

本文介绍了 two-server PAKE 协议的原理；给出两个 compilers 在 identity-based cryptography 的基础上，将 two-party PAKE 转换为 two-server PAKE，称为 1D2S PAKE 协议；在不使用 ROM 情况下证明安全性；与先前研究相比，性能得到提升。

- ② 对协议的分析较完整, 包括定义、执行流程、安全性分析及性能评估
- ③ 对PAKE的介绍与分类较好.

② 在 two-server 协议中, 存在以下两个问题: 1> 假设中提到认为两个服务器不会相互勾结, 这一点在实际中是否容易保证及实现; 2> 上述协议, C 与两个服务器均建立了会话密钥, 最终 C 与哪个服务器通信未交代清楚, 是否可以只有一个服务器通信, 另一个负责认证 C

② 协议为客户端带来了较大的开销(相比于前人研究运算时间变长),在实际应用中对客户端的要求较高。

PAKE	口令存储在单个服务器上 (server compromise 有隐患)	<ul style="list-style-type: none"> <li>Password-only PAKE</li> <li>PKI-based PAKE</li> <li>ID-based PAKE</li> </ul>
	口令存储在多个服务器上	<ul style="list-style-type: none"> <li>Threshold PAKE</li> <li>Two-server PAKE</li> </ul>

▲ two-server PAKE: 客户端将口令分成两个份额(share)分别存储到两个服务器中, 两个服务器在未知口令的情况下可以认证客户端, 即便攻击者破坏了一个服务器, 也无法伪装成客户端通过另一个服务器的认证。

▲在 identity-based cryptography 中, 通常使用 Private key Generator (PKG) 生成私钥。本文方案部署了多个 PKG 生成私钥并通过安全信道发送给 server 生成私钥。只要有一个 PKG 是好的, 就可以保证私钥有服务器可知。(Paterson-ACISP'06).

▲ 关键定义 (结合 Katz'05 和 Boneh'01 的模型定义).

假设: ①至少一个PKG正确执行协议 ②两个服务器不会勾结确定口令取值.

ID2S PAKE Protocol (P', 由 two-Party PAKE 协议  $P$  转换而来) based on IBS.

Initialization:  $m$  个 PKG 为协议  $P$  和 IBS 生成系统参数和公共参数, 确定公钥加密算法  $E$ , IBS 中有椭圆曲线上的子群, 阶为  $n$ ,  $E$  上为循环群  $(G, q, g)$ ,  $H: \{0, 1\}^* \rightarrow Z_n^*$ ,  $H: \{0, 1\}^* \rightarrow Z_n^*$ , PKG 会秘密共享  $master^{IBS}$ , 为服务器生成私钥  $ds$ .

$p_{wc}$  为客户端口  $c$ ，存储在服务器上的份额为： $p_{wc,A} + p_{wc,B} = p_{wc} \pmod{q}$   
 且空间  $N < \min(n, q)$

The diagram illustrates a secure communication protocol between two parties, C and B, using a gateway A. The process is as follows:

- Initial State:** C has a message  $m$  and a key  $K_C$ . B has a key  $K_B$ .
- Step 1:** C sends a message  $msg = \langle C, W_C \rangle$  to A. The message is encrypted with  $K_C$  to produce  $W_C$ .
- Step 2:** A forwards the message  $msg = \langle C, W_C \rangle$  to B. The message is decrypted with  $K_B$  to produce  $W_B$ .
- Final State:** B receives the message  $msg = \langle C, W_B \rangle$  and the key  $K_B$ .

$$\begin{aligned} \text{msg}_A &= \langle A, W_a, PK_a, S_a \rangle \\ \text{msg}_B &= \langle B, W_b, PK_b, S_b \rangle \end{aligned}$$

验证签名, 未通过返回 1

$$\begin{aligned} &acc = TRUE, sk_{CA} = W_A^c, sk_{CB} = W_B^c \\ &pw_1 \leftarrow Z_A^c, pw_2 \leftarrow pw_c - pw_1 \pmod{q_2} \\ &h_1 = H_2(C, W_c, A, W_A), h_2 = H_2(C, W_c, B, W_B) \\ &E_a = E(g^{pw_1 h_1}, pk_a), E_b = E(g^{pw_2 h_2}, pk_b) \end{aligned}$$

$msg_1 = \langle C, E_a \rangle$   $\rightarrow$   $msg_2 = \langle C, E_b \rangle$

$$h'_i = H_2(C, W_c, A, W_a),$$

$$W_a = \text{DKEC}, \text{sk}_a) \quad h'_i / g^{\text{puc}_A} = g^{\text{puc}_i - \text{puc}_A}$$

执行协议后,应删除  
在客户端的缓存.

否则返回 1

$$\begin{aligned} Y_b &\leftarrow Z_q^*, (pk_b, sk_b) \leftarrow \mathcal{R}_K \mathcal{G}^E(1^K) \\ W_b &= g^{Y_b}, h_b = H_0(B, w_b, c, w_c, pk_b) \\ S_b &= \text{Sign}(h_b, d_b) \end{aligned}$$
$$h_1^2 = H_2(C, W_C, B, W_B)$$

$$W_B = q^{p_{WC,B}} D(E_b, s_b) h_1^2 = q^{p_{WC,B}} - p_{W_2}$$

Wb)  $\rightarrow$  共享  $W_a = W_b$

若  $acc_B = TRUE$ ,  $acc_C = TRUE$ ,  $sk_{B,C} = W$   
 否则返回 1

### ▲ IBS 安全性证明:

基于假设: ①对 adaptive chosen message 攻击者, IBS 是不可伪造的 ②E 可抵抗选择密文攻击 ③在  $(G, g, q)$  上, DDH 问题是困难的 ④P 是安全的 ⑤提供显式认证的 PAKE ⑥H<sub>1</sub>, H<sub>2</sub> 抗碰撞 假设攻击者可以 corrupt(B), 拿到私钥和 de share.

$P_0$ : 协议的真实执行.

$P_1$  与  $P_0$  的不同当 oracle 生成的消息重复或  $H_1$  中产生的值碰撞时, 实验终止, 敌手失败。  $\text{Adv}_{A, \mathcal{G}}^{\text{PRG}}(n) = \Pr[A^{\text{PRG}}(n) = 1]$

B2: 与B1的不同, 在  $\text{Execute}(C, z, A, j, B, k)$  中, A可能询问  $\text{Corrupt}(B)$ , 将  $E_a$  中的明文替换为  $G$  中的随机元素, 做积有打破正的语言安全性才可区分  $\text{Adv}_A^{\text{B2}}(k) - \text{Adv}_A^{\text{B1}}(k)$  可忽略

$P_3$ : 与  $B$  的不同, 将  $SK_{C,A}$  和  $SK_{A,C}$  替换为  $G$  中相同的随机元素, 归约至解 DDH 问题.

P4: 以上可得 execute 函数中的任务归为三类逻辑, 其中, 1. 仅对  $\text{Send}(C, i, A, B, \text{msg}_A|\text{msg}_B)$  和  $\text{Send}(A, j, C, \text{msg}_i)$  的响应。若  $\text{msg}_A|\text{msg}_B$  是敌手生成的,  $\text{acc} \leftarrow \text{TRUE}$ , 则  $\text{msg}_A|\text{msg}_B$  是有效的, 若发现  $\text{msg}_A|\text{msg}_B$  是敌手生成的且有效, 则  $\text{simulator}$  halt 且  $\text{acc}$  赋值为 1, 有其余情况同 P3。此时认为敌手成功的条件包括  $\text{acc}$  为 1 的情况, 则:  $\text{Adv}_A^B(k) \leq \text{Adv}_A^{\text{P4}}(k)$

P5: 与P4的不同, 在敌手的Send(A, j, C, msg<sub>1</sub>)中, 将Ea中的 $g^{pw \cdot h^1}$ 替换为G中的随机值返回. 根据P2, 易知 $|Adv_A^P(k) - Adv_A^B(k)|$ 可忽略

分析P5: ① msg<sub>1</sub>是与pw<sub>C</sub>绑定的, 因此无法完成离线字典攻击

② 敌手只能在以下事件发生时才能成功:

a): 敌手query Send(C, z, A, B, msg<sub>A</sub> | msg<sub>B</sub>), msg<sub>A</sub> | msg<sub>B</sub>有效,  $acc_C^z = \top$ , Succ<sub>1</sub>.

b): 敌手query Send(A, j, C, msg<sub>1</sub>), msg<sub>1</sub>是敌手生成的且有效,  $acc_A^j = \top$ , Succ<sub>2</sub>.

c) 上述均未发生, 敌手通过Test query 赢得 game.

Succ<sub>1</sub>: 由于msg<sub>A</sub>中包含签名Sa, 敌手无法找到相同的ha(抗碰撞), 也无法伪造签名(IFS不可伪造), 故Pr[Succ<sub>1</sub>]可忽略.

Succ<sub>2</sub>: 假设敌手破坏了B, 则有以下3种情况: 1) 敌手发现H<sub>2</sub>发生碰撞.

2) 敌手更改了Ea中的明文 3) 敌手伪造了msg<sub>1</sub>.

1)和2)由假设可忽略, 3)中敌手可以猜测出, 服务器收到Ea后生成Wa, 在敌手未知Wa的情况下, 敌手能对P做在线猜测.

$$Pr[Succ_2] \leq \frac{Q(k)}{N} + \epsilon(k)$$

$$\text{故 } Pr_A^P[Succ] \leq \frac{Q(k)}{N} + \epsilon(k) + \frac{1}{2} \cdot (1 - Q(k) / (N - \epsilon(k))).$$

$$Adv_A^P(k) \leq Adv_A^B(k) + \epsilon(k) \leq \frac{Q(k)}{N} + \epsilon(k) \quad \square$$

ID2S PAKE Protocol based on IBE

Initialization: 与IBS不同的是, IBE的群G的阶为n, g为生成元

Password Generation: pw<sub>C</sub>仍是均匀随机分布, 服务器A, B分别存储 $(g^{pw_C \cdot A}, g^{pw_C \cdot A})$ 和 $(g^{pw_C \cdot B}, g^{pw_C \cdot B})$ , 其中:  $pw_C \cdot A + pw_C \cdot B = pw_C \pmod{n}$ ,  $pw_C^* \cdot A + pw_C^* \cdot B = pw_C \pmod{q}$

安全性证明:

假设①IBE可以抵抗选择密文攻击. ②E也可抵抗选择密文攻击 ③在(G, q)上, DDH是困难的 ④P是安全的, 提供显式认证的PAKE ⑤H<sub>1</sub>, H<sub>2</sub>抗碰撞.

P<sub>0</sub>: 协议的真实执行.

P<sub>1</sub>: 同IBS的P<sub>1</sub>

P<sub>2</sub>: 更改对Execute(C, z, A, j, B, k)的处理, A可能破坏了B, Ea, Ea'中的明文. 被替换为G中的随机元素. 由于假设①, ②, 易知 $|Adv_A^P(k) - Adv_A^B(k)|$ 可忽略.

P<sub>3</sub>: 更改对Send(A, j, C, msg<sub>1</sub>)和Send(C, z, A, B, msg<sub>A</sub> | msg<sub>B</sub>), 同IBS P<sub>4</sub>, 更改敌手的成功条件,  $Adv_A^P(k) \leq Adv_A^B(k)$ .

C:  $pw_C (= pw_C \cdot A + pw_C \cdot B \pmod{n})$   
 $(= pw_C^* \cdot A + pw_C^* \cdot B \pmod{q})$   
 $pw_1 \in \mathbb{Z}_n^*$ ,  $pw_2 = pw_C - pw_1 \pmod{n}$   
 $yc \in \mathbb{Z}_q^*$ ,  $(pk, sk) \in KGE(1^k)$   
 $Wc = g^{yc}$ ,  $h = H_1(C, C, Wc, pk)$   
 $Ea = IBE(g^{pw_1 \cdot h^1}, A)$   $Eb = IBE(g^{pw_2 \cdot h^2}, B)$

A:  $(g^{pw_C \cdot A}, g^{pw_C^* \cdot A}, da)$   
 B:  $(g^{pw_C \cdot B}, g^{pw_C^* \cdot B}, dB)$

msg<sub>2</sub> =  $\langle C, Wc, pk, Eb \rangle$

msg<sub>1</sub> =  $\langle C, Wc, pk, Ea \rangle$

$h' = H_1(C, Wc, pk)$   
 $Wa = IBD(Ea, da) \cdot g^{pw_C \cdot A} = g^{pw_1 \cdot pw_C \cdot A}$

$h' = H_1(C, Wc, pk)$   
 $Wb = g^{pw_C \cdot B} / IBD(Eb, dB) \cdot h' = g^{pw_2 \cdot pw_C \cdot B}$

$PC(Wa, Wb)$

若  $acc_A^j = \text{TRUE}$ ,  $yc \in \mathbb{Z}_q^*$ ,  $Wa = g^{yc}$   
 $ha = H_2(C, Wa, C, Wc)$   
 $Ea' = E(g^{pw_C^* \cdot A} \cdot ha^1, pk)$   
 $acc_A = \text{TRUE}$ ,  $sk_{A,C} = Wc^{yc}$   
 否则返回⊥

若  $acc_B^z = \text{TRUE}$ ,  $yc \in \mathbb{Z}_q^*$ ,  $Wb = g^{yc}$   
 $hb = H_2(B, Wb, C, Wc)$   
 $Eb' = E(g^{pw_C^* \cdot B} \cdot hb^1, pk)$   
 $acc_B = \text{TRUE}$ ,  $sk_{B,C} = Wc^{yc}$   
 否则返回⊥

msg<sub>A</sub> =  $\langle A, Wa, Ea' \rangle$

msg<sub>B</sub> =  $\langle B, Wb, Eb' \rangle$

$ha = H_2(C, Wa, C, Wc)$ ,  $hb = H_2(B, Wb, C, Wc)$   
 若  $D(Ea', sk) \cdot ha \cdot D(Eb', sk) \cdot hb = g^{pw_C}$ , 有:  
 $acc_C = \text{TRUE}$ ,  $sk_{C,A} = Wa^{yc}$ ,  $sk_{C,B} = Wb^{yc}$   
 否则返回⊥.

P<sub>4</sub>: 与P<sub>3</sub>不同的是, 将Send(A, j, C, msg<sub>1</sub>)和Send(C, z, A, B, msg<sub>A</sub> | msg<sub>B</sub>)的响应更改, 用G中随机元素替换Ea和Ea'中的明文. 由假设①, ②,  $|Adv_A^P(k) - Adv_A^B(k)|$ 可忽略.

对P<sub>4</sub>的分析与IBS P<sub>5</sub>中的Case 2类似, 可得:  $Adv_A^P(k) \leq Adv_A^B(k) + \epsilon(k) \leq Q(k) / (N - \epsilon(k))$ .

与Katz方案比较, 客户端的性能差, 而服务器的计算性能有较大提高