at Iraqi Council of Representatives. He received his education from United

Kingdom. Dr. Aboud has served his profession in many universities and he awarded the Quality Assurance Certificate of Philadelphia University, Faculty of Information Technology in 2002. Also, he awarded the Medal of Iraqi Council of Representatives for his conducting the first international conference of Iraqi Experts in 2008. His research interests include the areas of both symmetric and asymmetric cryptography, area of verification and validation, performance evaluation and e-payment schemes.

Abid T. Al Ajeeli is a Professor at Iraqi Council of Representatives. He received his education from United Kingdom. Dr. Ajeeli has served his profession in many universities and he awarded the Medal of Iraqi Council of Representatives for his conducting the first international conference of Iraqi Experts in 2008. His research interests include the areas of software engineering, verification and validation, information security and simulation.