Formal Analysis and Systematic Construction of Two-Factor Authentication Scheme (Short Paper)

Guomin Yang[1], Duncan S. Wong[1,⋆], Huaxiong Wang[2], and Xiaotie Deng[1]

[1] Department of Computer Science
City University of Hong Kong
Hong Kong, China
{csyanggm, duncan, deng}@cs.cityu.edu.hk
[2] Department of Computing
Macquarie University
Australia
hwang@ics.mq.edu.au

Abstract. One of the most commonly used two-factor authentication mechanisms is based on smart card and user's password. Throughout the years, there have been many schemes proposed, but most of them have already been found flawed due to the lack of formal security analysis. On the cryptanalysis of this type of schemes, in this paper, we further review two recently proposed schemes and show that their security claims are invalid. To address the current issue, we propose a new and simplified property set and a formal adversarial model for analyzing the security of this type of schemes. We believe that the property set and the adversarial model themselves are of independent interest.

We then propose a new scheme and a generic construction framework. In particular, we show that a secure password based key exchange protocol can be transformed efficiently to a smartcard and password based two-factor authentication scheme provided that there exist pseudorandom functions and collision-resistant hash functions.

1 Introduction

Password authentication with smart card is one of the most convenient and effective *two-factor authentication* mechanisms. This technology has been widely deployed for various kinds of authentication applications which include remote host login, online banking, access control of restricted vaults, activation of security devices, and many more. Although some smart-card-based password authentication systems have already been in use, many of them are having issues on both security and performance aspects.

A smart-card-based password authentication scheme involves a server S and a client A with identity ID_A. At the very beginning, S issues a smart card to

⋆ The author was supported by a grant from CityU (Project No. 7001959).

P. Ning, S. Qing, and N. Li (Eds.): ICICS 2006, LNCS 4307, pp. 82–91, 2006.

A with the smart card being personalized with respect to ID_A and some initial password. This phase is called the *registration phase* and is carried out only once for each client in some secure way. After obtaining the smart card, A can access S in the *login-and-authentication phase*. This phase can be carried out as many times as needed. However, in this phase, there could have various kinds of passive and active adversaries in the communication channel between A and S. They can eavesdrop messages and even modify, remove or insert messages into the channel. The security goal of the scheme in this phase is to ensure mutual authentication between A and S in the presence of these adversaries. In particular, it is required to both *have* A's smart card and *know* A's password in order to carry out the smart-card-based password authentication scheme successfully with server S, that is, maintaining *two-factor security* that the scheme should provide. There are also some other desirable properties people would like the scheme to possess. We will discuss these properties shortly.

Besides registration phase and login-and-authentication phase, A may want to *change password* from time to time. Conventionally, this activity usually has S involved and requires S to maintain a database for storing the passwords or some derived values of the passwords of its clients. In this paper, we promote the idea of letting A change the password at will without interacting with or notifying S (while ensuring two-factor security), and also eliminating any password database at the server side.

Current systems also suffer from other potential security vulnerabilities. One prominent issue is security against *offline guessing attack* (also known as offline dictionary attack). The purpose of offline guessing attack is to compromise a client's password through exhaustive search of all possible password values. In the context of a password-based cryptosystem, we consider that passwords are short in the sense that they are human memorizable. In other words, we assume that the password space is so small that an adversary is able to enumerate all possible values in the space within some reasonable amount of time.

A stronger notion of security against offline guessing attack is to require that compromising a client's smart card does not help the adversary launch offline guessing attack against the client's password. In practice, the adversary may steal the smart card and extract all the information stored in it through reverse engineering. This notion is reminiscent of password-based key exchange protocols [6]. The difference is that for password-based key exchange protocols, the focus is on preventing adversaries from getting any useful information about the password mainly from the transcripts of protocol runs, while for smart-card-based password authentication schemes, in addition to thwarting related attacks against password-based key exchange protocols, we also need to protect the password from being known even after the client's smart card is compromised.

Since Lamport [9] introduced a remote user authentication scheme in 1981, there have been many smart-card-based password authentication schemes proposed (some recent ones are [2,14,15,10]). These schemes are aimed for different security goals and properties, and noticeably, there is no common set of desirable security properties that has been widely adopted for the construction of this

type of schemes. Although the construction and security analysis of this type of schemes have a long history, recently proposed schemes are still having various security weaknesses being overlooked, and we can find many of these schemes broken shortly after they were first proposed [4,5,12,11,15].

1.1 Our Results

In this paper, we contribute on three areas:

1. We propose a new and simplified set of desirable security properties for a smart-card-based password authentication scheme. We also propose an adversarial model for formal analysis of the security of this type of schemes.
2. We show that two recently proposed schemes are insecure with respect to their claimed security properties which have also been captured in our desirable property set.
3. We propose a generic construction framework and show that a secure smart-card-based password authentication scheme can be constructed by *transforming* a proven secure password based key exchange protocol (under some appropriate security model which will be specified) provided that there exist pseudorandom unctions and collision-resistant hash functions. The transformation is very efficient. It essentially adds in only two additional hash evaluations and one pseudorandom function evaluation.

Paper Organization: In Sec. 2, we propose a set of desirable properties and an adversarial model for smart-card-based password authentication schemes. In Sec. 3, we review a scheme proposed by Liao et al. in [10] and show that the scheme is insecure. In Sec. 4, we propose a new scheme and show its security. In Sec. 5, we propose a generic construction framework that can be used to convert a proven secure password-based mutual authentication protocol to a smart-card-based password authentication scheme.

2 Security Properties

As introduced in Sec. 1, there are two phases and one activity in a smart-card-based password authentication system. The two phases are *registration phase* and *login-and-authentication phase*, and the activity is called *password-changing activity*.

In the registration phase, an authenticated and secure environment is assumed to present, and all parties are assumed to be honest and perform exactly according to the scheme specification. In the real world, this stage may require the client who is requesting for registration to show up in person at the server's office and then have a smart card initialized and personalized using a secure and isolated machine. The smart card is finally issued to the client at the end of the stage. After this phase is completed, the client is said to be *registered*. In the login-and-authentication phase, the communication channel between server

S and a registered client A is no longer considered to be secure. Both passive and active adversaries are present and their objective is to compromise the scheme's primary security goal, that is, mutual authentication between S and A. During the password-changing activity, a registered client A change the password and updates the smart card accordingly. A may need to interact with S for changing the password. However, this is undesirable due to the scalability issue and the concern of user friendliness. It will be better if A can change the password freely without the help or notification of S. In the following, we describe what we want a secure smart-card-based password authentication system to achieve (i.e. security goals / desirable properties) and what the capabilities of the adversary are (adversarial model).

2.1 Desirable Properties and Adversarial Model

Below are the five desirable properties that a smart-card-based password authentication system should achieve.

1. (*Client Authentication*) The server is sure that the communicating party is indeed the registered client that claims to be at the end of the protocol.
2. (*Server Authentication*) The client is sure that the communicating party is indeed the server S at the end of the protocol.
3. (*Server Knows No Password*) S should not get any information of the password of a registered client or anything derived from the password.
4. (*Freedom of Password Change*) A client's password can freely be changed by the client without any interaction with server S. S can be totally unaware of the change of the client's password.
5. (*Short Password*) The password space is small enough so that the underlying adversary can enumerate all the possible values of the space in a reasonable amount of time. We consider a human-memorizable password to be a value in this password space.

Adversarial Model. Consider an adversary $\mathcal{A}$ who has the full control of the communication channel between the server S and any of the *registered* clients. $\mathcal{A}$ can obtain all the messages transmitted between the server S and a registered client; $\mathcal{A}$ can also modify or block those transmitted messages; and $\mathcal{A}$ can even make up fake messages and send to any entity in the system while claiming that the messages are from another entity in the system (i.e. impersonation). To simulate insider attack [1], we also allow $\mathcal{A}$ to know the passwords and all information stored in the smart cards of all the clients except those of a client who is under attack from $\mathcal{A}$. In addition, we also allow $\mathcal{A}$ to *either* compromise the password *or* the smart card of the client under attack, but not both. However, $\mathcal{A}$ is not allowed to compromise S.

Discussions. In the list of desirable properties above, the first two constitute the primary security requirement of a secure smart-card-based password authentication scheme, that is, mutual authentication between the server S and a registered client A. The third property helps solve the scalability problem at

the server side. In addition, since there is no information about clients' passwords stored at the server side, the property also alleviate damage entailed to the clients if the server is compromised. The fourth property will help improve the user friendliness of the system as there is no additional communication overhead when a client changes her password. One should note that property 3 does not imply property 4. It is always possible to construct a scheme such that the server does not have any information of a client's password while the client cannot change the password either once after registration. The fifth property means that we always consider that if an adversary launches an attack which needs to search through the password space (for example, an offline guessing attack), the adversary can always evaluate all the possible values in the space within the running time of the adversary. To prevent an adversary from launching offline guessing attack, we therefore need to make sure that the scheme is not going to leak any information useful about the client's password to the adversary, even though the password is considered to be weak and low-entropy.

Note that the adversary can always launch the *online* guessing attack. In this attack, the adversary impersonates one of the communicating parties and sends messages based on a trial password chosen by the adversary. If the trial password is guessed incorrectly, the other party will reject the connection. If so, the adversary will try another password and repeat the steps until a trial password leads to an acceptance of connection. Online guessing attack is easy to defend against in practice. Conventionally, a system can set up a policy mandating that if the password of a client is entered incorrectly for three times in a row, then the client will be blocked and refused to connect any further. This policy works well in practice and can effectively defend against online guessing attack if the attack only allows the adversary to try one password in each impersonation attack. However, we should also note that a secure scheme should not allow the adversary to test two passwords or more in each of this impersonation attack.

In our full paper [13], we also present a comparison between our model and a set of requirements for smart-card-based password authentication schemes recently proposed by Liao et al. [10].

3 Offline Guessing Attack Against a Smart-Card-Based Password Authentication Scheme

In this section, we show that the scheme proposed by Liao et al. [10] is insecure against offline guessing attack. In our full paper [13], we show that another scheme recently proposed by Yoon and Yoo [15] is insecure either.

Here are the notations that we will use for describing Liao et al.'s scheme. Let p be a 1024-bit prime. Let g be a generator of $\mathbb{Z}_p^*$. The server S chooses a secret key x. In [10], the authors did not specify the length of x, however, in order to prevent brute-force search, we assume x to be a random string of at least 160 bits long. Let h be a hash function (e.g. SHA-256) and $a\|b$ denote the concatenation of a and b.

Registration phase: Server S issues a smart card to a client A as follows.

1. A arbitrarily chooses a *unique* identity ID_A and password PW_A. PW_A is a short password that is appropriate for memorization. A then calculates $h(PW_A)$ and sends $(ID_A, h(PW_A))$ to S.
2. S calculates $B = g^{h(x\|ID_A)+h(PW_A)} \bmod p$ and issues A a smart card which has (ID_A, B, p, g) in it.

Login-and-authentication phase: A attaches the smart card to an input device and keys in ID_A and PW_A. Afterwards, S and A (the smart card) carry out the following steps.

1. A sends a login request to S.
2. On receiving the login request, S calculates $B'' = g^{h(x\|ID_A)R} \bmod p$ where $R \in \mathbb{Z}_p^*$ is a random number, and sends $h(B'')$ and R to A.
3. Upon receiving the message from S, A calculates $B' = (Bg^{-h(PW_A)})^R \bmod p$ and checks if $h(B'') = h(B')$. If they are not equal, S is rejected. Otherwise, A calculates $C = h(T\|B')$ where T is a timestamp, and sends (ID_A, C, T) to S.
4. Let T' be the time when S receives (ID_A, C, T). S validates A using the following steps.
   (a) S checks if ID_A is in the correct format[1]. If it is incorrect, S rejects.
   (b) Otherwise, S compares T with T'. If $T' - T \geq \Delta T$, S rejects, where ΔT is the legal time interval for transmission delay.
   (c) S then computes $C' = h(T\|B'')$ and checks if $C = C'$. If they are not equal, S rejects. Otherwise, S accepts.

3.1 Offline Guessing Attack

Malicious user offline guessing attack. In [10], the scheme above is claimed to be secure against offline guessing attack even if the client's smart card is compromised. In the following, we show that this is not true. Suppose client A's smart card is compromised by an adversary $\mathcal{A}$. $\mathcal{A}$ can carry out the offline guessing attack as follows.

1. $\mathcal{A}$ impersonates A and sends a login request to S.
2. S calculates $B'' = g^{h(x\|ID_A)R} \bmod p$ and sends back $(h(B''), R)$.
3. $\mathcal{A}$ then carries out offline guessing attack by checking if

$$h(B'') = h((Bg^{-h(PW_A^*)})^R \bmod p)$$

   for each trial password PW_A^* (i.e. $\mathcal{A}$'s guess of PW_A).

[1] In [10], the format of identity ID_A was not given. We hereby assume that there is some pre-defined format for all the identities used in their system.

Note that after $\mathcal{A}$ receives the message from S in step (2), $\mathcal{A}$ does not need to provide any response to S and therefore S does not know whether the communicating party is launching an attack or simply the message sent by S is lost during transmission. This makes the guessing attack described above difficult to detect. Also notice that if $\mathcal{A}$ possesses a past communication transcript Trans between A and S, $\mathcal{A}$ can perform the offline guessing attack directly without interacting with S.

4 A New Scheme

In this section, we propose a new smart-card-based password authentication scheme which is proven secure and also satisfies all the properties we described in Sec. 2. This new scheme can also be extended to a generic construction framework which allows us to convert most of the proven secure password-based key exchange protocols [6] to smart-card-based versions. The significance of this framework is that we can now design provably secure smart-card-based password authentication scheme in a systematic way and make use of those proven secure password-based key exchange protocols as the main building blocks. The schemes constructed in this framework will also have session keys established that are generally useful for target applications. More details of the generic construction framework will be given in Sec. 5. In this section, we focus on describing how the new scheme is constructed.

In [3], Halevi and Krawczyk defined a security model for password-based authentication and also proposed a protocol of this type. The definition of security in this model essentially says that the "best" possible strategy for the adversary to compromise user authentication is online guessing attack, which can be thwarted in practice by limiting the number of consecutive authentication failures that each user is allowed. Based on the Halevi-Krawczyk one-way password-based authentication protocol, we build a proven secure password-based authenticated key exchange (PWAKE) protocol, and then "upgrade" the PWAKE protocol to our final smart-card-based password authentication scheme. Here we merely present the PWAKE protocol and the final smart-card-based password authentication scheme, for all the details, readers can refer to our full paper [13].

A PWAKE Protocol. Let G be a subgroup of prime order q of a multiplicative group $\mathbb{Z}_p^*$. Let g be a generator of G. Let $(\mathrm{PK}_S, \mathrm{SK}_S)$ denote a public/private key pair of the server S. User A has a password PW_A which is shared with S.

$$
\begin{aligned}
&A \rightarrow S : A, sid, g^{\hat{x}} \\
&A \leftarrow S : S, sid, g^{\hat{y}}, \mathrm{SIG}_{\mathrm{SK}_S}(S, A, sid, g^{\hat{x}}, g^{\hat{y}}) \\
&A \rightarrow S : A, sid, c = \mathrm{ENC}_{\mathrm{PK}_S}(PW_A, A, S, sid, g^{\hat{x}}, g^{\hat{y}})
\end{aligned}
$$

The session key is calculated as $\sigma = g^{xy}$.

4.1 A Smart-Card-Based Password Authentication Scheme

Notations: let p, G, g, q be the group parameters defined as above. Besides a public/private key pair $(\mathrm{PK}_S, \mathrm{SK}_S)$, the server S also maintains a long-term secret x which is a random string of length k. Let $H : \{0,1\}^* \rightarrow \{0,1\}^k$ denote a collision resistant hash function and $PRF_K : \{0,1\}^k \rightarrow \{0,1\}^k$ a pseudorandom function keyed by K.

Registration phase: Server S issues a client A as follows.

1. A arbitrarily chooses a unique identity ID_A and sends it to S.
2. S calculates $B = PRF_x(H(ID_A)) \oplus H(PW_0)$ where PW_0 is the initial password (e.g. a default such as a string of all '0').
3. S issues A a smart card which contains $\mathrm{PK}_S, ID_A, B, p, g, q$. In practice, we can have all these parameters except B be "burned" in the read-only memory of the smart card when the smart card is manufactured.
4. On receiving the smart card, A changes the password immediately by performing the password-changing activity (described below).

Login-and-authentication phase: A attaches the smart card to an input device, and then keys in ID_A and PW_A. The smart card checks if the identity is equal to the value stored in it. If not, the smart card will refuse carrying out any further operation. Otherwise, the smart card retrieves the value $LPW = B \oplus H(PW_A)$. A (actually performed by the client's smart card) and S then use LPW as the password to perform the PWAKE protocol.

Password-changing activity: If A wants to change the password, A carries out the following steps.

1. Select a new password PW'_A.
2. Compute $Z = B \oplus H(PW_A) \oplus H(PW'_A)$, where PW_A is the old password.
3. Replace B with Z in the smart card.

Remarks: The "password" used in the login-and-authentication phase is LPW, instead of the real password PW_A. Note that S can compute the value of LPW once after receiving ID_A. Hence it does not violate property 3 (Server Knows No Password) in Sec. 2. From the password-changing activity above, it is obvious that the scheme also satisfies property 4 (Freedom of Password Change).

In the two-factor security, we do not consider the case that both the password and the smart card are compromised, but we need to consider the other three cases: (1) neither the password nor the smart card is compromised; (2) the password is leaked while the smart card remains secure; (3) the smart card is compromised but the password remains secure. It is obvious that security under case (1) can be ensured if security under either case (2) or case (3) is guaranteed. And our goal is to achieve security under both case (2) and case (3). In other words, compromising one factor should not affect the other.

Case (2) Security. If the smart card is not compromised (even when the password is leaked), our proposed scheme deduces the success probability of the adversary to a negligible level by assuming that pseudo-random functions exist.

Theorem 1. *If the smart card is not compromised, and $PRF_K(\cdot)$ is a pseudorandom function, then the adversary has only a negligible success probability in the Halevi-Krawczyk model.*

The proof is given in our full paper [13].

Case (3) Security. If the smart card is compromised while the password remains secure, there is no security "upgrade" when compared with a password-based protocol. It is easy to see that if $PRF_K(\cdot)$ is replaced by a random function, then the protocol provides the same security as the password protocol. And by using the same approach as in the proof of Theorem. 1, we can show that our scheme provides almost the same security level (with at most a negligible gap) when compared with the password-based protocol.

5 A Generic Construction Framework

Up to this point, readers may have already realized that a smart-card-based password authentication scheme can readily be built from a proven secure password-based mutual authentication protocol by applying the *upgrading technique* of Sec. 4.1. The resulting scheme will then be secure under a model similar to the security model for the original password-based protocol, but extended according to the discussions in Sec. 4.1.

For example, we may choose an efficient password-based mutual authentication (and key exchange) protocol, such as [8,7], then we "upgrade" it to an efficient smart-card-based password authentication scheme using the technique described in Sec. 4.1. Interestingly, both of the protocols in [8,7] are proven secure without random oracle. Our upgrading technique does not rely on random oracle either. The "upgraded" smart-card-based scheme will then be secure with security statements similar to that of Theorem 1 (but now in the corresponding model of the original password-based authentication protocol) and also with respect to Case (2) as well as Case (3) Security. We refer readers to [6] for other examples of password-based mutual authentication (and key exchange) protocols.

Efficiency. The "upgrading" technique proposed in Sec. 4.1 is very efficient. During the login-and-authentication phase, the smart card only needs to carry out one pseudorandom function evaluation and two hashes in addition to the operations incurred by the underlying password-based protocol. The generic construction framework allows us to choose a password-based protocol which is efficient enough when implemented on smart cards.

A Practical Issue. In the description above, we consider the server S to maintain one single long-term secret x for communication with all the clients. As a result, the secrecy of x is utmost important because the security of the entire system essentially relies on the security of x. In practice, we can alleviate the damage caused to a system by using multiple values of x to partition the system, and in each partition, a randomly generated x is used by a disjoint set of clients. Each partition is to be handled by a distinct and independent server. Compromising one server will therefore only affect the security of the corresponding

partition of clients rather than the entire system. Note that this partitioning method does not affect the fulfillment of any of the desirable properties for a secure smart-card based password authentication scheme proposed in Sec. 2. Another mechanism which can be used in conjunction with the mechanism above is to set each long-term secret x with a validity period. Usually, smart cards are used such that they are valid only for a period of time. Hence for a different period of time, a fresh long-term secret x can be used.

References

1. C. Boyd and A. Mathuria. *Protocols for Authentication and Key Establishment*. Springer-Verlag, 2003.
2. H. Y. Chien, J. K. Jan, and Y. M. Tseng. An efficient and practical solution to remote authentication: Smart card. *Computers and Security*, 21(4):372–375, 2002.
3. S. Halevi and H. Krawczyk. Public-key cryptography and password protocols. *ACM Trans. Inf. Syst. Secur.*, 2(3):230–268, 1999.
4. M.-S. Hwang. Cryptanalysis of remote login authentication scheme. *Computer Communications*, 22(8):742–744, 1999.
5. M.-S. Hwang, C.-C. Lee, and Y.-L. Tang. An improvement of SPLICE/AS in WIDE against guessing attack. *Internat. J. Inform.*, 12(2):297–302, 2001.
6. IEEE. *P1363.2 / D23: Standard Specifications for Password-based Public Key Cryptographic Techniques*, March 2006. Available at `http://grouper.ieee.org/groups/1363/passwdPK/draft.html`.
7. S. Jiang and G. Gong. Password based key exchange with mutual authentication. In *11th International Workshop on Selected Areas in Cryptography (SAC 2004)*, pages 267–279. Springer-Verlag, 2005. LNCS 3357.
8. J. Katz, R. Ostrovsky, and M. Yung. Efficient and secure authenticated key exchange using weak passwords. *Journal of the ACM, to appear*, 2006.
9. L. Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–771, November 1981.
10. I-En Liao, Cheng-Chi Lee, and Min-Shiang Hwang. A password authentication scheme over insecure networks. *J. Comput. Syst. Sci.*, 72(4):727–740, 2006.
11. M. Scott. Cryptanalysis of an id-based password authentication scheme using smart cards and fingerprints. *SIGOPS Oper. Syst. Rev.*, 38(2):73–75, 2004.
12. B. Wang, J. H. Li, and Z. P. Tong. Cryptanalysis of an enhanced timestamp-based password authentication scheme. *Comput. Secur.*, 22(7):643–645, 2003.
13. G. Yang, D. S. Wong, H. Wang, and X. Deng. Formal analysis and systematic construction of two-factor authentication scheme. Cryptology ePrint Archive, Report 2006/270, 2006.
14. E. J. Yoon, E. K. Ryu, and K. Y. Yoo. Efficient remote user authentication scheme based on generalized elgamal signature scheme. *IEEE Transactions on Consumer Electronics*, 50(2):568–570, May 2004.
15. E.-J. Yoon and K.-Y. Yoo. New authentication scheme based on a one-way hash function and Diffie-Hellman key exchange. In *4th International Conference of Cryptology and Network Security (CANS 2005)*, pages 147–160. Springer-Verlag, 2005. LNCS 3810.