社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－ＣＳＥＣ－38 (22)
2007／7／19

On Anonymous Password-Authenticated Key Exchange

辛 星漢† 古原 和邦† 今井 秀樹†,††

† 〒101-0021 東京都千代田区外神田 1-18-13 産業技術総合研究所情報セキュリティ研究センター
†† 〒112-8551 東京都文京区春日 1-13-27 中央大学理工学部 電気電子通信工学科

E-mail: †seonghan.shin@aist.go.jp

あらまし An anonymous password-authenticated key exchange (PAKE) protocol is designed to provide both user's password-based authentication and anonymity against a semi-honest server. However, the computation and communication costs of the previous construction grow linearly with the number of users. In this paper, we propose two *efficient* anonymous PAKE (called, MEAP and VEAP) protocols which provide unconditional anonymity of the involved user. If the pre-computation is allowed, the overall computation cost of the MEAP protocol is independent of the number of users. We also show how the VEAP protocol works where the overall computation and communication costs are completely independent of the number of users. In the VEAP protocol, user (resp., server) needs only 2 (resp., 3) on-line modular exponentiations. The security of both protocols is based on the CT-CDH (Chosen Target CDH) problem in the random oracle model.

キーワード 鍵交換プロトコル、パスワード、オンライン攻撃、オフライン攻撃、匿名

On Anonymous Password-Authenticated Key Exchange

SeongHan SHIN†, Kazukuni KOBARA†, and Hideki IMAI†,††

† Research Center for Information Security, AIST, 1-18-13 Sotokannda, Chiyoda-ku, Tokyo 101-0021 Japan
†† Chuo University

E-mail: †seonghan.shin@aist.go.jp

Abstract An anonymous password-authenticated key exchange (PAKE) protocol is designed to provide both user's password-based authentication and anonymity against a semi-honest server. However, the computation and communication costs of the previous construction grow linearly with the number of users. In this paper, we propose two *efficient* anonymous PAKE (called, MEAP and VEAP) protocols which provide unconditional anonymity of the involved user. If the pre-computation is allowed, the overall computation cost of the MEAP protocol is independent of the number of users. We also show how the VEAP protocol works where the overall computation and communication costs are completely independent of the number of users. In the VEAP protocol, user (resp., server) needs only 2 (resp., 3) on-line modular exponentiations. The security of both protocols is based on the CT-CDH (Chosen Target CDH) problem in the random oracle model.

Key words authenticated key exchange, passwords, on-line and off-line dictionary attacks, anonymity

1. Introduction

In 1976, Diffie and Hellman published their seminal paper that introduced how to share a secret over public networks [8]. Since then, many researchers have tried to design secure cryptographic protocols for realizing secure channels. These protocols are necessary because application-oriented protocols are frequently developed assuming the existence of such secure channels. In the 2-party setting (e.g., a user and a server), this can be achieved by an authenticated key exchange (AKE) protocol at the end of which the two parties authenticate each other and share a common and temporal session key to be used for subsequent cryptographic algorithms (e.g., AES-CBC or MAC). For authentication, the parties typically share some information in advance. The shared information may be the form of high-entropy cryptographic keys: either a secret key that can be used for symmetric-key encryption or message authentication code, or public keys (while the corresponding private keys are kept secret) which can be used for public-key encryption or digital signatures.

In practice, low-entropy human-memorable passwords such as 4-digit pin-code or alphanumerical passwords are commonly used rather than high-entropy keys because of its convenience in use. Many password-based AKE protocols have been extensively investigated for a long time where a user remembers a short password and the corresponding server holds the password or its verification data that is used to verify the user's knowledge of the password. However, one should be careful about two major attacks on passwords: on-line and off-line dictionary attacks. The on-line dictionary attack is a series of exhaustive searches for a secret performed on-line, so that an adversary can sieve out possible secret candidates one by one communicating with the target party. In contrast, the off-line dictionary attack is performed off-line in parallel where an adversary exhaustively enumerates all possible secret candidates, in an attempt to determine the correct one, by simply guessing a secret and verifying the guessed secret with recorded transcripts of a protocol. While on-line attacks are applicable to all of the password-based protocols equally, they can be prevented by letting a server take appropriate intervals between invalid trials. But, we cannot avoid off-line attacks by such policies, mainly because the attacks can be performed off-line and independently of the party.

1.1 Password-Authenticated Key Exchange and Anonymity

In 1992, Bellovin and Merritt [3] discussed an interesting problem about how to design a secure password-only protocol where a user remembers his/her password only (without any device and any additional assumption) and the counterpart server has password verification data. Their proposed protocols are good examples (though some are turned out insecure) that a combination of symmetric and asymmetric cryptographic techniques can prevent an adversary from verifying a guessed password (i.e., doing off-line dictionary attacks). Later, their AKE protocols have formed the basis for what we call Password-Authenticated Key Exchange (PAKE) protocols. Such protocols have been in standardization of IEEE P1363.2 [9].

In PAKE protocols, a user should send his/her identity clearly in order to authenticate each other and share a master-secret that may be the Diffie-Hellman key or a shared secret to be used for generating authenticators and session keys. Let us suppose an adversary who fully controls the networks. Though the adversary cannot impersonate any party in PAKE protocols with non-negligible probability, it is easy to collect a user's personal information about the communication history itself (e.g., history of access to ftp servers, webmail servers, Internet banking servers or shopping mall servers). These information may reflect the client's life pattern and sometimes can be used for spam mails. For this problem, Viet et al., [11] have proposed an anonymous PAKE protocol and its threshold construction that simply combine a PAKE protocol [1] for generating secure channels with an Oblivious Transfer (OT) protocol [7], [10] for user's anonymity. The anonymity is guaranteed against an active adversary as well as a semi-honest server, who follows the protocol honestly but it is curious about identity of user involved with the protocol.

They also gave an application for a company's public bulletin board to which any employee can upload opinions in a password-authenticated and anonymous way.

1.2 Our Contributions

The motivation comes from the fact that the computation and communication costs of [11] grow linearly with the number of users. In this paper, we propose two *efficient* anonymous PAKE (called, MEAP and VEAP) protocols which provide unconditional anonymity of the involved user. If the pre-computation is allowed, the overall computation cost of the MEAP protocol is independent of the number of users. In the VEAP protocol, the overall computation and communication costs are completely independent of the number of users. Specifically, user (resp., server) needs only 2 (resp., 3) online modular exponentiations. The security of both protocols is based on the CT-CDH (Chosen Target CDH) problem in the random oracle model.

2. Preliminary

2.1 Notation

In this subsection, we explain some notation to be used throughout this paper. Let G_p be a finite, cyclic group of prime order p and g be a generator of G_p. This parameter (G_p, p, g) is public to everyone. In the aftermath, all the subsequent arithmetic operations are performed in modulo p unless otherwise stated.

Let l and κ be the security parameters for hash functions and a symmetric-key encryption scheme, respectively. Let $\{0,1\}^*$ be the set of finite binary strings and $\{0,1\}^l$ be the set of binary strings of length l. Let "||" be the concatenation of bit strings in $\{0,1\}^*$. If D is a set, then $d \stackrel{R}{\leftarrow} D$ indicates the process of selecting d at random and uniformly over D. Let us denote $G_p^* = G_p \backslash \{1\}$ where 1 is the identity element of G_p. The hash function $\mathcal{G}$ is a full-domain hash (FDH) function that maps $\{0,1\}^*$ to the elements of G_p^*. While $\mathcal{F} : \{0,1\}^* \rightarrow \{0,1\}^\kappa$, the other hash functions are denoted $\mathcal{H}_k : \{0,1\}^* \rightarrow \{0,1\}^{l_k}$, for $k = 1, 2$ and 3. Let $U = \{U_1, U_2, \cdots, U_n\}$ and S be the identities of a group of n users and server, respectively, with each ID $\in \{0,1\}^*$.

2.2 The Formal Model

Here, we introduce the security model based on [2], [5] and security notions.

2.2.1 Security Model

We consider U_i (i.e., any user of group U) and S as two parties that participate in the key exchange protocol P. In P, each user U_i $(1 \leqq i \leqq n)$ and server S share a low-entropy secret pw_i drawn from a small dictionary of password D_{Password}, whose cardinality is N. Each of U_i and S may have several instances called oracles involved in distinct, possibly concurrent, executions of P. We denote U_i (resp., S) instances by U_i^μ (resp., S^ν) where $\mu, \nu \in N$, or by I in case of any instance. We suppose that an adversary $\mathcal{A}$ is neither any user U nor server S (however, this can be removed depending on the security notions). Note that the adversary has the entire control of networks during the protocol execution which can be represented by allowing $\mathcal{A}$ to ask several queries to oracles. Let us show the capability of adversary $\mathcal{A}$ each query captures:

- Execute(U_i^μ, S^ν): This query models passive attacks, where the adversary gets access to honest executions of P between the instances U_i^μ and S^ν by eavesdropping.
- Send(I, m): This query models active attacks by having $\mathcal{A}$ send a message to instance I. The adversary $\mathcal{A}$ gets back the response I generates in processing the message m according to the protocol P. A query Send$(U_i^\mu, \texttt{Start})$ initializes the key exchange protocol, and thus the adversary receives the first message.
- Reveal(I): This query handles the misuse of the session key by any instance I. The query is only available to $\mathcal{A}$, if the instance actually holds a session key, and at that case the key is released to $\mathcal{A}$.
- Test(I): This query is used to measure how much the adversary can obtain information on the session key. The Test-query can be asked at most once by the adversary $\mathcal{A}$ and is only available to $\mathcal{A}$ if the instance I is "fresh" in that the session key is not obviously known to the adversary. This query is answered as follows: one flips a private coin $b \in \{0,1\}$ and forwards the corresponding session key SK (Reveal(I) would output) if $b = 1$, or a ran-

dom value with the same size except the session key if $b = 0$.

2.2.2 Security Notions

The adversary $\mathcal{A}$ is provided with random coin tosses, some oracles and then is allowed to invoke any number of queries as described above, in any order. The aim of the adversary is to break the privacy of the session key (a.k.a., semantic security) in the context of executing P. The AKE security is defined by the game $\mathbf{Game}^{\mathsf{ake}}(\mathcal{A}, P)$, in which the ultimate goal of the adversary is to guess the bit b involved in the Test-query by outputting this guess b'. We denote the AKE advantage, by $\mathsf{Adv}_P^{\mathsf{ake}}(\mathcal{A}) = 2\Pr[b = b'] - 1$, as the probability that $\mathcal{A}$ can correctly guess the value of b. The protocol P is said to be (t, ε)-AKE-secure if $\mathcal{A}$'s advantage is smaller than ε for any adversary $\mathcal{A}$ running time t.

The anonymity is defined by the probability distribution of messages in P. As [11], we consider a semi-honest server S who honestly follows the protocol P but it is curious about the involved user's identity. Let $P(U_i, S)$ (resp., $P(U_j, S)$) be the transcript of P between U_i (resp., U_j) and S. We can say that the protocol P is *anonymous* if, for any two users $U_i, U_j \in U$, $\mathsf{Dist}[P(U_i, S)] = \mathsf{Dist}[P(U_j, S)]$ where $\mathsf{Dist}[c]$ denotes c's probability distribution. This notion means that any legitimate user can establish a session key with the server, however, the latter gets no information about the user's identity.

2.3 The Underlying Assumptions

COMPUTATIONAL DIFFIE-HELLMAN (CDH) PROBLEM Let $G_p = \langle g \rangle$ be a finite cyclic group of prime order p with g as a generator. A (t, ε)-CDH_{g,G_p} attacker is a probabilistic machine $\mathcal{B}$ running in time t such that its success probability $\mathsf{Succ}^{\mathsf{cdh}}_{g,G_p}(\mathcal{B})$, given random elements g^x and g^y to output g^{xy}, is greater than ε:

$$\mathsf{Succ}^{\mathsf{cdh}}_{g,G_p}(\mathcal{B}) = \Pr[\mathcal{B}(g^x, g^y) = g^{xy}] \geqq \varepsilon \ . \qquad (1)$$

We denote by $\mathsf{Succ}^{\mathsf{cdh}}_{g,G_p}(t)$ the maximal success probability over every adversaries running within time t. The CDH-Assumption states that $\mathsf{Succ}^{\mathsf{cdh}}_{g,G_p}(t) \leqq \varepsilon$ for any t/ε not too large.

CHOSEN TARGET CDH (CT-CDH) PROBLEM [4] Let $G_p = \langle g \rangle$ be a finite cyclic group of prime order p with g as a generator. Let x be a random element of Z_p^* and let $X \equiv g^x$. A (t, ε)-$\mathsf{CT\text{-}CDH}_{g,G_p}$ attacker is a probabilistic machine $\mathcal{B}$ that has access to the target oracle $\mathcal{T}_{G_p}$, returning random points W_i in G_p^*, and the helper oracle $(\cdot)^x$. Let q_T (resp., q_H) be the number of queries $\mathcal{B}$ made to the target (resp., helper) oracles. The advantage of $\mathcal{B}$ attacking the chosen-target CDH problem $\mathsf{Adv}^{\mathsf{ct-cdh}}_{g,G_p}(\mathcal{B})$ is defined as the probability of $\mathcal{B}$ to output a set V of m pairs $((j_1, K_1), \cdots, (j_m, K_m))$ where, for all $1 \leqq i \leqq m$, $\exists 1 \leqq j_i \leqq q_T$ such that $K_i \equiv W_{j_i}^x$, all K_i are distinct and $q_H < q_T$. The CT-CDH-Assumption states that $\mathsf{Adv}^{\mathsf{ct-cdh}}_{g,G_p}(t) \leqq \varepsilon$ for any t/ε not too large. Note that if $\mathcal{B}$ makes one query to the target oracle then the chosen-target CDH assumption is equivalent to the above CDH assumption.

A SYMMETRIC-KEY ENCRYPTION SCHEME A symmetric-key encryption scheme $\mathcal{SE}$ consists of the following two algorithms $(\mathcal{E}, \mathcal{D})$, with a symmetric-key $\mathcal{K}$ uniformly distributed in $\{0,1\}^\kappa$:

- The symmetric-key encryption algorithm $\mathcal{E}$: Given a message M and a symmetric-key $\mathcal{K}$, $\mathcal{E}$ produces a ciphertext $C = \mathcal{E}_{\mathcal{K}}(M)$.
- The symmetric-key decryption algorithm $\mathcal{D}$: Given a ciphertext C and a symmetric-key $\mathcal{K}$, $\mathcal{D}$ recovers a message $M = \mathcal{D}_{\mathcal{K}}(C)$.

The classical security for $\mathcal{SE}$ is that it is infeasible for an adversary to distinguish the encryptions of two messages M_0 and M_1 (of the same length), even though the adversary has access to the en/decryption oracles. We denote by $\mathsf{Succ}^{\mathsf{se}}(t, q)$ the maximal success probability of a distinguisher running within time t and making at most q queries to the $\mathcal{E}$ and $\mathcal{D}$ oracles.

3. A More Efficient Anonymous PAKE Protocol

In this section, we propose a more efficient anonymous PAKE (for short, MEAP) protocol than the construction of [11]. The MEAP protocol guarantees not only semantic security of session keys against an active adversary but also unconditional anonymity of user against a semi-honest server, who follows the protocol honestly. Here we assume that each user U_i $(1 \leqq i \leqq n)$ of the group $U = \{U_1, U_2, \cdots, U_n\}$

has registered his/her password pw_i to server S. For simplicity, we assign the users consecutive integer i ($1 \leqq i \leqq n$) where U_i can be regarded as the i-th user of U.

3.1 The MEAP Protocol

In the MEAP protocol, any user U_i can share an authenticated session key anonymously with server S.

Step 0 (Pre-computation of S): At first, server S chooses a random number x from Z_p^* and a random master secret MS from $\{0,1\}^l$, and computes its Diffie-Hellman public value $X \equiv g^x$. Then, the server repeats the followings for all clients U_j ($1 \leqq j \leqq n$): 1) it computes $W_j \leftarrow \mathcal{G}(U_j, pw_j)$; 2) it computes $K_j \equiv (W_j)^x$ and derives a symmetric-key $\mathcal{K}_j \leftarrow \mathcal{F}(U_j, X, W_j, K_j)$; and 3) it generates a ciphertext $C_j = \mathcal{E}_{\mathcal{K}_j}(\mathsf{MS})$ for master secret MS. If the MEAP protocol is used in the SSL/TLS, this pre-computation can be done when user U_i and server S exchange "hello" messages each other.

Step 1: The user U_i chooses a random number a from Z_p^* and computes $W_i \leftarrow \mathcal{G}(U_i, pw_i)$. The W_i is blinded with g^a so that the resultant value A is computed in a way of $A \equiv g^a \times W_i$. The user sends A and the group U of users' identities to server S.

Step 2: For the received value A, server S computes A^x with the exponent x. Also, the server generates an authenticator $V_S \leftarrow \mathcal{H}_1(U||S||A||A^x||X||\{C_j\}_{1\leqq j\leqq n}||\mathsf{MS})$. Then, server S sends its identity S, X, A^x, $\{C_j\}_{1\leqq j\leqq n}$ and V_S to user U_i.

Step 3: After receiving the message from server S, user U_i first computes $K_i \equiv A^x/X^a$ and $\mathcal{K}_i \leftarrow \mathcal{F}(U_i, X, W_i, K_i)$. With $\mathcal{K}_i$, the user decrypts the i-th ciphertext C_i as follows: $\mathsf{MS}' = \mathcal{D}_{\mathcal{K}_i}(C_i)$(注 1) If the received V_S is not valid (i.e., $V_S \neq \mathcal{H}_1(U||S||A||A^x||X||\{C_j\}_{1\leqq j\leqq n}||\mathsf{MS}')$), user U_i terminates the protocol. Otherwise, the user generates an authenticator $V_{U_i} \leftarrow \mathcal{H}_2(U||S||A||A^x||X||\{C_j\}_{1\leqq j\leqq n}||\mathsf{MS}')$ and a session key $SK \leftarrow \mathcal{H}_3(U||S||A||A^x||X||\{C_j\}_{1\leqq j\leqq n}||\mathsf{MS}')$. The authenticator V_{U_i} is sent to server S.

(注 1) : If there is no specific order of i, user U_i can test all of the ciphertexts until the correct MS is found.

Step 4: If the received V_{U_i} is not valid (i.e., $V_{U_i} \neq \mathcal{H}_2(U||S||A||A^x||X||\{C_j\}_{1\leqq j\leqq n}||\mathsf{MS})$), server S terminates the protocol. Otherwise, the server generates a session key $SK \leftarrow \mathcal{H}_3(U||S||A||A^x||X||\{C_j\}_{1\leqq j\leqq n}||\mathsf{MS})$.

3.2 Security

Here, we show that the MEAP protocol distributes semantically-secure session keys in the random oracle model [6] and provides unconditional anonymity of user.

[Theorem 1] **(AKE Security)** Under the chosen-target CDH assumption, the MEAP protocol is provably secure in the random oracle model.

We leave the proof in the full version of this paper.

[Theorem 2] **(Anonymity)** The MEAP protocol provides unconditional anonymity of user against a semi-honest server.

[Proof 1] Consider server S who follows the protocol honestly, but it is curious about user's identity involved with the MEAP protocol. It is obvious that server S cannot get any information about the user's identity since the A has a unique discrete logarithm of g and, with the randomly-chosen number a, it is the uniform distribution over G_p^*. This also implies that the server cannot distinguish A_i (of user U_i) from A_j (of $U_{j\neq i}$) because they are completely independent one another. In addition, even if server S receives the authenticator V_{U_i} the A does not reveal any information about the user's identity from the fact that the probability for all users to get MS is equal. Therefore, $\mathsf{Dist}[P(U_i, S)] = \mathsf{Dist}[P(U_j, S)]$ for any two users $U_i, U_j \in U$.

4. A Very Efficient Anonymous PAKE Protocol

Though the MEAP protocol of Section 3.1 reduces significantly the computation cost of server S, its overall communication cost still remains relevant to the number of users. In this section, we propose a very efficient anonymous PAKE (for short, VEAP) protocol where both the computation and communication costs are independent of the number of users. The VEAP protocol also guarantees not only semantic security of session keys against an active adversary but also unconditional anonymity of user against a semi-honest server, who follows the pro-

tocol honestly. Moreover, the VEAP protocol satisfies the following security properties: 1) session key privacy against other legitimate users (holding the secret MS) and 2) perfect forward secrecy.

4.1 The VEAP Protocol

The VEAP protocol is made of the following two phases.

- **[Publication of temporarily-fixed values]:** In this phase, server S publishes temporarily-fixed values X and $\{C_j\}_{1 \leq j \leq n}$ on its public bulletin board. The posted values would be used for all users during a time period t. Note that we do not require any security for the bulletin board.

At first, server S chooses a random number x from Z_p^* and a random master secret MS from $\{0,1\}^l$, and computes its Diffie-Hellman public value $X \equiv g^x$. Then, the server repeats the followings for all clients U_j $(1 \leq j \leq n)$: 1) it computes $W_j \leftarrow \mathcal{G}(U_j, pw_j)$; 2) it computes $K_j \equiv (W_j)^x$ and derives a symmetric-key $\mathcal{K}_j \leftarrow \mathcal{F}(U_j, X, W_j, K_j)$; and 3) it generates a ciphertext $C_j = \mathcal{E}_{\mathcal{K}_j}(\mathsf{MS})$ for master secret MS. The values X and $\{C_j\}_{1 \leq j \leq n}$ are posted on server S's public bulletin board along with other information (e.g., posted time, users U, valid period t and so on).

- **[Protocol execution up to t]:** In this phase, any user U_i and server S actually share an authenticated session key anonymously where server S uses x, X and $\{(\mathcal{K}_j, C_j)\}_{1 \leq j \leq n}$ instead of $\{(U_j, pw_j)\}_{1 \leq j \leq n}$.

Step 1: The user U_i chooses two random numbers (a, b) from Z_p^*, and computes the Diffie-Hellman public value $B \equiv g^b$ and $W_i \leftarrow \mathcal{G}(U_i, pw_i)$. The W_i is blinded with g^a so that the resultant value A is computed in a way of $A \equiv g^a \times W_i$. The user sends A, B and the group U of users' identities to server S.

Step 2: While waiting for the message from user U_i, server S chooses a random number y from Z_p^* and computes its Diffie-Hellman public value $Y \equiv g^y$. For the received values A and B, the server computes A^x, B^x and B^y with the exponents x and y. Also, server S generates an authenticator $V_S \leftarrow \mathcal{H}_1(U||S||\mathrm{TRANS}||B^x||B^y||\mathsf{MS})$ where $\mathrm{TRANS} = A||A^x||X||B||Y||\{C_j\}_{1 \leq j \leq n}$. Then, the server sends its identity S, A^x, Y and V_S to user U_i.

Step 3: After receiving the message from server S, user U_i first computes $K_i \equiv A^x / X^a$ and $\mathcal{K}_i \leftarrow \mathcal{F}(U_i, X, W_i, K_i)$. With $\mathcal{K}_i$, the user decrypts the i-th ciphertext C_i as follows: $\mathsf{MS}' = \mathcal{D}_{\mathcal{K}_i}(C_i)$. If the received V_S is not valid (i.e., $V_S \neq \mathcal{H}_1(U||S||\mathrm{TRANS}||X^b||Y^b||\mathsf{MS}')$), user U_i terminates the protocol. Otherwise, the user generates an authenticator $V_{U_i} \leftarrow \mathcal{H}_2(U||S||\mathrm{TRANS}||X^b||Y^b||\mathsf{MS}')$ and a session key $SK \leftarrow \mathcal{H}_3(U||S||\mathrm{TRANS}||X^b||Y^b||\mathsf{MS}')$. The authenticator V_{U_i} is sent to server S.

Step 4: If the received V_{U_i} is not valid (i.e., $V_{U_i} \neq \mathcal{H}_2(U||S||\mathrm{TRANS}||B^x||B^y||\mathsf{MS})$), server S terminates the protocol. Otherwise, the server generates a session key $SK \leftarrow \mathcal{H}_3(U||S||\mathrm{TRANS}||B^x||B^y||\mathsf{MS})$.

4.2 Security

The semantic security of session keys and unconditional anonymity of user in the VEAP protocol directly follow those in the MEAP protocol (see Theorem 1 and 2). Since server S uses the exponent x and the master secret MS during a time period t in the VEAP protocol, we have to consider session key privacy against other legitimate users (holding the MS) and perfect forward secrecy. Note that these two security properties already hold in the MEAP protocol by choosing x and MS randomly and every time.

[Theorem 3] **(Session Key Privacy)** The VEAP protocol has session key privacy against other legitimate users.

[Proof 2] We leave the proof in the full version of this paper.

[Theorem 4] **(Perfect Forward Secrecy)** The VEAP protocol provides perfect forward secrecy under the Gap Diffie-Hellman assumption.

[Proof 3] We also leave the proof in the full version of this paper.

5. Comparison

This section shows how much the MEAP and VEAP protocols are efficient compared to the original anonymous PAKE protocol (in Section 3.2 of [11]) in terms of computation and communication costs to be required (see Table 1). In general, the

表 1 Comparison of anonymous PAKE protocols as for efficiency where n is the number of clients

Protocols	The number of modular exponentiations		Communication costs
	User U_i	Server S	
APAKE [11]	6 (4)	$4n+2$ $(3n+1)$	$\|U\|+\|S\|+(n+1)\|\mathcal{H}\|+(n+2)\|p\|$
MEAP	2 (1)	$n+2$ (1)	$\|U\|+\|S\|+2\|\mathcal{H}\|+3\|p\|+n\|\mathcal{E}\|$
VEAP	5 (2)	$n+5$ (3)	$\|U\|+\|S\|+2\|\mathcal{H}\|+4\|p\|$

number of modular exponentiations is a major factor to evaluate efficiency of a cryptographic protocol because that is the most power-consuming operation. So we count the number of modular exponentiations as computation costs of user U_i and server S. The figures in the parentheses are the remaining number of modular exponentiations after excluding those that are pre-computable. In terms of communication costs, $|\mathcal{H}|$ and $|\mathcal{E}|$ indicate the bit-length of hash function $\mathcal{H}$ and of symmetric-key encryption $\mathcal{E}$, respectively.

With respect to computation costs in the MEAP protocol, user U_i (resp., server S) is required to compute 2 (resp., $n+2$) modular exponentiations. When pre-computation is allowed, the remaining costs of user U_i (resp., server S) are only 1 (resp., 1) modular exponentiations. In case of the VEAP protocol, user U_i (resp., server S) is required to compute 5 (resp., $n+5$) modular exponentiations. When pre-computation is allowed, the remaining costs of user U_i (resp., server S) are only 2 (resp., 3) modular exponentiations. One can easily see that the MEAP and VEAP protocols significantly reduces the number of modular exponentiations for both user and server from the APAKE protocol. With respect to communication costs, the MEAP protocol requires a bandwidth of $(2|\mathcal{H}|+3|p|+n|\mathcal{E}|)$-bits except the length of identities U and S where the bandwidth for $|\mathcal{H}|$ and the modulus size $|p|$ is independent from the number of users while the APAKE protocol is not. In case of the VEAP protocol, it requires a bandwidth of $(2|\mathcal{H}|+4|p|)$-bits except the length of identities U and S where the bandwidth is completely independent from the number of users. If we consider the minimum security parameters recommended in practice ($|p|=1024$, $|\mathcal{H}|=160$ and $|\mathcal{E}|=128$), the gap of communication costs between our and APAKE protocols becomes larger as the number of users increases.

文　献

[1] M. Abdalla and D. Pointcheval. Simple Password-Based Encrypted Key Exchange Protocols. In *Proc. of CT-RSA 2005*, LNCS 3376, pp. 191-208. Springer-Verlag, 2005.

[2] E. Bresson, O. Chevassut, and D. Pointcheval. New Security Results on Encrypted Key Exchange. In *Proc. of PKC 2004*, LNCS 2947, pp. 145-158. Springer-Verlag, 2004.

[3] S. M. Bellovin and M. Merritt. Encrypted Key Exchange: Password-based Protocols Secure against Dictionary Attacks. In *Proc. of IEEE Symposium on Security and Privacy*, pp. 72-84, 1992.

[4] A. Boldyreva. Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme. In *Proc. of PKC 2003*, LNCS 2567, pp. 31-46. Springer-Verlag, 2003. The full version is availabel at `http://www.cse.ucsd.edu/users/aboldyre/papers/b.html`.

[5] M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated Key Exchange Secure against Dictionary Attacks. In *Proc. of EUROCRYPT 2000*, LNCS 1807, pp. 139-155. Springer-Verlag, 2000.

[6] M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *Proc. of ACM CCS'93*, pp. 62-73. ACM Press, 1993.

[7] C. K. Chu and W. G. Tzeng. Efficient k-Out-of-n Oblivious Transfer Schemes with Adaptive and Non-adaptive Queries. In *Proc. of PKC 2005*, LNCS 3386, pp. 172-183. Springer-Verlag, 2005.

[8] W. Diffie and M. Hellman. New Directions in Cryptography. In *IEEE Transactions on Information Theory*, Vol. IT-22, No. 6, pp. 644-654, 1976.

[9] `http://grouper.ieee.org/groups/1363/passwdPK/submissions.html`

[10] W. G. Tzeng. Efficient 1-Out-n Oblivious Transfer Schemes. In *Proc. of PKC 2002*, LNCS 2274, pp. 159-171. Springer-Verlag, 2002.

[11] D. Q. Viet, A. Yamamura, and H. Tanaka. Anonymous Password-Based Authenticated Key Exchange. In *Proc. of INDOCRYPT 2005*, LNCS 3797, pp. 244-257. Springer-Verlag, 2005.