# 基于智能卡的强安全认证与密钥协商协议

李晓伟[1],张玉清[1,2],张格非[2],刘雪峰[1],范 丹[2]
(1.西安电子科技大学 ISN 国家重点实验室,陕西西安 710071;2.中国科学院大学国家计算机网络入侵防范中心,北京 100049)

**摘 要:** 将认证与密钥协商(Authenticated Key Agreement,AKA)协议所需的一种强安全属性——抗临时密钥泄露攻击引入到基于智能卡和口令的 AKA 协议中,基于 NAXOS 方法分别提出了基于智能卡的两方强安全 AKA 协议和三方强安全 AKA 协议.同时,首次给出了包含临时密钥泄露攻击的基于智能卡和口令的 AKA 协议的安全模型,并在该模型下给了所提出协议的安全性证明.此外,文中还分析了抗临时密钥泄露攻击不能在仅使用口令的 AKA 协议中实现的原因.

**关键词:** 认证与密钥协商协议; 临时密钥泄露攻击; 智能卡和口令; 安全模型
**中图分类号:** TP393.08 **文献标识码:** A **文章编号:** 0372-2112 (2014)08-1587-07
**电子学报 URL:** http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2014.08.020

# Strongly Secure Authenticated Key Agreement Protocol Using Smart Card

LI Xiao-wei[1], ZHANG Yu-qing[1,2], ZHANG Gei-fei[2], LIU Xue-feng[1], FAN Dan[2]
(1. *The State Key Lab of ISN, Xidian University, Xi'an, Shaanxi* 710071, *China*;
2. *National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing* 100049, *China*)

**Abstract:** Bring a strong security property, resistance to ephemeral key reveal attack which is needed in the authenticated key agreement (AKA) protocols, to the AKA protocols using smart card and password. A strongly secure two-party AKA protocol using smart card and a strongly secure three-party AKA protocol using smart card were proposed respectively. Meanwhile, the first security model for AKA protocols using smart card and password which includes the ephemeral key reveal attack was proposed. The security proof of the proposed protocols was given in the new security model. The reason why the property of ephemeral key reveal attack can not be achieved in AKA protocols using only password was also given in this paper.

**Key words:** authenticated key agreement protocol; ephemeral key reveal attack; smart card and password; security model

## 1 引言

认证与密钥协商(Authenticated Key Agreement,AKA)协议可以保证两方或者多方用户在不安全的信道上进行安全的通信,它是密码学中最广泛使用的协议之一.如今,许多 AKA 协议已经被应用到电子商务、电子政务以及移动通信之中[1].在现代通信网络中,人们往往追求具有越来越高安全强度的密码协议从而可以防止他们所能预料到的甚至预料不到的攻击.在这样的情况下,抗泄露密码学(Leakage Resilient Cryptography)应运而生[2].抗临时密钥泄露攻击作为抗泄露密码学在认证与密钥协商(AKA)协议中的一个分支,近年来受到广泛关注[3~9].临时密钥泄露攻击模拟的是现实中存在的一种高强度攻击,即允许敌手在 AKA 协议运行时可以获取通信一方或者多方所选取的临时秘密信息,敌手根据这些临时秘密信息来冒充诚实通信者进行通信.临时密钥泄露攻击最初由 LaMacchia 等人[3]于 2007 年引入到 AKA 协议设计之中,由于其模拟的是现实中的一种高强度攻击,因此,自该攻击提出之后越来越多的学者开始在 AKA 协议设计时关注临时密钥泄露攻击.设计可以抵抗临时密钥泄露安全的 AKA 协议也已成为 AKA 协议设计时的一个重要指标.

## 2 相关工作

抗临时密钥泄露攻击的 AKA 协议(也称强安全 AKA 协议)首先出现在基于公钥的 AKA[3~6]协议中.在基于公钥的强安全 AKA 协议设计中,Alwen 等人[4]在 CK 模型下给出了一个强安全 AKA 协议并在随机预言

收稿日期:2013-06-21;修回日期:2013-10-15;责任编辑:蓝红杰
基金项目:国家自然科学基金(No.61272481);北京市自然科学基金(No.4122089)

模型下证明了协议的安全性. Okamoto[5]提出了一个在标准模型可证安全的强安全 AKA 协议及密钥封装协议,并在 eCK 模型[3]下证明了方案的安全性. *Kim* 等人[6]给出了一个不使用 NAXOS 方法[4]的强安全 AKA 协议. 张延红等[7]给出了抗临时密钥泄露攻击的无证书 AKA 协议. 虽然许多强安全 AKA 协议已经在基于公钥的密码学中得以实现,但是我们知道基于公钥的 AKA 协议需要用户记忆一个较长的私钥,有时还需要复杂的密钥管理,因此在某些场景中其应用会受到限制. 基于口令的认证与密钥协商协议(PAKA 协议)可以解决这个问题. 然而,PAKA 协议虽然方便用户使用,但是由于口令长度较短,其容易遭受离线的口令猜测攻击. 因此,如何将强安全的概念引入基于口令的 AKA 协议当中是一个很有意义却又困难的工作. Yoneyama[8]提出了第一个口令场景下的强安全 PAKA 协议. 文献[8]采用的是用户-服务器模型,在认证过程中,用户直接采用服务器的公钥来加密自己的口令然后发给服务器来进行认证. 这样的方法虽然简单,但是一旦服务器的密钥被攻陷则所有用户的口令也同样会被攻陷. Zhao 等人[9]提出了一个三方强安全 PAKA 协议,并在 eCK 模型下证明了其方案的安全性. 然而文献[9]虽然采用的是 eCK 安全模型,但是在其协议证明过程中并没有完全赋予敌手临时密钥揭露的能力,因此该方案提出不久后 Nam 等人[10]指出了该方案存在离线字典攻击.

本文基于智能卡和口令提出了一个抗临时密钥泄露攻击的 PAKA 协议. 针对协议的安全性证明,首次在智能卡和口令场景下提出了包含临时密钥泄露攻击的安全模型,并在该模型下基于 GDH(Gap Diffie-Hellamn)假设证明了协议的安全性. 同时,在所设计的两方强安全 PAKA 协议基础上给出了一个三方强安全 PAKA 协议. 同已有基于口令的强安全 PAKA 协议相比,所提出的协议在安全性和效率方面具有一定优势.

## 3 安全模型

本节我们给出适用于智能卡和口令相结合的强安全 AKA 协议的安全模型. 该模型是在 Bellare 等人的 BPR[11]模型以及 LaMacchia 等人[3]的 eCK 模型下建立的. 据我们所知,这是第一个在智能卡和口令的 AKA 协议场景中考虑临时密钥泄露攻击的安全模型. 模型中有若干诚实用户,服务器 $S$ 和一个可以控制通信信道的敌手 $\mathcal{A}$. 每个诚实用户都拥有一个口令和一个智能卡. 敌手所具有的能力是通过对协议实例的询问来实现的.

### 3.1 敌手能力

*Execute*( $\prod_U^i, \prod_S^j$ ):这个询问模拟的是敌手的被动攻击. 敌手 $\mathcal{A}$ 可以监听到协议实例 $\prod_U^i$ 和 $\prod_S^j$ 的具体通信过程. 这个询问返回的是协议 $P$ 的一个诚实实例的全部通信信息.

*Send*( $\prod_U^i/\prod_S^j, m$ ):这个协议模拟的是敌手的主动攻击. 敌手 $\mathcal{A}$ 可以发送消息 $m$ 给协议实例 $\prod_U^i$ 或者 $\prod_S^j$,消息 $m$ 可能是被敌手篡改或者插入的消息. $\prod_U^i$ 或者 $\prod_S^j$ 按照协议 $P$ 的运行,对消息 $m$ 进行若干操作后,将结果返回给敌手 $\mathcal{A}$.

*Reveal*( $\prod_U^i/\prod_S^j$ ):这个询问模拟的是已知会话密钥攻击,它可以用来确认两次会话所协商出的会话密钥是否是相互独立的. 这个询问所返回的是协议实例 $\prod_U^i/\prod_S^j$ 所得出的会话密钥.

*Long-term key reveal*( $U$, *password*):这个询问模拟的是口令丢失攻击. 敌手通过这个询问可以获得用户 $U$ 的口令信息.

*Smartcard reveal*( $U$, *smartcard*):这个询问模拟的是智能卡丢失攻击. 敌手通过这个询问可以获得存储在智能卡中的信息.

*Ephemeral key reveal*( $\prod_U^i/\prod_S^j$ ):这个询问模拟的是临时密钥泄露攻击. 敌手通过这个询问可以获得和协议实例 $\prod_U^i/\prod_S^j$ 相关的临时秘密信息.

*Test*( $U^i/S^j$ ):这个询问的回答如下:如果测试会话没有计算出会话密钥,则输出 $\perp$,即终止这次询问;否则,抛掷一枚硬币,若硬币正面朝上,设 $b=1$,则返回真实的会话密钥;否则,硬币反面朝上,设 $b=0$,则从密钥空间中随机选择一个随机数返回. 敌手 $\mathcal{A}$ 在获得返回的值后可以继续上述所有询问. 在询问之后,$\mathcal{A}$ 需要回答 *Test*( $U^i/S^j$ )询问所返回的数是真实的会话密钥还是从密钥空间中选取的随机数. 这里需要说明的是 *Test* 询问只能对新鲜的会话而进行的,并且只能进行一次 *Test* 询问.

### 3.2 安全性定义

匹配会话:设一次会话的所有消息的级联为该会话的会话标示. 那么我们说若 $\prod_{i,j}^n$ 和 $\prod_{j,i}^t$ 的会话标示相同,则他们互为匹配会话

新鲜会话:我们称一个会话 $\prod_U^i$ (或者 $\prod_S^j$ )是新鲜的,如果下面的条件成立:

(1)这个会话已经达到了接受状态;

(2) $\prod_U^i$ (或者 $\prod_S^j$ )以及和他匹配的会话没有收到 *Reveal* 询问;

(3)如果对智能卡进行了 *Smartcard reveal* ( $U$,

*smartcard*)询问,则不能向同一个用户进行 *Long-term key reveal*($U$, *password*)询问;

(4)如果对智能卡进行了 *Smartcard reveal*($U$, *smartcard*)询问,则不能再对测试会话进行 *Ephemeral key reveal*($\prod_U^i$)询问.

AKA 安全(语义安全):设敌手正确猜测 *Test* 询问所返回的数是随机数还是真实的会话密钥的概率为 $Adv_{P,D}^{ake}(\mathcal{A})$,则敌手赢得这个游戏的优势可以定义为 $Adv_{P,D}^{ake}(\mathcal{A}) = 2 \cdot Pr[b = b'] - 1$.我们称协议 $P$ 为语义安全的如果 $Adv_{P,D}^{ake}(\mathcal{A})$ 等于 $O(q_{send})/|D_{PW}|$ 加上一个可忽略的量.其中,$q_{send}$ 表示敌手进行 *send* 询问的次数,$|D_{PW}|$ 表示口令空间的大小.

## 4 基于智能卡和口令的强安全认证与密钥协商协议

本节我们使用 NAXOS 方法[4]给出两个基于智能卡和口令的强安全认证与密钥协商协议,我们称之为 SS-PAKA 协议(Strongly Secure PAKA)和 SS-3PAKA 协议.其中 SS-PAKA 协议为两方 PAKA 协议,SS-3PAKA 是在两方 SS-PAKA 协议上扩展的基于服务器的三方 PAKA 协议.

**表 1　协议中所需符号及其定义**

| 符号 | 定义 |
|---|---|
| $S$ | 服务器 |
| $ID_A$, $ID_B$ | 用户 $A$ 和用户 $B$ 的身份标示 |
| $p$, $q$ | 两个大素数,其中 $q \mid p-1$ |
| $G$ | 一个阶为 $q$,生成元为 $g$ 的循环群 |
| $H_1$ | 从 $\{0,1\}^*$ 到 $Z_q^*$ 的 Hash 函数 |
| $h$, $H_2$ | 从 $\{0,1\}^*$ 到 $\{0,1\}^\lambda$ 的 Hash 函数,其中 $\lambda$ 为安全参数 |
| $PW_A$, $PW_A$ | 用户 $A$ 和用户 $B$ 的口令 |
| $x$, $X$ | 服务器 $S$ 的长期私钥及其对应的公钥,其中 $X = g^x$ |
| $sk_{XY}$ | $X$ 和 $Y$ 之间的会话密钥 |

### 4.1 SS-PAKA 协议

图 1 中描述了 SS-PAKA 协议的运行过程.表 1 给出了协议中所用到的一些符号的定义.协议中包含三个阶段,分别是用户注册阶段、用户认证阶段和口令更改阶段.

#### 4.1.1 注册阶段

(1)用户 $A$ 选择一个安全的并且便于记忆的值 $PW_A$ 作为自己的口令.然后选择一个随机数 $b \in \{0,1\}^{1024}$来提高 $PW_A$ 的熵值.$A$ 计算 $h(PW_A, b)$,并将 $h(PW_A, b)$连同自己的身份 $ID_A$ 一起通过一个安全信道发送给服务器 $S$.

(2)$S$ 选择 $u \in \{0,1\}^{64}$并产生一个新的身份 $ID_A = ID_A || u$ 作为用户 $A$ 的新 $ID$.然后,$S$ 利用自己的私钥 $x$ 计算 $W_A = h(x, ID_A) \oplus h(PW_A, b)$,将 $\{ID_A, W_A\}$ 存入智能卡内发送给 $A$.

(3)在收到智能卡后,$A$ 将自己选择的随机数 $b$ 输入到智能卡中.此时,智能卡中的信息为 $\{ID_A, b, W_A\}$.

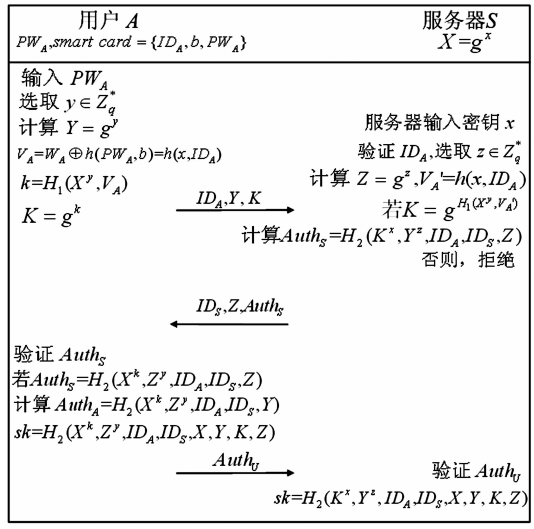


**图1　SS-PAKA协议**

#### 4.1.2 认证阶段

当用户 $A$ 想要登录到服务器 $S$ 获得服务时,他和服务器之间需要进行相互认证.

(1)$A$ 插入智能卡到智能卡终端并输入自己的口令.智能卡提取其存储的信息并计算 $V_A = W_A \oplus h(PW_A, b) = h(x, ID_A)$.然后,选择一个随机数 $y \in Z_q^*$,计算 $Y = g^y$、$k = H_1(X^y, V_A)$以及 $K = g^k$.智能卡将 $\{ID_A, Y, K\}$发送给 $S$.这里的临时密钥指的是用户所选取的秘密值 $y$.

(2)在收到$\{ID_A, Y, K\}$后,$S$ 验证 $ID_A$ 是否正确.若正确,则 $S$ 利用自己的私钥 $x$ 计算 $V_A' = h(x, ID_A)$.然后 $S$ 继续验证$K = g^{H_1(X^y, V_A')}$是否成立.如果等式成立,$S$ 选择随机数 $z \in Z_q^*$ 并计算 $Z = g^z$ 以及 $S$ 的认证信息 $Auth_S = H_2(K^x, Y^z, ID_A, ID_S, Z)$.然后,$S$ 将 $\{ID_S, Z, Auth_S\}$发送给 $A$.

(3)在收到$\{ID_S, Z, Auth_S\}$后,$A$ 利用自己的秘密信息验证 $Auth_S = H_2(X^k, Z^y, ID_A, ID_S, Z)$是否成立.如果不成立,则拒绝.否则,智能卡计算出 $A$ 与 $S$ 之间的会话密钥 $sk = H_2(X^k, Z^y, ID_A, ID_S, X, Y, K, Z)$以及一个确认信息 $Auth_A = H_2(X^k, Z^y, ID_A, ID_S, Y)$.然后将 $Auth_A$ 发送给 $S$.

(4)在收到 $Auth_A$ 后,$S$ 验证 $Auth_A = H_2(K^x, Y^z, ID_A, ID_S, Y)$是否成立.如果不成立,则拒绝.否则,$S$ 计算出

$A$ 与 $S$ 之间的会话密钥 $sk = H_2(K^x, Y^z, ID_A, ID_S, X, Y, K, Z)$.

#### 4.1.3 口令更改阶段

当用户 $A$ 想要更改自己的口令时,他首先插入智能卡到智能卡接收装置,然后输入自己旧的口令 $PW_{old}$ 以及一个新的口令 $PW_{new}$. $A$ 的智能卡计算 $W_{new} = W_{old} \oplus h(PW_{old}, b) \oplus h(PW_{new}, b)$ 然后更新 $W_{old}$ 为 $W_{new}$.

### 4.2 SS-3PAKA 协议

在 SS-PAKA 协议的基础上我们给出一个基于服务器的三方 PAKA 协议我们称之为强安全三方 SS-3PAKA 协议.图 2 中描述了 SS-3PAKA 协议的运行过程.

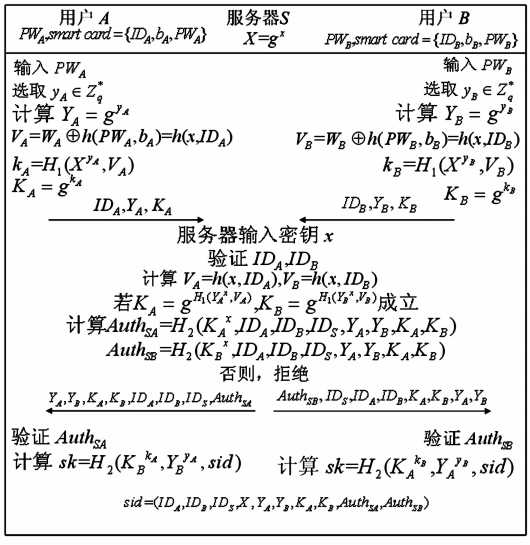


图2 SS-3PAKA协议

#### 4.2.1 注册阶段

这个过程和上面提到的两方 SS-PAKA 协议的注册过程是相同的.我们假设两个用户 $A$ 和 $B$ 分别在服务器 $S$ 处进行了注册.于是,$A$ 注册后得到了口令 $PW_A$ 和智能卡 $\{ID_A, b_A, W_A\}$, $B$ 注册后得到了口令 $PW_B$ 和智能卡 $\{ID_B, b_B, W_B\}$.

#### 4.2.2 认证阶段

这个阶段,用户 $A$ 和 $B$ 可以在服务器 $S$ 的帮助下完成一次认证和密钥协商.

(1) $A$ 插入他的智能卡到智能卡接收终端并输入自己的口令 $PW_A$.然后智能卡选择一个随机数 $y_A \in Z_q^*$,并计算 $Y_A = g^{y_A}$、$k_A = H_1(X^{y_A}, V_A)$ 以及 $K_A = g^{k_A}$.智能卡将 $\{ID_A, ID_B, Y_A, K_A\}$ 发送给 $S$.同时在另一端,$B$ 进行类似的操作. $B$ 插入他的智能卡到智能卡接收终端并输入自己的口令 $PW_B$.然后智能卡选择一个随机数 $y_B \in Z_q^*$,并计算 $Y_B = g^{y_B}$、$k_B = H_1(X^{y_B}, V_B)$ 以及 $K_B = g^{k_B}$.智能卡将 $\{ID_B, ID_A, Y_B, K_B\}$ 发送给 $S$.

(2)在收到 $\{ID_A, ID_B, Y_A, K_A\}$ 和 $\{ID_B, ID_A, Y_B, K_B\}$ 后,$S$ 首先验证用户的身份是否正确.若正确,则 $S$ 利用自己的私钥 $x$ 计算 $V_A = h(x, ID_A)$ 以及 $V_B = h(x, ID_B)$.然后 $S$ 继续验证 $K_A = g^{H_1(Y_A^x, V_A)}$ 以及 $K_B = g^{H_1(Y_B^x, V_B)}$ 是否成立.若成立,则计算认证消息 $Auth_{SA} = H_2(K_A^x, ID_A, ID_B, ID_S, Y_A, Y_B, K_A, K_B)$ 和 $Auth_{SB} = H_2(K_B^x, ID_A, ID_B, ID_S, Y_A, Y_B, K_A, K_B)$. $S$ 发送 $Y_A, Y_B, K_A, K_B, ID_A, ID_B, ID_S, Auth_{SA}$ 给 $A$,发送 $Y_B, Y_A, K_B, K_A, ID_B, ID_A, ID_S, Auth_{SB}$ 给 $B$.

(3) $A$ 在收到 $Y_A, Y_B, K_A, K_B, ID_A, ID_B, ID_S, Auth_{SA}$ 后,验证 $Auth_{SA} = H_2(X^{k_A}, ID_A, ID_B, ID_S, Y_A, Y_B, K_A, K_B)$ 是否正确.若正确,则 $A$ 可以成功的计算出 $A$ 和 $B$ 之间的会话密钥 $sk = H_2(K_B^{k_A}, Y_B^{y_A}, sid)$,其中 $sid = (ID_A, ID_B, ID_S, X, Y_A, Y_B, K_A, K_B, Auth_{SA}, Auth_{SB})$.另一端,$B$ 进行类似的操作.验证 $Auth_{SB} = H_2(X^{k_B}, ID_A, ID_B, ID_S, Y_A, Y_B, K_A, K_B)$ 是否正确.若正确则 $B$ 可以成功的计算出 $A$ 和 $B$ 之间的会话密钥 $sk = H_2(K_A^{k_B}, Y_A^{y_B}, sid)$.

## 5 安全性和性能分析

### 5.1 安全性分析

由于 SS-3PAKA 协议是在 SS-2PAKA 协议基础上的扩展,所以这里仅给出两方协议的安全性证明,三方协议的证明基本类似这里不再详述.在给出具体的证明之前,我们首先给出证明过程中所需要用到的一些安全性假设.

**离散对数假设**(Discrete Logarithm Assumption) 设 $G$ 是一个阶为大素数 $q$ 的循环群, $g$ 是它的生成元, $g^a$ 是群 $G$ 中的元素, $\mathscr{A}$ 是一个要破解 DL 问题的概率多项式时间的敌手.设敌手可以根据 $g^a$ 而计算出 $a$ 的概率为 $Adv_G^{DL}(\mathscr{A})$.则若 $Adv_G^{DL}(\mathscr{A})$ 是一个可忽略的量,则 DL 假设成立.

**间隙 Diffie-Hellman 假设**(Gap Diffie-Hellman (GDH) Assumption) 设 $g^a, g^b$ 是群 $G$ 中的两个元素, $\mathscr{A}$ 是一个要破解 GDH 问题的概率多项式时间的敌手.敌手 $\mathscr{A}$ 可以对一个 DDH(Decisional Diffie-Hellman)问题预言机进行询问,该预言机可以判定其输入是否为一个 Diffie-Hellman 三元组.即每当 $\mathscr{A}$ 向预言机输入一个三元组 $g^a, g^b, Z$ 时,该预言机可以判定该三元组是否满足 $Z = g^{ab}$,然后将判定结果返回给 $\mathscr{A}$.设敌手在拥有一个 DDH 预言机的条件下,可以根据 $g^a, g^b$ 而计算出 $g^{ab}$ 的概率为 $Adv_G^{GDH}(\mathscr{A})$.则若 $Adv_G^{GDH}(\mathscr{A})$ 是一个可忽略的量,则 GDH 假设成立.

**定理 1** 设 $G$ 是一个乘法循环群, $D_{PW}$ 是口令的分布空间, $|D_{PW}|$ 表示这个空间的大小. SS-PAKA 是我们所提出的两方协议.对于任何运行在概率多项式时间 $t$

内的敌手 $\mathcal{A}$ 来说,若他进行少于 $q_s$ 次会话、少于 $q_{send}$ 次 *Send* 询问、少于 $q_{exe}$ 次 *Execute* 询问以及少于 $q_{H_i}(1\leqslant i\leqslant 2)$ 次 Hash 询问,则敌手攻破 SS-PAKA 协议的语义安全性的概率为:

$$Adv^{aka}_{SS-PAKA,G}(\mathcal{A})\leqslant\frac{2q_{send}}{|D_{PW}|}+2q_s^2q_{H_1}Adv^{DL}_G(t)+$$

$$2q_s^2q_{H_2}Adv^{GDH}_G(t)+\frac{(q_{send}+q_{exe})^2+q_{H_2}^2}{2^\lambda}+\frac{q_{H_1}^2}{q}\quad(1)$$

**证明**:设 $\mathcal{A}$ 是一个要破坏 SS-PAKA 协议的语义安全性的多项式时间的敌手,$S$ 是一个模拟器用来为 $\mathcal{A}$ 模拟 SS-PAKA 协议.协议的安全性证明是通过 $S$ 和 $\mathcal{A}$ 之间的一系列游戏来完成的.这里我们定义从 $G_0$ 到 $G_4$ 的 5 个游戏.在系统初始化时,所有用户都需要在 $S$ 处注册.$S$ 自己保留一个列表存储所有用户的注册信息 $(ID_U,V_U)$.我们定义 $Succ_i$ 表示敌手成功区分 $G_i$ 中的 *Test* 询问所返回的数是随机数还是会话密钥这一事件,用 $\Pr[succ_i]$ 表示发生这一事件的概率.下面我们开始具体的证明过程.

游戏 $G_0$ 该游戏是真实的游戏.由语义安全的定义可知:

$$Adv^{ake}_{P,D}(\mathcal{A})=2\cdot\Pr[succ_0]-1\quad(2)$$

游戏 $G_1$ $S$ 在随机预言模型下模拟 SS-PAKA 协议.$S$ 模拟两个 *Hash* 函数 $H_1,H_2$ 作为随机预言机,并维护两个列表 $List_{H_1}$,和 $List_{H_2}$ 用来保存对预言机的询问及其回答.我们可以看出在随机预言模型下 $G_1$ 和 $G_0$ 是不可区分的.

$$\Pr[succ_1]-\Pr[succ_0]=0\quad(3)$$

游戏 $G_2$ 在这个游戏中 $S$ 依然像在游戏 $G_1$ 中一样模拟 SS-PAKA 协议,只是排除 Hash 函数上的碰撞以及协议实例之间的碰撞,即 $\{ID_A,Y,K\}$ 和 $\{ID_S,\mathcal{A},Auth_S\}$ 的碰撞.根据生日悖论,我们可以得出:

$$\Pr[succ_2]-\Pr[succ_1]\leqslant\frac{(q_{send}+q_{exe})^2}{2^{\lambda+1}}+\frac{q_{H_1}^2}{2q}+\frac{q_{H_2}^2}{2^{\lambda+1}}\quad(4)$$

游戏 $G_3$ 在这个游戏中,当敌手 $\mathcal{A}$ 向 $H_1$ 进行 $(X^y,V_A)$ 询问时,$S$ 停止协议的模拟,其中 $(X^y,V_A)$ 为测试会话中的消息.我们定义敌手 $\mathcal{A}$ 在测试会话中对 $(X^y,V_A)$ 进行 $H_1$ 询问的事件为 $Ask_{H_1}$.那么下面我们证明,如果 $Ask_{H_1}$ 发生则我们可以利用 $\mathcal{A}$ 来解决离散对数(DL)问题.$S$ 从 $[1,\cdots,q_s]$ 个会话中选择一个会话作为测试会话,并将一个 DL 问题 DL($g^a$)嵌入 SS-PAKA 协议中来代替服务器的公钥.$S$ 像 $G_2$ 中一样回答敌手 $\mathcal{A}$ 的所有询问,只是当敌手 $\mathcal{A}$ 对 $(X^{y'},V_A)$ 进行 $H_1$ 询问时有所不同,其中 $y'\in Z_q^*$ 为某个非测试会话中敌手选取或者诚实用户所选择的随机数.此时,为了正确模拟非测试会话,当收到对 $(X^{y'},V_A)$ 的 $H_1$ 询问时,$S$ 首先检查列表 $List_{H_1}$ 来确认是否有这样的记录.若有,$S$ 将相应的记录返回.否则,$S$ 将通过一个 DDH 预言机来验证 DDH($X$,$Y'$) = $X^{y'}$ 是否成立.如果不成立,则拒绝回答.若成立,$S$ 选择一个随机数 $H_1^i$ 来回答该 Hash 询问.同时,$S$ 记录下 $(X,Y',X^{y'},V_A,H_1^i)$.这里 $S$ 不能直接回答敌手对 $H_1$ 的询问,因为 $S$ 不知道服务器公钥 $g^a$ 中的 $a$,因此它不能计算出 $X^{y'}$.为了正确模拟非测试会话,在敌手向 $H_1$ 进行询问时,$S$ 需要对所询问的消息进行一个 DDH 询问.现在对于敌手来说非测试会话已经被正确模拟了,下面我们来看测试会话.在测试会话中如果 $Ask_{H_1}$ 事件发生,则说明 $\mathcal{A}$ 计算出了 $(X^y,V_A)$,并对 $(X^y,V_A)$ 进行 $H_1$ 询问.$\mathcal{A}$ 可以通过临时密钥揭露询问 Ephemeral key reveal($\prod_U^i$)计算出 $X^y$,但是对于 $V_A$,则需要 $\mathcal{A}$ 自己计算.因为 $V_A=h(PW_A,a)$,这样如果敌手 $\mathcal{A}$ 对测试会话中的信息进行了 Hash 询问,那么就意味着 $\mathcal{A}$ 计算出了 $a$,这样 $S$ 就可以利用 $\mathcal{A}$ 而破解 DL 问题.设敌手 $\mathcal{A}$ 刚好选择 $S$ 所选择的会话作为测试会话以及相应的会话作为测试会话的匹配会话的概率为 $1/q_s^2$,则我们可以得出游戏 $G_3$ 和 $G_2$ 是不可区分的,除非 $Ask_{H_1}$ 发生,即敌手破解了 DL 问题.这样我们可以得到以下不等式:

$$\Pr[succ_3]-\Pr[succ_2]\leqslant q_s^2q_{H_1}Adv^{DL}_G(t)\quad(5)$$

游戏 $G_4$ 在游戏 $G_3$ 中我们可以看出,敌手 $\mathcal{A}$ 不能在测试会话对 $(X^y,V_A)$ 进行 $H_1$ 询问除非他破解 DL 问题.这样 Hash 函数 $H_1$ 的输出将是完全随机的并且敌手不能获得关于 $k=H_1(X^y,V_A)$ 的任何信息.因此,在游戏 $G_4$ 中,$S$ 就可以将所有协议实例中的 $K=g^k$ 随机化,即用一个随机数替代原有的 $K$.敌手不能区分出他是在进行游戏 $G_3$ 还是游戏 $G_4$,除非敌手在测试会话中对 $(X^k,Z^y,\cdots)$ 进行 $H_2$ 询问,也就是说敌手已经计算出了正确的会话密钥,从而可以区分测试会话所返回的是随机数还是真实的会话密钥.我们设敌手在测试会话中对 $(X^k,Z^y,\cdots)$ 进行 $H_2$ 询问的事件为 $Ask_{H_2}$,若 $Ask_{H_2}$ 发生的话,则 $S$ 可以利用 $\mathcal{A}$ 破解 GDH 问题.为了证明这一论断,$S$ 嵌入一个 GDH 问题 GDH($g^a$,$g^b$)到测试中来代替服务器公钥和测试会话中的消息 $K$.由于在 $G_3$ 中我们已经得出所有 $H_1$ 的输出都是随机的,因此敌手不可能获得关于认证消息 $K$ 的任何信息,即任何由敌手伪造的消息 $K$ 都将被发现.那么在这种情况下,$Ask_{H_2}$ 只能发生在下面两种情形中:

情形 1　$\mathcal{A}$ 只是窃听诚实用户之间的通信,并不伪造任何消息,但是他可以进行除 *Send* 询问外的任何询问.

情形 2　$\mathcal{A}$ 窃取用户 $A$ 的智能卡信息,并且猜测用

户 $A$ 的口令,从而伪装 $A$ 来运行协议.然后 $\mathcal{A}$ 选择一次成功的会话作为他所选择的测试会话.

情形 1,$S$ 从[1,…,$q_s$]个会话中选择一个会话作为测试会话.如上所述,$S$ 嵌入一个 GDH 问题 GDH($g^a$, $g^b$)到测试中来代替服务器公钥和测试会话中的消息 $K$.那么,如果 $\mathcal{A}$ 选择了 $S$ 选择的会话作为测试会话并且正确区分测试会话所返回的是随机数还是真实的会话密钥,也就是事件 $Ask_{H_2}$ 发生,那么 $S$ 可以利用 $\mathcal{A}$ 来计算出 $g^{ab}$.因为如果敌手 $\mathcal{A}$ 赢得了 AKA 安全游戏,就意味着敌手在测试会话中对($g^{ab}$, $Z^y$,…)进行了 $H_2$ 询问.这样 $S$ 就可以利用 $\mathcal{A}$ 的问而获得 $g^{ab}$,从而解决 GDH 问题.注意为了使模拟过程是正确的,$S$ 在模拟非测试会话时如果敌手对($m$, $n$, $ID_A$, $ID_S$, $g^a$, $g^c$, $g^{h(g^{ac},V_A)}$, $g^d$)进行 $H_2$ 询问,$S$ 不能直接回答这个询问,因为 $S$ 不知道 $g^a$ 中的 $a$.因此,$S$ 在收这样的询问后,首先检查列表 $List_{H_2}$ 中是否有相同的记录,若有则将结果返回.若没有该记录,$S$ 则根据 DDH 预言机验证 $m$ = CDH($g^a$, $g^{h(g^{ac},V_A)}$)是否成立.若不成立,则拒绝回答.反之,若 $m$ = CDH($g^a$, $g^{h(g^{ac},V_A)}$),说明敌手进行了正确的 Hash 询问,$S$ 选择一个随机数 $H_2^i$ 作为该询问的结果返回,同时将该 $Hash$ 询问记录到列表 $List_{H_2}$ 中.现在我们可以得出非测试会话的模拟过程是正确的,因此在情形 1 中,如果 $Ask_{H_2}$ 发生,则我们可以解决 GDH 问题.

$$\Pr[Ask_{H_2}] \leq {q_s}^2 q_h Adv_G^{GDH}(t) \tag{6}$$

情形 2,$\mathcal{A}$ 窃取用户 $A$ 的智能卡信息,得到$\{ID_A, b, W_A\}$,然后猜测用户 $A$ 的真实口令.设 $A$ 的真实口令为 $PW_A$,敌手的猜测口令为 $PW_A'$.敌手 $\mathcal{A}$ 利用智能卡内的信息和猜测的口令来伪装 $A$ 和服务器通信.$\mathcal{A}$ 选择一个随机数 $y \in Z_q^*$,计算 $V_A' = W_A \oplus h(PW_A', b)$.然后,选择一个随机数 $y \in Z_q^*$,计算 $Y = g^y$、$k' = H_1(X^y, V_A')$以及 $K' = g^{k'}$.然后 $\mathcal{A}$ 发送$\{ID_A, Y, K'\}$给服务器.如果该消息通过了服务器的验证,那么 $\mathcal{A}$ 的口令猜测是正确的.这样 $\mathcal{A}$ 就可以选择本次会话作为测试会话.此时,我们可以看到事件 $Ask_{H_2}$ 会发生.设敌手成功猜测用户 $\mathcal{A}$ 的口令的概率为 $q_{send}/|PW_A|$,则我们有:

$$\Pr[Ask_{H_2}] \leq \frac{q_{send}}{|PW_A|} \tag{7}$$

由上面的分析可知如果 $Ask_{H_2}$ 不发生,则敌手 $\mathcal{A}$ 不能区分他是在进行游戏 $G_4$ 还是 $G_3$.因此,我们有:

$$\Pr[succ_3] - \Pr[succ_2] \leq \Pr[Ask_{H_2}]$$

$$\leq \frac{q_{send}}{|PW_A|} + {q_s}^2 q_h Adv_G^{GDH}(t) \tag{8}$$

在游戏 $G_4$ 中我们已知 $Ask_{H_2}$ 不会发生,因此 $G_4$ 中所返回的消息 $K$, $Auth_S$, $Auth_U$, $sk$ 均是随机的.对于一个所有消息都是随机的协议来说,敌手区分测试会话所返回的数是随机数还是真实的会话密钥的概率是:

$$\Pr[succ_4] = \frac{1}{2} \tag{9}$$

综合上面所有的不等式,我们可以得到定理 1.

## 5.2 性能分析

我们所提出的基于智能卡的两方强安全 PAKA 协议是首次考虑将临时会话泄露攻击引入基于智能卡和口令的认证协议中,我们将其同最新的基于智能卡的 PAKA 协议[12]进行对比.对于计算消耗,我们只考虑指数运算因为指数运算的时间远远高于 Hash 运算和对称加解密所需时间.如表 2 所示,我们的方案在计算消耗上仅仅略高于文献[12]中的协议,然而文献[12]中的方案并没有考虑临时密钥泄露攻击,如果将其考虑在内其计算消耗将有所增加.表 3 是我们的三方强安全 PAKA 协议同已有的三方基于口令的强安全协议[8,9]在性能上的对比.从表 3 可以看出,文献[8]的计算消耗是三个协议中最低的,但是文献[8]所使用的方法是用服务器公钥直接加密用户的口令.这样假如服务器所选取的临时密钥被泄露的话,那么就意味着用户口令将要被泄露(如果使用抗临时密钥泄露攻击的公钥加密则系统复杂性大大增加).同文献[9]中的方案对比,SS-3PAKA 协议所需的计算消耗低于文献[9]的方案,尤其是用户端的计算消耗.在安全性上文献[9]中的方案容易遭受离线字典攻击[10],而我们的方案是可证明安全的.因此总体来说,我们所提出的方案虽然采用了智能卡和口令相结合的认证方式,但是同仅仅使用口令的认证方式相比安全性大大提升.因此,综合安全性和效率,我们的方案在总体性能上具有一定优势.

**表 2 同最新的两方基于智能卡的 PAKA 协议对比**

| 协议 | 是否考虑临时密钥泄露攻击 | 安全性 | 计算消耗 | |
|---|---|---|---|---|
| | | | 用户 | 服务器 |
| 文献[12] | 否 | 启发式证明 | 4exp | 3exp |
| SS-PAKA | 是 | 可证明安全 | 5exp | 5exp |

**表 3 同已有的三方强安全 PAKA 协议对比**

| 协议 | 是否考虑临时密钥泄露攻击 | 安全性 | 计算消耗 | | 是否使用智能卡 |
|---|---|---|---|---|---|
| | | | 用户 | 服务器 | |
| 文献[8] | 是 | 字典攻击 | 2exp | 2exp | 否 |
| 文献[9] | 是 | 字典攻击 | 9exp | 4exp | 否 |
| SS-3PAKA | 是 | 可证明安全 | 6exp | 6exp | 是 |

## 5.3 讨论

抗临时密钥泄露攻击是 AKA 协议中一个非常强的安全属性.许多强安全 AKA 协议在基于公钥的密码系

统中被提出[3~6].但是将该属性引入基于口令的认证协议当中却是一个困难问题.尽管有些文献[8,9]提出了在仅仅使用口令的强安全 PAKA 协议,但是他们被证明是不安全的[10].因此,虽然没有给出严格的证明,但是我们认为强安全属性不能在仅仅使用口令的 PAKA 协议中实现.原因在于,我们知道在仅仅使用口令的 PAKA 协议中,用户的秘密值只有他所选取的临时密钥和口令,此外没有任何安全保障.那么当临时密钥泄露后,敌手可以很容易的通过协议中交互的消息来猜测用户的口令.而这一过程往往都是离线的,并不会被用户和服务器所察觉.这也是为什么我们使用智能卡和口令相结合的方式来在 PAKA 协议中实现强安全属性的原因.因为相对于只使用口令的 PAKA 协议来说,基于智能卡和口令的 PAKA 协议多了智能卡这一层安全保障.在协议运行时,尽管用户的临时密钥泄露了,只要智能卡没有泄露,用户所协商出的会话密钥仍然是安全的.

## 6　结论

本文将临时密钥泄露攻击的概念引入基于智能卡和口令的认证协议中,设计了基于智能卡和口令的抗临时密钥泄露攻击的 PAKA 协议.首次提出了包含临时密钥泄露攻击的基于智能卡和口令的 PAKA 协议的安全模型.在该模型下我们基于 GDH 假设证明了所提出方案的安全性.同时,我们讨论了抗临时密钥泄露攻击不能在只使用口令的 PAKA 协议中实现的原因.我们下一步的工作将是用形式化的方法证明抗临时密钥泄露攻击不能在只使用口令的 PAKA 协议中实现,同时给出更加高效的基于智能卡和口令的强安全认证协议.

### 参考文献


[1] 刘云,杨亮,等.一种改进的动态用户认证协议[J].电子学报,2012,41(1):42 – 46.
Liu Yun, Yang Liang, et al. Improved dynamic user authentication protocol[J]. Acta Electronica Sinica, 2012, 41(1): 42 – 46. (in Chinese)

[2] A Akavia, S Goldwasser, V Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks[A]. Proceedings of the 6th Theory of Cryptography Conference (TCC 2009)[C]. Berlin: Springe-Verlag, 2009. 474 – 495.

[3] B Lamacchia, K Lauter, A Mityagin. Stronger security of authenticated key exchange[A]. Proceedings of ProvSec 2007[C]. Berlin: Springe-Verlag, 2007. 1 – 16.

[4] J Alwen, Y Dodis, D Wichs. Leakage-resilientpublic-key cryptography in the bounded-retrieval model[A]. Advances in Cryptology, Crypto 2009[C]. Berlin: Springe-Verlag, 2009. 36 – 54.

[5] T Okamoto. Authenticated key exchange and key encapsulation in the standard model[A]. Advances in Cryptology, Asiacrypt 2007[C]. Berlin: Springe-Verlag, 2007. 474 – 484.

[6] M Kim, A Fujioka, B Ustaoğlu. Strongly secure authenticated key exchange without NAXOS' approach[A]. Proceedings of 4th International Workshop on Security (IWSEC'09)[C]. Berlin: Springe-Verlag, 2009. 174 – 191.

[7] 张延红,陈明.标准模型下强安全的无证书认证密钥协商协议[J].四川大学学报,2013,45(1):125 – 132.
Zhang Yan hong, Chen Min. Strongly secure certificateless authenticated key agreement protocol in standard model[J]. Journal of Sichuan University, 2013, 45(1): 125 – 132. (in Chinese)

[8] K Yoneyama. Efficient and strongly secure password – based server aided key exchange[A]. Proceedings of 9th International Conference on Cryptology in India(Indocrypt'08)[C]. Berlin: Springe – Verlag, 2010. 172 – 184.

[9] J Zhao, D Gu. Provably secure three-party password-based authenticated key exchange protocol[J]. Information Sciences, 2012, 184(1): 310 – 323, 2012.

[10] J Nam, J Paik, D Won. Security analysis of Zhao and Gu's key exchange protocol[EB/OL]. http://onlinepresent.org/proceedings/vol2-2012/15.pdf.

[11] M Bellare, D Pointcheval, P Rogaway. Authenticated key exchange secure against dictionary attacks[A]. Advances in Cryptology, Eurocrypt 2000[C]. Berlin: Springe-Verlag, 2000. 139 – 155.

[12] X Li, J Niu, MK Khan, et al. An enhanced smart card based remote user password authentication scheme[J]. Journal of Network and Computer Applications, 2013, 36(5): 1365 – 1371.


### 作者简介


**李晓伟**　男,1985 年生于吉林通化.西安电子科技大学通信工程学院博士生.研究方向为认证与密钥协商协议、无线网络安全.
E-mail: lixiaowei_xidian@163.com



**张玉清**　男,1966 年生于陕西宝鸡.教授、博士生导师.现任国家计算机网络入侵防范中心主任.研究方向为网络攻防及系统安全、无线网络安全.