J. Shanghai Jiaotong Univ. (Sci.), 2014, 19(4): 436-439
DOI: 10.1007/s12204-014-1521-7

Provably Secure Authenticated Diffie-Hellman Key Exchange for Resource-Limited Smart Card

CHIEN Hung-yu (简宏宇)
(Department of Information Management, National Chi-Nan University, Nantou 542, Taiwan, China)

Abstract: Authenticated Diffie-Hellman key agreement is quite popular for establishing secure session keys. As resource-limited mobile devices are becoming more popular and security threats are increasing, it is desirable to reduce computational load for these resource-limited devices while still preserving its strong security and convenience for users. In this paper, we propose a new smart-card-based user authenticated key agreement scheme which allows users to memorize passwords, reduces users' device computational load while still preserves its strong security. The proposed scheme effectively improves the computational load of modular exponentiations by 50%, and the security is formally proved.

Key words: Diffie-Hellman, key agreement, forward secrecy, authentication, password
CLC number: TP 309.2 **Document code:** A

0 Introduction

User authentication (and remote user login) is one of those important security topics in our daily information activities, and it has been intensively studied for quite a long time. For user's convenience, password is still very popular in facilitating user authentication. For those authentication schemes that use passwords, we can further categorize them into two sets. One just uses passwords to activate some tokens like smart cards which pre-store some strong secrets and use the stored secrets to facilitate authentication with their servers; that is, the servers do not store users' password. Those smart-card-based remote authentication schemes like Refs. [1-5] belong to this category. However, those remote authentication schemes did not provide authenticated key agreement during authentication. The other set of protocols does store users' passwords in both the clients and the servers, and the communication protocols do involve some computation of passwords. Those schemes like Refs. [6-13] belong to this category.

Due to low entropy of passwords, several kinds of password guessing attacks threat the security of password-based authentication schemes. These password guessing attacks include off-line guessing attack, on-line guessing attack and on-line undetectable guessing attack[7-8]. In on-line detectable guessing attack, an attacker enumerates all the possible passwords, iteratively picks up one possible password to conduct on-line transactions with the server, and then verifies his guess using the response from the server. In on-line un-detectable guessing attack, the on-line un-detectable guessing attack happens when an attacker attempts to use a guessed password to start an on-line transaction. The attacker guesses the password, verifies the correctness of his guess by using the responses from the server until the correct password is found; however, during the trial attempts, the server cannot distinguish a malicious request from an honest request. In off-line guessing attack, an attacker iteratively picks up one possible password, and verifies his guess by using some collected data (either information eavesdropped during communications or encoded password files stolen from the server) in an off-line manner. Since the server is not involved in the attack scenarios, the attacker can repeatedly launch his attacks without arousing any attention of the server until he gets a hit.

For those authentication schemes that do not share passwords with servers, one common technique to conquer possible password-guessing attacks is to use user's password to activate a tamper-proof device like smart card which pre-stores some strong secrets and then run secure key agreement protocols like Diffie-Hellman key agreement to accomplish the authenticated key agreement. Since users' passwords do not involve in key agreement process, these key agreement protocols are resistant to password-guessing attacks.

For those authenticated key agreement schemes that

Received date: 2013-10-30
Foundation item: the National Science Council (No. NSC102-2221-E-260-011)
∗**E-mail:** hychien@ncnu.edu.tw

do involve users' passwords, the threat of password-guessing attacks should be considered and it is very challenging to design secure password-based authenticated key agreement schemes. The previous schemes like Refs. [6-13] belong to this category.

In many authenticated key agreement schemes, computational Diffie-Hellman problem (CDHP) plays the basic block for the security, owing to its non-polynomial (NP) property. However, Diffie-Hellman computation involves modular exponentiation which is costly for those resource-limited devices. As mobile devices like smart phones, ZigBee, radio frequency identification (RFID) and embedded chips are becoming more and more popular and ubiquitous, it is desirable to reduce the computational load on these devices while still preserving the strong security of CDHP.

Therefore, this study aims at designing password-based authenticated Diffie-Hellman key agreement schemes that reduce the computational load on smart card while still preserve the strong security.

1 New Password-Based Authenticated Diffie-Hellman Key Agreement Scheme

1.1 The Requirements

The requirements of password-based authenticated Diffie-Hellman key agreement schemes could be categorized into three parts: efficiency, resistance to various password-guessing attacks, and general security properties for any secure key agreement schemes.

Efficiency The protocols should reduce the computational load on clients as far as possible.

Resistance to password-guessing attack As a two-party protocol, here we should consider off-line guessing attack and on-line guessing attack.

General security properties These properties for secure key agreement schemes include: ① resistance to replay attack, impersonation attack and man-in-the-middle attack; ② resistance to known key attack, perfect forward secrecy and perfect backward secrecy. Here perfect forward secrecy and perfect backward secrecy mean that, even if we assume that long term secrets of a user are disclosed someday, the previous communications (the session keys before the disclosure) and the following communications (the session keys after the disclosure) are respectively secure.

1.2 The Protocol

Our study focuses on reducing computational load on clients for authenticated Diffie-Hallman key agreement schemes, where the CDHP is introduced as follows.

Definition 1 The CDHP over G is defined as follows: given g, g^x and g^y, where x and y are random numbers and g is a generator for the G, computing g^{xy} is believed to be a hard problem.

Now we introduce a new problem, i.e., the modified Diffie-Hellman problem on which we will build our authenticated Diffie-Hellman key agreement scheme.

Definition 2 The modified CDHP problem over G is defined as follows: given $x+t$, g, g^t and g^y, where t, x and y are random numbers and g is a generator for the G, the problem is to compute g^{xy}.

Theorem 1 The modified CDHP problem is as hard as the CDHP problem.

We will prove the theorem and the security of our scheme in Section 2.

Before introducing our protocol, the notations are introduced as follows: C denotes the client (here a client includes a user and his smart card); S denotes the server; p denotes a large prime, g denotes the primitive generator for $\mathbb{Z}_p$; for simplicity, we will omit all $\bmod p$ notations when the context is clear; x and y represent the ephemeral random numbers chosen by the client and the server, respectively; t denotes client's long term secret which is chosen by the client; $\oplus$ denotes exclusive OR operation, and $\|$ denotes concatenation; $h(\cdot)$ is secure one-way hash function; s is server's long term secret key; K_1 and K_2 are two secret values stored in client's smart card; ID denotes client's identity, and PW denotes his password.

The proposed scheme consists of two phases: initialization and authentication. The scheme is depicted in Fig. 1, where the dashed rectangular denotes the values the entity holds. The details of the scheme are introduced as follows.

Initialization (via a secure channel) The client sends his ID and PW to the server. The server computes $K_1 = h(\mathrm{ID}\|s) \oplus \mathrm{PW}$, writes K_1, p and g into the secure memory of a smart card, and issues the card to the client. Upon receiving the card, the client chooses a random number t, computes $g^t \bmod p$ (we will use g^t for simplicity in the rest of this paper), and writes t and g^t into the secure memory of the client.

Authentication

(1) The user inputs his ID and PW into the card, and then the card extracts $h(\mathrm{ID}\|s)$ by computing $K_1 \oplus \mathrm{PW}$. The card chooses two random values x and r, computes $M_1 = h(h(\mathrm{ID}\|s)\|r) \oplus (x+t)$ and sends (ID, r, M_1, $M_2 = g^t$) to the server.

$$C \rightarrow S : \mathrm{ID}, r, M_1, M_2.$$

(2) Upon receiving the request, the server first computes $h(\mathrm{ID}\|s)$ and then extracts $(x+t)$ by computing $M_1 \oplus h(h(\mathrm{ID}\|s)\|r)$. The server chooses a random value y, computes $g^x = g^{(x+t)}/M_2$, $M_3 = g^y$, $K_{\mathrm{ses}} = (g^x)^y = g^{xy}$ and $M_4 = h(h(\mathrm{ID}\|s)\|K_{\mathrm{ses}})$, and sends ($M_3$, M_4) to the client.

$$S \rightarrow C : M_3, M_4.$$

(3) The client computes $K_{\mathrm{ses}} = (M_3)^x = g^{xy}$ and uses the computed value to verify the validity of the

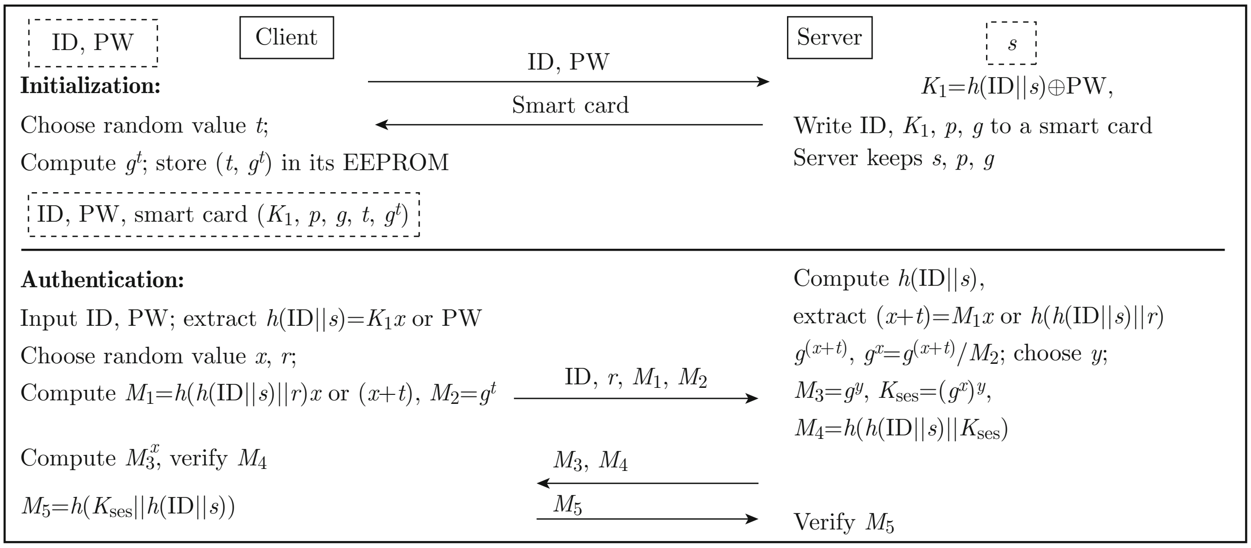
ID, PW
Client
Server
s
Initialization:
ID, PW
K_1=h(ID$||s$)⊕PW,
Smart card
Choose random value t;
Write ID, K_1, p, g to a smart card
Compute g^t; store (t, g^t) in its EEPROM
Server keeps s, p, g
ID, PW, smart card (K_1, p, g, t, g^t)
Authentication:
Compute h(ID$||s$),
Input ID, PW; extract h(ID$||s$)=K_1x or PW
extract (x+t)=M_1x or h(h(ID$||s$)$||r$)
Choose random value x, r;
$g^{(x+t)}$, g^x=$g^{(x+t)}/M_2$; choose y;
ID, r, M_1, M_2
Compute M_1=h(h(ID$||s$)$||r$)x or (x+t), M_2=g^t
M_3=g^y, K_{ses}=$(g^x)^y$,
M_4=h(h(ID$||s$)$||K_{\mathrm{ses}}$)
Compute M_3^x, verify M_4
M_3, M_4
M_5=h($K_{\mathrm{ses}}||h$(ID$||s$))
M_5
Verify M_5

Fig. 1 The proposed password-based Diffie-Hellman key agreement for smart card

received M_4. If the verification is satisfied, then the client computes $M_5 = h(K_{\mathrm{ses}}\|h(\mathrm{ID}\|s))$, sends M_5 to the server, and accepts K_{ses} as the session key. Upon receiving M_5, the server verifies the validity by using its local secrets. If the verification succeeds, then it accepts the client and the session key K_{ses}.

$$C \to S : M_5.$$

2 Performance Evaluation and Security Analysis

2.1 Performance Evaluation

We examine the communication cost first. The scheme requires three message steps (i.e., the optimal number of steps required for mutual authentication between two entities). Regarding computational cost, T_{h} denotes the cost for one hash operation, and T_{E} denotes that for one modular operation. In our scheme, the client requires $3T_{\mathrm{h}} + T_{\mathrm{E}}$ and the server needs $4T_{\mathrm{h}} + 4T_{\mathrm{E}}$ during the authentication phase. The Naïve implementation of Diffie-Hellman key computation without authentication would require at least two modular exponentiations on client. Even though our scheme reduces only one modular exponentiation here, our scheme improves the computation on client significantly because modular exponentiation is one of the most intensive computations and many mobile devices (like smart card, RFID, ZigBee, etc.) are resource-limited.

Regarding the security, our scheme achieves resistance to password guessing attacks. Since the password is used only to activate the smart card, and the computation and communication of the protocol does not involve password. Therefore, password guessing attacks do not work on our scheme.

Regarding the security, our scheme achieves mutual authentication and resistance to replay attack, impersonation attack and known key attack. The security of the scheme depends on the secret key $h(\mathrm{ID}\|s)$. The client uses it to securely transmit $(x + t)$ and uses it in generating validation value M_5, while the server uses it to generate the validation value M_4. The security of the session key K_{ses} is based on the CDHP problem, and the key is enclosed in the computation of M_3 and M_4; this ensures the mutual authentication, resistance to replay attack and impersonation attack. Each session key is a new Diffie-Hellman value which ensures its resistance to known key attack.

Regarding the security, our scheme achieves forward secrecy. Forward secrecy discusses the security of session keys under the assumption that the system long-term secrets are disclosed. Now we discuss it in the following several cases. ① In our scheme, if we assume the server's long-term secret key s is disclosed, then the attacker can derive $(x+t)$ from M_1 and but it still cannot compute the value K_{ses}. ② If we assume the client's long-term secrets (t, PW) are disclosed, then the attacker can derive x from M_1 and compute $K_{\mathrm{ses}} = M_3^x$, i.e., our scheme only partially satisfies forward secrecy.

The performance of our scheme is summarized as follows. The number of communication steps is three; the computational load on client is $3T_{\mathrm{h}} + T_{\mathrm{E}}$; the computational load on server is $4T_{\mathrm{h}} + 4T_{\mathrm{E}}$; the security of the proposed scheme satisfies resistance to replay attack, password-guessing attack and known key attack; the scheme only partially satisfies forward secrecy.

2.2 Security Proof

We prove the security (the privacy of the session key) of our proposed Diffie-Hellman key agreement scheme. We first prove that the modified CDHP problem is as hard as the CDHP problem, and then prove that breaking the security of the proposed scheme is equivalent to the modified CDHP problem.

Theorem 2 The modified CDHP problem is as hard as the CDHP problem.

Proof We prove this by reduction.

(1) The modified CDHP problem is reduced to the CDHP. Given an instance of the modified CDHP problem, i.e., $((x+t)$, g, g^t and $g^y)$, we compute $g^x = g^{(x+t)}/g^t$ and get the instance (g, g^x and g^y). Assume there is one oracle that can answer the CDHP problem. Then we input the instance (g, g^x and g^y) to the oracle, and we get the answer g^{xy}.

(2) The CDHP problem is reduced to the modified CDHP problem. Assume there is one oracle that can answer the modified CDHP problem: given $((x+t)$, g, g^t and $g^y)$, the oracle outputs g^{xy}. Now given an instance of the CDHP problem, i.e., (g, g^x and g^y), we then choose a random value t, and input the instance (t, g, g^x and g^y) to the oracle. The oracle will answer $(g^t/g^x)^y = g^{ty-xy}$. Using the response, we can derive $(g^{ty-xy}g^{-ty})^{-1} = g^{xy}$, and we get the answer for the CDHP problem-(g, g^x and g^y).

Based on the above arguments, we prove the theorem.

Theorem 3 The proposed authenticated Diffie-Hellman key agreement scheme is secure if the modified CDHP problem is hard.

Proof We prove this by contradiction. In the following proof, we can simplify the session key as g^{xy} without losing the soundness of the proof. Given an instance of our scheme, i.e., (ID_A, r, $h(h(\mathrm{ID}_A\|s)\|r) \oplus (x+t)$, g, g^t and g^y), we define two adversaries AD_1 and AD_2, where AD_1 is a basic adversary who tries to output g^{xy} by using only the eavesdropped instance (ID_A, r, $h(h(\mathrm{ID}_A\|s)\|r) \oplus (x+t)$, g, g^t and g^y) while AD_2 is an advanced adversary who has compromised all the secret keys of clients. Here AD_2 has cracked all $h(\mathrm{ID}_A\|s)$ and can derive $((x+t)$, g, g^t and $g^y)$. Apparently, AD_2 is more powerful than AD_1.

Now we assume AD_2 can break our scheme; equivalently, given $((x+t)$, g, g^t and $g^y)$, AD_2 can output g^{xy}. Apparently, we can construct an adversary AD_3 for breaking the modified CDHP problem by using AD_2 as an oracle. This contradicts the modified CDHP assumption. That is, AD_2 cannot break our scheme. Since AD_1 is less powerful than AD_2, AD_1 cannot break our scheme with non-negligible probability.

This proves our theory.

3 Conclusion

This paper has proposed a new Diffie-Hellman key agreement scheme and has proved its security being equivalent to the modified CDHP problem. The proposed scheme can effectively improve the computational load on client by 50% in terms of modular exponentiations. This improvement is significant for those resource-limited devices. One future work is to extend the current version to achieve perfect forward secrecy.

References

[1] Sandirigama M, Shimizu A, Noda M T. Simple and secure password authentication protocol (SAS) [J]. *IEICE Transactions on Communications*, 2000, **E83-B**(6): 1363-1365.

[2] Chien H Y, Jan J K, Tseng Y M. A modified remote login authentication scheme based on geometric approach [J]. *The Journal of Systems and Software*, 2001, **55**: 287-290.

[3] Sun H M, Li L H. An efficient remote user authentication scheme using smart cards [J]. *IEEE Transactions on Consumer Electronics*, 2000, **46**(4): 958-961.

[4] Wu T C. Remote login authentication scheme based on a geometric approach [J]. *Computer Communications*, 1995, **18**(12): 959-963.

[5] Hwang M S. Cryptanalysis of a remote login authentication scheme [J]. *Computer Communications*, 1999, **22**(8): 742-744.

[6] Bellare M, Canetti R, Krawczyk H. A modular approach to the design and analysis of authentication and key exchange protocols [C]//*Proceedings of 30th Annual Symposium on the Theory of Computing.* New York, USA: ACM, 1998: 419-428.

[7] Bellare M, Pointcheval D, Rogaway P. Authenticated key exchange secure against dictionary attacks [J]. *Lecture Notes in Computer Science*, 2000, **1807**: 139-155.

[8] Ding Y, Horster P. Undetectable on-line password guessing attacks [J]. *ACM Operating Systems Review*, 1995, **29**(4): 77-86.

[9] Brusilovsky A, Faynberg I, Zeltsan Z, et al. RFC683-password-authenticated key (PAK) Diffie-Hellman exchange [EB/OL]. (2013-10-30). http://tools.ietf.org/ html/rfc5683.

[10] Boyko V, Mackenzie P, Patel S. Provably secure password authenticated key exchange using Diffie-Hellman [J]. *Lecture Notes in Computer Science*, 2000, **1807**: 156-171.

[11] Kwon T. Authentication and key agreement via memorable password [C]// *Proceedings of the ISOC Network and Distributed System Security Symposium.* [s.l.]: International Association for Cryptologic Research, 2001: 1-13.

[12] Kwon T. Practical authenticated key agreement using passwords [J]. *Lecture Notes in Computer Science*, 2004, **3225**: 1-12.

[13] IEEE. P1363.2 standard specifications for password-based public key cryptographic techniques [EB/OL]. http://grouper.ieee.org/groups/1363/december 2002.