*2011 2nd International Conference on Networking and Information Technology*
*IPCSIT vol.17 (2011) © (2011) IACSIT Press, Singapore*

# An improved N-Party PAKE Protocol

Liu Xiumei[1], Liu Junjiang[2] and Chang Guiran[1]

[1]Computing Center, Northeastern University, Shenyang, China
[2]Tax Department, Neusoft Corporation, Shenyang, China
liuxm@cc.neu.edu.cn,liujunjiang@neusoft.com,chang@neu.edu.cn

**Abstract.** To improve the operation efficiency of N-Party AKE protocol, to reduce the communication costs, we present an improved N-Party PAKE protocol. In N-Party PAKE protocol, the adjacent clients share a temprory encryption key with the help of server, so it reduces the communication numbers between clients and server. And we also show that the N-Party PAKE protocol can resist various attacks.

**Keywords-**PAKE; N-Party;DPWA

## 1. Introduction

In [1-2], Bresson et al. presented the first formal model of the authenticated group Diffie-Hellman key exchange protocol to enable a treatment of dictionary attacks, which allows group members to agree on a session key. And this model is base on *shared password-authentication*(SPWA, for short), that is all group members use a shared password. However, Byun and Lee[3] pointed out that SPWA is not practical since a password is not a common secret but a secret depending on an individual, and they proposed two provably secure protocols, N-party EKE-U and N-party EKE-M, which are based on *different password-authentication*(DPWA, for short) model, that is group members use different passwords.

N-party EKE-U protocol is designed for unicast network environment. Byun and Lee have proved it is secure under the Diffie-Hellman-like assumptions. However, Tang et al.[4] show that the N-party EKE-U protocol suffer from off-line dictionary attack. Phan et al.[5] show that the N-party EKE-U protocol suffer from in-side attack. And the operation of TF protocol in N-party EKE-U brings the low efficiency. And there have been proposed lots of group PAKE protocols[6-10].

To improve the operation efficiency, we propose an improved EKE-U protocol for group members based on password in this paper, called N-Party PAKE. In the protocol, N-Party are clients $C_1, C_2, \cdots, C_{n-1}$ and the server *S*. Each client $C_i (1 \le i \le n-1)$ shares a different password $pw_i$ with the server.

The remainder of this paper is organized as follows. In Section II, we present N-Party PAKE protocol. Next, in Section III, we show the security analysis and efficiency of N-Party PAKE protocol. Finally, we make a conclusion in Section IV.

## 2. N-Party PAKE Protocol

The communication flows for the N-party PAKE protocol include two stages, called up-flow and down-flow. In the up-flow, each client $C_i (1 \le i \le n-1)$ share a password $pw_i$ with the server *S*, and the adjacent clients $C_i (1 \le i \le n-1)$ and $C_{i+1} (1 \le i \le n-2)$ also share a temporary encryption key with the help of Server *S*. The key will be used to encrypt and decrypt the message from $C_i$ to $C_{i+1}$. And each $C_i (1 \le i \le n-1)$ will raise the received intermediate values to the power of its own secret $x_i$ and forwards the resulting values to the next client $C_{i+1} (1 \le i \le n-2)$. In the flows, there are two functions defined as follows:

$$\phi(\alpha_1, ..., \alpha_{i-1}, \alpha_i, x) = \{\alpha_1^x, ..., \alpha_{i-1}^x, \alpha_i\} \in \overline{G}^i \quad (1)$$

$$\pi(\{\alpha_1,...,\alpha_{i-1},\alpha_i\},m,n)=\{\alpha_m,\alpha_{m+1},...,\alpha_{m+(n-1)}\}\in \bar{G}^n \quad (2)$$

## 2.1. Up-Flow of N-Party PAKE

The up-flow includes up-flow1 and up-flow2. The up- flow1 of N-Party PAKE protocol proceeds as Figure 1.

1) The client $C_1$ chooses a random value $a_1 \in Z_q^*$, computes $g^{a_1}$ and encrypts $X_1 = E_{pw_1}(g^{a_1})$ with its password $pw_1$. Then $C_1$ send $X_1$ to $C_2$. When client $C_2$ receives $X_1$, it also chooses a random value $a_2 \in Z_q^*$, computes and encrypts $X_2 = E_{pw_2}(g^{a_2})$, then sends message $X_1|X_2$ to the next client. Each client $C_i(3 \le i \le n-1)$ receives message $\{X_j\}_{j=1}^{i-1}$, inserts their own message $X_i = E_{pw_i}(g^{a_i})$ and then passes it to the next client $C_{i+1}(3 \le i \le n-2)$. The last client $C_{n-1}$ sends $\{X_j\}_{j=1}^{n-1}$ to server *S*.

2) The server *S* can decrypt every $X_i$ and get each $g^{a_i}$ by using password $pw_i$. The server S chooses random values $b_i \in Z_q^*(1 \le i \le n-1)$, computes $g^{b_i}$, $g^{a_i b_i}$, $sk_i = H(C_i \mid S \mid g^{a_i} \mid g^{b_i} \mid g^{a_i b_i})$, $Mac_i = H(sk_i \mid g^{a_i})$. and encrypts message $Y_i = E_{pw_i}(g^{b_i})$. To help the adjacent clients $C_i(1 \le i \le n-1)$ and $C_{i+1}(1 \le i \le n-2)$ generate a temporary key, *S* also chooses random values $r_i \in Z_q^*$, computes $g^{r_i}$, $g^{pw_i r_i}$, and encrypts $\eta_i = E_{sk_i}(g^{pw_{i+1}r_{i+1}} \mid g^{r_i})(1 \le i < n-2)$ and $\eta_{n-1} = E_{sk_{n-1}}(g^{r_{n-i}})$. Then S sends $\{Y_i \mid \eta_i \mid Mac_i\}_{i=1}^{n-1}$ to $C_1$. The $Mac_i$ is used to verify the temporary key by client $C_i$

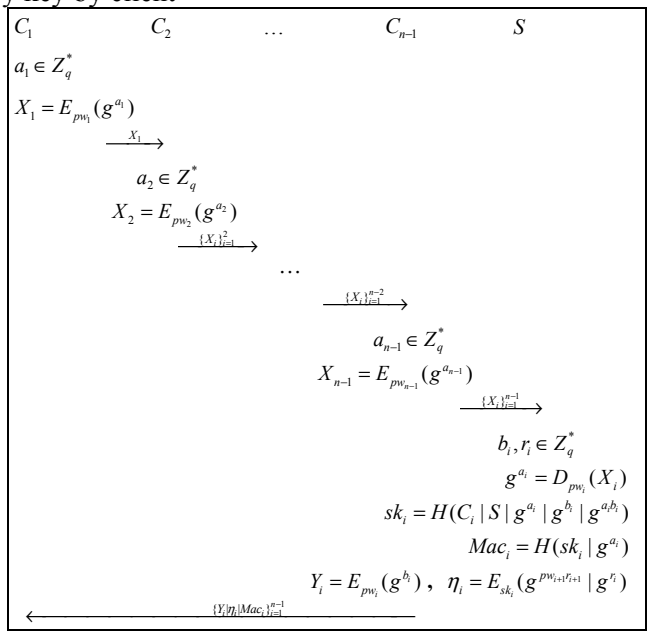
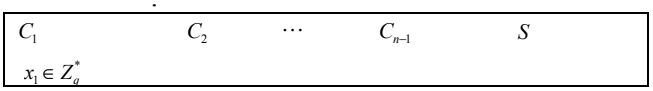



Figure 1. The up-flow1 of the N-Party PAKE protocol

The up- flow2 of N-Party PAKE protocol proceeds as Figure 2.

1) $C_1$ obtains and decrypts $Y_1$ and $\eta_1$, computes and verifies $g^{a_1 b_1}$ and $sk_1$. Then $C_1$ can obtain $g^{pw_2 r_2}$ and $g^{r_1}$, compute $\tau_1 = H(g^{r_1} \mid g^{b_1})$ and $m_0 = \{g^{r_1}\}$. $C_1$ also chooses a random value $x_1 \in Z_q^*$, computes by using $\phi$ function and encrypts $m_1' = E_{g^{pw_2 r_2}}(m_1)$. The $C_1$ sends $m_1' \mid \tau_1 \mid \{Y_i \mid \eta_i \mid Mac_i\}_{i=2}^{n-1}$ to $C_2$.

2) $C_2$ can obtain $Y_2$, $\eta_2$, $sk_2$, $g^{pw_3 r_3}$ and $g^{r_2}$, compute $\tau_2 = H(g^{r_2} \mid g^{r_2})$, $m_2 = \phi(m_1, x_2) = \{g^{r_1 x_2}, g^{r_1 x_1}\}$ and $m_2' = E_{g^{pw_3 r_3}}(m_2)$, then $C_2$ sends $m_2' \mid \{\tau_i\}_{i=1}^{2} \mid \{Y_j \mid \eta_j \mid Mac_j\}_{j=3}^{n-1}$ to the next client. The next client $C_i(3 \le i \le n-1)$ obtains the message $Y_i$ and $\eta_i$, computes $m_i$ and $\tau_i$, then sends $m_i' \mid \{\tau_j\}_{j=1}^{i} \mid \{Y_k \mid \eta_k \mid Mac_k\}_{k=i+1}^{n-1}$ to the next $C_{i+1}(3 \le i \le n-2)$.



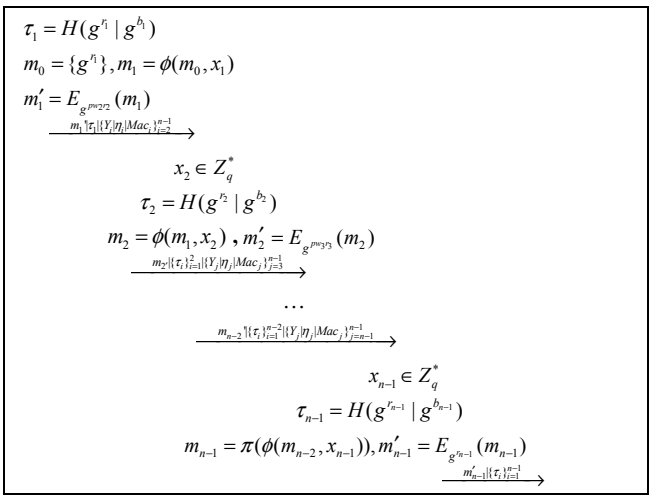



Figure 2. The up-flow2 of the N-Party PAKE protocol

3) The last client $C_{n-1}$ decrypts $m_{n-2}'$, chooses a random value $x_{n-1} \in Z_q^*$, and computes $m_{n-1} = \pi(\phi(m_{n-2}, x_{n-1}))$ by using the function $\phi$ and $\pi$. Then $C_{n-1}$ sends all $\{\tau_i\}_{i=1}^{n-1}$ to server *S*.

## 2.2. Down-Flow of N-Party PAKE

The down-flow of N-Party PAKE protocol proceeds as Figure 3.

1) The server S receives the message and verifiers all $\tau_i$. Then S decrypts $m_{n-1}'$ and obtains each $\tilde{m}_i = g^{r_1 x_1 \cdots x_{i-1} x_{i+1} \cdots x_{n-1}} (1 \le i \le n-1)$. S computes each $\tilde{m}_{n,i} = E_{sk_i}(\tilde{m}_i)(1 \le i \le n-1)$ and sends it to $C_{n-1}$.

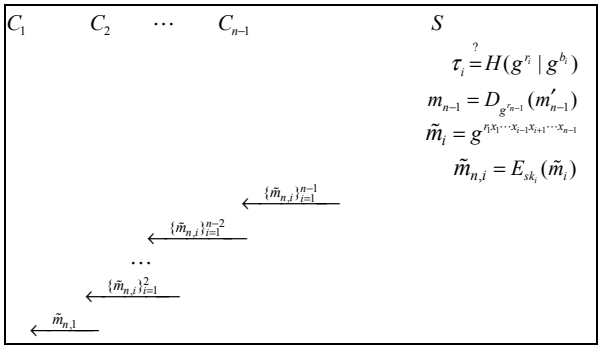



Figure 3. The down-flow of the N-Party PAKE protocol

2) The message $\tilde{m}_{n,i} = E_{sk_i}(\tilde{m}_i)(1 \le i \le n-1)$ send from $C_{n-1}$ to $C_1$. Each client decrypts the message and computes $K = (\tilde{m}_i)^{x_i} (1 \le i \le n-1)$ by using the random value $x_i \in Z_q^*$. And then, each client can compute the $SK = H(K \mid C)$ by $C = \{C_1, \cdots, C_{n-1}\}$.

# 3. Security Analysis and Efficiency

## 3.1. Security Analysis

In this section, we briefly discuss the security against several conventional attacks.

1) Tang-Chen attack

Assume that $C_1$ is a inside attacker, it wants to obtain the password of $C_3$. $C_1$ intercepts the message $m_2'$ from $C_2$ to $C_3$. Because the message is $m_2' = E_{g^{pw_3 r_3}}(\{g^{r_1 x_2}, g^{r_1 x_1}, g^{r_1 x_1 x_2}\})$ and $x_1 \in Z_q^*$ is chose by $C_1$, $C_1$ wants to guess $pw_3$ from $g^{pw_3 r_3}$. But $r_3 \in Z_q^*$ is chose by server *S*, even guessed $g^{pw_3 r_3}$, $C_1$ still can't figure out $pw_3$. Therefore, we could say the proposed protocol is secure against Tang-Chen attack.

2) Undetectable on-line dictionary attack

Assume $n = 4$, the N-Party are $C_1$, $C_2$, $C_3$ and *S*, and an outside attacker is *A*. *A* attempts to guess the password of $C_3$ on-line. First, *A* intercepts messages $\{X_i\}_{i=1}^{2}$ from $C_2$ to $C_3$. *A* chooses a random value

$a \in Z_q^*$ to compute $g^a$, and uses the guessed password $pw_3'$ to encrypt the message $X_3'$, and then pass it to *S*. After receiving the messages, *S* uses the correct password $pw_3$ to decrypt the message, gets $g^{a'}$, and then chooses a random number $b_3 \in Z_q^*$, makes $Y_3 = E_{pw_3}(g^{b_3})$, $sk_3' = H(C_3 \mid S \mid g^{a'} \mid g^{b_3} \mid g^{a'b_3})$, $\eta_3' = E_{sk_3'}(g^{r_3})$, and other calculations are performed correctly according to the protocol. Next, the attacker *A* intercepts the message from $C_2$ to $C_3$ again, using the guessed password $pw_3'$ to decrypt $Y_3$, to get $g^{b_3'}$, and then to calculate $sk_3'$ to decrypt message $\eta_3'$, get $g^{r_3}$. Although *A* can calculate the message $m_3'$ to pass the message to the server *S*, it can't obtain the right $\tau_3$ without the real $g^{b_3}$. So the messages will not pass the authentication from server *S*. And the on-line attack can be detected by *S*. Therefore, the N-Party PAKE protocol can prevent undetectable on-line dictionary attacks.

3) Off-line dictionary attack

The passwords $pw_i$ of clients used only in the up-flow1 to encrypt the message $X_i = E_{pw_i}(g^{a_i})$ from $C_i$ to S, and the returned message $Y_i = E_{pw_i}(g^{b_i})$. In these two messages, whether internal or external attacker wants to launch off-line dictionary attack are impossible. It is because that $a_i, b_i \in Z_q^*$ are chose by the clients $C_i$ and *S* respectively. The attackers can't obtain the random values, so they can't guess the passwords. Therefore, the N-Party PAKE protocol can resist off-line dictionary attacks.

4) Man-in-the-Middle attack

Man-in-the-Middle attack means an adversary deceive the parties in a legal communication.

Similarly, assume $n = 4$, the N-Party are $C_1$, $C_2$, $C_3$ and *S*, and an outside attacker is *A*. *A* attempts to impersonate a party communicate with the other parties. First, *A* intercepts message $X_1$ from $C_1$, then chooses a random value $a \in Z_q^*$ and computes $X_1' = E_{pw_1'}(g^a)$ by using a guessed password $pw_1'$, then sends $X_1' = E_{pw_1'}(g^a)$ to *S*. *S* decrypts $X_1'$ and obtains $g^{a'}$ by using the real $pw_1$, then S chooses a random value $b_1 \in Z_q^*$, computes $g^{b_1}$, $g^{a'b_1}$, $Y_1 = E_{pw_1}(g^{b_1})$, $sk_{A,S} = H(C_1 \mid S \mid g^{a'} \mid g^{b_1} \mid g^{a'b_1})$, $Mac_{A,S} = H(sk_{A,S} \mid g^{a'})$ and $\eta_{A,S} = E_{sk_{A,S}}(g^{pw_2 r_2} \mid g^{r_2})$. A intercepts the message from S, chooses $b \in Z_q^*$, and computes the false message $Y_1' = E_{pw_1'}(g^b)$, $sk_{A,C} = H(C_1 \mid S \mid g^{a_1'} \mid g^b \mid g^{a_1'b})$, $Mac_{A,C} = H(sk_{A,C} \mid g^{a_1'})$, Then A sends the false message to $C_1$. When $C_1$ receives the message, $C_1$ will verify the message by using the real $g^{a_1}$. Obviously, the verifying can not be passed. Therefore, the N-Party PAKE protocol can resist Man-in-the-Middle attack.

5) Forward secrecy

If an attacker obtained all passwords $pw_i$ and the early messages, he can only get $g^{a_i}$ and $g^{b_i}$. But he has no idea about $g^{a_i b_i}$, then he can't compute the session key $sk_i$. If the attacker obtained the session key $sk_i$, he still can't compute the early group session key *SK*. Therefore, the N-Party PAKE protocol is forward secrecy.

## 3.2. Efficiency Analysis

In Table 1, we compare the Byun's N-Party EKE-U protocol with the N-Party PAKE protocol in computational costs and communication costs.

TABLE I. EFFICIENCY COMPARISON

| Protocol / Costs | N-Party PAKE protocol | Byun's N-Party EKE-U protocol |
|---|---|---|
| *Computational costs for clients* | $(n^2+5n-6)/2$ | $(n^2+6n-9)/2$ |
| *Computational costs for server* | $4n-5$ | $(n^2+3n-6)/2$ |
| *Total computational costs* | $(n^2+13n-16)/2$ | $(2n^2+9n-15)/2$ |
| *Communication costs* | $3n-2$ | $4n-6$ |

From the comparison, we could see that the N-Party PAKE protocol has some improvements in computational costs and communication costs.

## 4. Conclusion

To improve the operation efficiency, we propose an N-Party PAKE protocol. The protocol has some improvements in computational costs and communication costs. The improved protocol can resist many familiar attacks.

## 5. References


[1] Bresson, O. Chevassut, and D. Pointcheval, Group diffie-hellman key exchange secure against dictionary attacks, In proceedings of Asiacrypt'02, LNCS Vol. 2501, Springer-Verlag, 2002, pp. 497-514.

[2] E. Bresson, O. Chevassut, D. Pointcheval, and J. J. Quisquater, Provably authenticated group diffie-hellman key exchange, In proceedings of 8th ACM Conference on Computer and Communications Security, 2001, pp. 255-264.

[3] J.W.Byun, D.H.Lee , N-Party Encrypted Diffie-Hellman Key Exchange Using Different Passwords. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, Springer, Heidelberg, 2005, pp. 75–90.

[4] Q. Tang, L. Chen: Weaknesses in two group Diffie-Hellman Key Exchange Protocols, Cryptology ePrint Archive (2005), 2005/197.

[5] R.C.W. Phan, B.M.Goi: Cryptanalysis of the N-party encrypted diffie-hellman key exchange using different passwords, Lecture Notes in Computer Science: Proceedings of the 4th International Conference on Applied Cryptography and Network Security, ACNS(2006). Singapore, pp. 226-238.

[6] E. Bresson, O. Chevassut, and D. Pointcheval. The Group Diffie-Hellman Problems. In H. Heys and K. Nyberg, editors, Proc. Of SAC '2002, LNCS. Springer-Verlag, August 2002.

[7] J.O.KWON, I.R.JEONG, D.H.LEE. Provably-secure two-round password-authenticated group key exchange in the standard model[A]. IWSEC 2006[C]. Kyoto, Japan, 322-336(2006)

[8] R.M.SAYED, M.H.IBRAHIM, Z.B.NOSSAIR. Group key exchange protocol for users with individual passwords[J]. Journal of Engineering and Applied Science, 55(8):327-342(2008)

[9] E.BRESSON, O.CHEVASSUT, D.POINTCHEVAL. Group Diffie-Hellman key exchange secure against dictionary attacks[A]. Asiacrypt'02[C]. Queenstown, New Zealand, 497-514(2002)

[10] Z.WAN, R.H.DENG, F.BAO, et al. nPAKE+: a hierarchical group password-authenticated key exchange protocol using different passwords[A]/ ICICS 2007[C]. Zhengzhou, China, 31-43 (2007)