计算机科学
COMPUTER SCIENCE

基于口令和智能卡的双因素身份认证与盲云存储方案

王怡, 胡学先, 魏江宏

引用本文

王怡, 胡学先, 魏江宏. 基于口令和智能卡的双因素身份认证与盲云存储方案[J]. 计算机科学, 2024, 51(1): 363-370.

WANG Yi, HU Xuexian, WEI Jianghong. Two-factor Authentication Scheme for Blind Cloud Storage System Based on Password and SmartCard [J]. Computer Science, 2024, 51(1): 363-370.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230700090

基于口令和智能卡的双因素身份认证与盲云存储方案

王 怡 胡学先 魏江宏
中国人民解放军战略支援部队信息工程大学 郑州 450001
(adapter202010@163.com)

摘 要 面向大规模用户数据的存储需求,如何安全地使用云存储技术实现用户数据的远程存取,同时保证数据的可移植性和安全性是当前的一个研究热点。在2022年的USENIX Security会议上,Chen等针对用户仅拥有一个低熵口令的情形,提出了一种高效可移植的盲云存储方案,然而该方案不可避免地继承了口令难以抵抗在线字典攻击的弱点。为弥补单一口令认证方式带来的安全性缺陷,文中提出了一种基于口令和智能卡的双因素身份认证与盲云存储方案。安全性分析和仿真实验结果表明,该方案在保证良好的可移植性、可部署性和盲云存储特性的同时,实现了比纯口令方案更高的安全性,且具有相当的计算和通信效率。

关键词:智能卡;低熵口令;身份认证;双因素;盲云存储
中图分类号 TP309.2

Two-factor Authentication Scheme for Blind Cloud Storage System Based on Password and Smart Card

WANG Yi,HU Xuexian and WEI Jianghong
PLA Strategic Support Force Information Engineering University,Zhengzhou 450001,China

Abstract Aiming at the demand for large-scale data storage,how to securely realize remote access to user data using cloud storage technologies while retaining data portability and security is a research hotspot at present. In USENIX Security 2022,Chen et al. proposed an efficient and portable blind cloud storage scheme for the case where users just hold one low-entropy password. However,the scheme inevitably inherits the weakness of passwords unresistant to online dictionary attack. To compensate the security shortage of password-only authentication,this paper designs a two-factor authentication scheme for blind cloud storage system based on password and smart card. Experimental results show that the proposed scheme not only realizes portability,deployability and blind cloud storage,but also achieves a higher level of security over password-only authentication schemes with equivalently computation and communication efficiency.

Keywords Smart card,Low-entropy password,Identity authentication,Two-factor,Blind cloud storage

1 引言

近年来,商业云存储服务得到了广泛的应用。商业云存储服务可以为不同的企业和个人提供所需的云存储资源,而无需其独立建设基础设施,极大地提高了企业运营效率并节约了建设维护成本,但随之而来的是用户隐私数据在公共云中的安全性与可移植性等诸多矛盾问题。为了兼顾用户在使用商业云存储服务中对可移植性、可部署性及盲云存储的需求,Chen等在2022年USENIX Security会议上提出了一类基于口令的便携盲云存储系统(Portable Blind Cloud Storage,PBCS)[1],该系统基于用户-云(数据)服务器-密钥(应用)服务器三方架构,集成了APP登录模块,允许用户使用一个低熵口令在任何设备上登录应用程序,并安全地访问其存储在云服务器上的数据,且不需要云服务器提供额外的计算。

但是,由于该方案采用基于口令的单一认证方式,因此不可避免地存在一定的安全风险。首先,口令认证存在天然的低熵弱点,易受字典攻击。其次,用户选择的口令具有明显的不均衡性[2],这为针对口令的攻击提供了更多的便利。另外,在实践中发现,随着用户持有的个人账号越来越多,用户不可避免地会重复使用一些口令。微软2014年的一项研究显示,每个用户平均持有25个需要口令的账户,但平均仅维护6.5个口令,这表明一个口令用于近4个不同的账户,一旦在其中一个平台出现口令泄露问题,就将威胁用户在其他多个平台的隐私数据安全。为了弥补这一缺陷,双因素或多因素的

到稿日期:2023-07-12 返修日期:2023-09-20
基金项目:国家自然科学基金(62172433,62172434);河南省自然科学基金(222300420099)
This work was supported by the National Natural Science Foundation of China(62172433,62172434) and Natural Science Foundation of Henan Province,China(222300420099).
通信作者:胡学先(xuexian_hu@hotmail.com)

认证方案被提出，通过将两种或两种以上的认证因素，如口令、公钥证书、智能卡、生物特征等进行组合认证，可以显著提升方案的安全性。其中，基于智能卡的双因素认证方案是应用最为广泛的方案之一。

本文在 Chen 等的 PBCS 方案[1]的基础上，结合基于智能卡的双因素认证方法，提出了一种基于口令和智能卡的双因素身份认证与盲云存储方案，该方案在保证可移植性、可部署性及盲云存储的基础上，通过增加智能卡认证，避免了单一口令认证方式带来的安全性不足，进一步提高了方案的安全性。仿真实验结果表明，与 PBCS 方案相比，该方案的执行效率相当但安全性更高。

1.1 相关工作

1991 年，Chang 等[3]首先将智能卡认证方式引入基于口令的远程身份认证方案中。随后，基于智能卡的双因素认证协议得到进一步推广，并逐渐成为双因素乃至多因素认证协议中最常用的机制之一[4]。在早期的双因素方案中，智能卡被看作是安全的、防篡改的设备[5-6]，但是后续的研究发现，智能卡在侧信道攻击、逆向技术等攻击下并不安全[7]。文献[8]指出，在假定存储在智能卡中的信息可能以某种方式被泄露的情况下设计双因素方案更为可取。Wang 等[7-11]分析了一系列双因素身份认证方案中的安全性问题，讨论了对双因素身份认证方案的安全性评估标准，并提出了通用且有效评估双因素身份认证方案的系统框架。

根据使用的底层密码学组件的不同，现有的基于智能卡的双因素认证方案大致可以分为基于哈希[5-6,12-13]、基于公钥[4,14-17]、基于椭圆曲线[18-21]、基于椭圆曲线上的双线性对[22-24]以及基于混沌映射[25-26]等几类。其中基于公钥的方案往往具有更高的安全性，但计算开销大且运行效率较低，不适用于便携式终端。基于哈希的方案使用单向哈希、消息认证码和按位异或等运算作为组件，相比其他方案更为轻量。

文献[27-28]基于时间戳技术，提出了基于智能卡的三方口令认证与密钥协商协议，但是在多用户网络中，时间戳会带来复杂的时间同步问题，并且容易发生通信与计算错误。文献[29]采用随机挑战-响应方案代替时间戳，避免了时钟同步问题，更适合在用户数量较大的云存储系统中使用。文献[5]于 2014 年提出了一种基于对称加密的轻量级双因素认证方案，仅使用哈希和异或操作，并启发式地证明了其方案能够抵御重放攻击、中间人攻击、智能卡泄露攻击等常见攻击。2016 年，Chang 等[6]指出文献[5]中的方案并不能抵抗智能卡泄露攻击且易受传感器节点欺骗攻击等其他攻击，同时提出了两种不同的改进方案。2018 年，Wang 等[8]在双因素方案安全性评估框架[7-11]的基础上，基于 CDH 假设和哈希函数等组件，提出了一类具有更高安全性的双因素认证方案。Wang 的方案[8]通过在智能卡中存储一个长期密钥以及用于保护该密钥的额外参数实现了“模糊验证器”，同时通过“honey-words”检测并限制读取智能卡内参数的次数，来避免在线字典攻击。

1.2 主要贡献和结构

本文借鉴了文献[8]的研究思路，在 Chen 等的 PBCS 方案[1]的基础上进行了改进。为了实现便携性和高效性，采用基于哈希的密码学组件设计了基于口令与智能卡的双因素认证方案，并在 Wang 等提出的安全性评估标准[8-12]启发下开展了安全性分析。

本文的主要贡献如下：

1)将 PBCS 方案中基于单一口令的认证方式修改为基于口令和智能卡的双因素认证，用户需要持有口令和经密钥服务器认证的智能卡才能完成登录并访问存储在云端的私有数据，在保持原有的可部署性、可移植性与盲云存储的基础上，实现了更高的安全性。

2)在随机谕示模型下，采用启发式分析方法证明了本文方案具备对口令泄露攻击、智能卡泄露攻击的安全性，能够实现包括认证安全、主密钥安全等在内的安全目标。

3)在实验环境下验证了本文方案的可用性，并对方案的计算与通信效率进行了比较分析。实验与分析的结果表明，本文方案在不损失计算与通信效率的情况下，相比 Chen 等的 PBCS 方案[1]实现了更高的安全性。

本文第 2 章介绍了需要用到的基础知识和方案构建使用的组件；第 3 章详细描述了所提方案的执行过程；第 4 章给出了启发式安全性分析；第 5 章展示了本文方案的实验仿真结果及与其他同类型方案的效率对比情况；最后总结全文。

2 预备知识

2.1 通用注册登录机制

为了满足可移植性和可部署性的需求，与 PBCS 方案[1]一样，本文采用云服务器和应用程序内置的通用注册登录接口进行用户身份验证。由于在实际中，不同的云存储服务平台和应用程序提供的注册登录接口存在差异，在不失一般性的前提下，在此将其进行抽象为注册算法 AuthReg 和登录算法 Login。

注册算法：$\gamma_{id}=\mathrm{AuthReg}(id,auth_{id})$。用户向服务器提交身份标识 id 和认证凭据 $auth_{id}$，注册算法输出 γ_{id}，并将其存储于云服务器端，作为下一次用户登录的存根。用户在登录时，需提交相同的身份标识 id 和认证凭据 $auth_{id}$ 方能完成认证。

登录算法：$b=\mathrm{Login}(id,auth_{id},\gamma_{id})$。登录时，用户向服务器提交身份标识 id 和认证凭据 $auth_{id}$，服务器使用存储的存根 γ_{id} 对用户身份进行验证，若验证成功，则输出结果 $b=1$，否则输出 $b=0$，表示验证失败。

通用注册登录机制的安全性的定义如下。

正确性(correctness)：如果 γ_{id} 是注册算法 AuthReg 在输入 $(id,auth_{id})$ 下的输出，则登录算法 $\mathrm{Login}(id,auth_{id},\gamma_{id})=1$ 始终成立。

ε-安全性(ε-security)：称注册登录方案(AuthReg, Login)具有 ε-安全性，如果对于任意 PPT 攻击者 $\mathcal{A}$，如下表达式成立：

$$\Pr\left[\begin{array}{l} id^* \leftarrow_{\$} \mathcal{A} \\ auth_{id^*} \leftarrow_{\$} \chi \\ \gamma_{id^*} \leftarrow_{\$} \mathrm{AuthReg}(id^*, auth_{id^*}), : b=1 \\ auth_{id^*}{}' \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\mathrm{AuthReg}(\cdot,\cdot)}, \mathcal{O}_{\mathrm{Login}(\cdot,\cdot,\cdot)}} \\ b=\mathrm{Login}(id^*, auth_{id^*}{}', \gamma_{id^*}) \end{array}\right] < \varepsilon$$

其中,χ 表示用户登录凭证 $auth_{id^*}$ 的分布;$\mathcal{O}_{\mathrm{AuthReg}(\cdot,\cdot)}$ 表示模拟注册算法 AuthReg 的谕示图灵机,对于输入(id^*,$auth_{id^*}$),输出 $\gamma_{id^*}=\mathrm{AuthReg}(id^*, auth_{id^*})$;$\mathcal{O}_{\mathrm{Login}(\cdot,\cdot,\cdot)}$ 表示模拟登录算法 Login 的谕示图灵机,对于输入(id^*,$auth_{id^*}$,γ_{id^*}),输出 $b=\mathrm{Login}(id^*, auth_{id^*}, \gamma_{id^*})$,同时进行计数,限制对同一个 id^* 进行查询的次数不超过 B 次。

在实践中,对用户认证凭据进行加密存储是注册登录机制中常见的方法之一,下面以加密存储方法为例,给出一类注册登录机制的实例,描述如下。

注册算法 AuthReg:$\gamma_{id} \leftarrow \varepsilon.Enc(id, auth_{id})$:用户 U_{id} 注册时,输入(id,$auth_{id}$),平台使用加密算法 $\varepsilon.Enc(\cdot)$ 对输入的注册信息进行加密后生成存根 γ_{id},存于服务器内。

登录算法 Login:$b \leftarrow \mathrm{verify}\ \gamma_{id}=\varepsilon.Enc(id, auth_{id}{}')$:用户 U_{id} 登录时,输入(id,$auth_{id}{}'$),平台从服务器中读取存根 γ_{id},验证 $\gamma_{id}=\varepsilon.Enc(id, auth_{id}{}')$ 是否成立,若成立,则返回 $b=1$,登录成功;否则,返回 $b=0$,登录失败。

2.2　KEM-DEM 混合加密机制

KEM-DEM 混合加密机制分为两个部分:密钥封装机制(KEM)和数据封装机制(DEM),前者在通信双方安全地共享一个对称密钥,后者利用该密钥使用对称加密算法加密消息。其通用构造步骤如图 1 所示[30]。

图 1　混合加密机制

Fig. 1　Hybrid encryption mechanism

KEM 部分定义为一种算法三元组[31-32](KEM_Gen,KEM_Enc,KEM_Dec):

1)$(pk,sk) \leftarrow \mathrm{KEM\_Gen}(1^\lambda)$:密钥生成算法,输入安全参数 λ,输出一对公私钥(pk,sk)。

2)$(K,c) \leftarrow \mathrm{KEM\_Enc}(pk)$:密钥封装算法,输入公钥 pk,输出对称密钥 K 和对应的密文 c。

3)$(K) \leftarrow \mathrm{KEM\_Dec}(sk,c)$:密钥解封算法,输入私钥 sk 和密文 c,输出对称密钥 K 或错误标识符 $\perp$。

DEM 部分定义为一种算法二元组[31-32](DEM_Enc,DEM_Dec):

1)$(C) \leftarrow \mathrm{DEM\_Enc}(K,m)$:加密算法,输入密钥 K 和任意长度的消息 m,输出密文 C。

2)$(m) \leftarrow \mathrm{DEM\_Dec}(K,C)$:解密算法,输入密钥 K 和密文 C,输出消息 m 或错误标识符 $\perp$。

2.3　基于身份的不经意伪随机函数

不经意伪随机函数[33-37](Oblivious Pseudorandom Function,OPRF)使得一个持有秘密 x 的用户通过与持有密钥 k 的服务器交互,不经意地生成一个伪随机函数值 $F(k,x)$,并且满足用户无法获知服务器的密钥 k 且服务器无法获知生成的结果 $F(k,x)$,如图 2 所示。

图 2　不经意伪随机函数

Fig. 2　OPRF

根据实现方法的不同,OPRF 可以分为基于 Diffie-Hellman、基于不经意传输协议(OT)、基于 RSA 等不同类型。文献[33,37]介绍了基于 OMGDH(One More Gap Diffie Hellman)问题的 DH-OPRF,如图 3 所示。其中,交互双方被称为发送方(Sender)与接收方(Receiver),发送方持有密钥 $k \in \mathbb{Z}_p^*$,接收方持有秘密 $x \in \{0,1\}^*$,哈希函数 $H:\{0,1\}^* \rightarrow \mathbb{Z}_p^*$。接收方对函数进行调用并获取最终的输出值。调用时,接收方首先随机选取 $r \in \mathbb{Z}_p^*$,将 $H(x)^r$ 发送至发送方,发送方返回 $(H(x)^r)^k$,接收方计算 $H'(H(x)^k)=H'(((H(x)^r)^k)^{1/r})$,即为函数输出结果。其中 $H':\mathbb{Z}_p^* \rightarrow \{0,1\}^\lambda$。

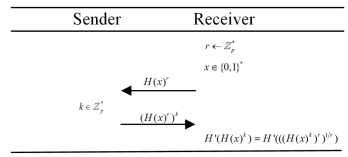


图 3　基于 DH 的不经意伪随机函数

Fig. 3　DH-OPRF

Chen 等[1]基于 DH-OPRF 提出了一种基于用户 ID 的不经意伪随机函数 IBOPRF(Identity- Based OPRF),使得输出的值可以与用户的唯一身份标识进行关联,从而更加适用于单服务器对多用户的场景。具体来说,Chen 等提出的 IBOPRF 方案 $\mathcal{F}(k,ID,x)$ 被描述为一个挑战-响应机制[1],包含 4 个语句(Setup,CEval_1,SEval_1,CEval_2),详细描述如下。

$(pp,k) \leftarrow \mathrm{Setup}(1^\lambda)$:给定一个安全参数 λ,Setup 语句为服务器生成一个 λ 长度的密钥 k 并输出公共参数 $pp=(p,\mathbb{G},H_1,H_2,H_3)$,其中 p 为安全素数,$\mathbb{G}$ 为 p 阶乘法群,$H_1:\{0,1\}^k \rightarrow \mathbb{Z}_p$,$H_2:ID \times \{0,1\}^\lambda \rightarrow \mathbb{Z}_p^*$,$H_3:\{0,1\}^k \times \mathbb{Z}_p \rightarrow \{0,1\}^\lambda$ 为 3 个不同的哈希函数。

$(ch,r) \leftarrow \mathrm{CEval}_1(pp,ID,x)$:用户选择身份 ID 和秘密 x,根据输入(pp,ID,x),调用 CEval_1 语句计算(ch,r),其中 $r \leftarrow_{\$} \mathbb{Z}_p^*$ 作为用户的内部状态保存在用户端,$ch=H_1(x)^r$ 作为挑战发送给服务器。

$rp \leftarrow \mathrm{SEval}_1(pp,ch,k)$:服务器收到挑战后,根据输入($pp$,$ch$,$k$),调用 SEval_1 语句生成一个基于 ID 的伪随机函数值 $kid=H_2(ID,k)$,然后计算响应值 $rp=ch^{kid}$ 发送给用户。

$y \leftarrow \mathrm{CEval}_2(pp,rp,r,x)$:用户收到响应值后,读取内部状态 r,根据输入(pp,rp,r,x),调用 CEval_2 语句计算 $y=H_3(x,rp^{1/r})$,即为 IBOPRF 的最终输出结果。

IBOPRF 方案 $\mathcal{F}(k,ID,x)$ 满足以下 3 种安全性。

唯一性(Uniqueness):称一个 IBOPRF 方案满足唯一性,如果对任意的 $(ID',x') \neq (ID,x)$,IBOPRF 方案输出 $\mathcal{F}(k,ID',x')=\mathcal{F}(k,ID,x)$ 的概率是可忽略的。

伪随机性(Pseudorandomness):称一个 IBOPRF 方案

满足(ε,d,B)-伪随机性，若对于任意 PPT 攻击者 $\mathcal{A}$ 都成立：

$$\Pr\left[\begin{array}{l} b\leftarrow_{\$}\{0,1\} \\ (pp,k)\leftarrow_{\$}\mathrm{Setup}(1^{\lambda}) \\ ID^{*}\leftarrow_{\$}\{0,1\}^{\lambda} \\ x^{*}\leftarrow_{\$}\mathcal{D} \\ (ch,r)\leftarrow_{\$}\mathrm{CEval}_1(pp,ID^{*},x^{*}),\ :b'=b \\ rp\leftarrow_{\$}\mathrm{SEval}_1(pp,ID^{*},ch,k) \\ y_0\leftarrow_{\$}\mathrm{CEval}_2(pp,rp,r,x^{*}) \\ y_1\leftarrow_{\$}\{0,1\}^{\lambda} \\ b'\leftarrow_{\$}\mathcal{A}^{\mathrm{S}(\cdot,\cdot)}(ID^{*},y_b) \end{array}\right]<\frac{1}{2}+\varepsilon$$

其中，$\mathcal{D}$ 表示 x 的分布，d 表示分布 $\mathcal{D}$ 的最小熵，$\mathrm{S}(\cdot,\cdot)$ 表示模拟服务器端的谕示图灵机，对于输入(pp,ID^{*},ch,k)，计算 $rp\leftarrow\mathrm{SEval}_1(pp,ID^{*},ch,k)$，同时进行计数，限制对同一个 ID^{*} 查询次数不超过 B 次。

不经意性(Obliviousness)：称一个 IBOPRF 方案满足(ε,d,k)-不经意性，若对于任意 PPT 攻击者 $\mathcal{A}$ 都成立：

$$\Pr\left[\begin{array}{l} b\leftarrow_{\$}\{0,1\} \\ (pp,k)\leftarrow_{\$}\mathrm{Setup}(1^{\lambda}) \\ ID^{*}\leftarrow_{\$}\{0,1\}^{\lambda} \\ x^{*}\leftarrow_{\$}\mathcal{D} \\ (ch,r)\leftarrow_{\$}\mathrm{CEval}_1(pp,ID^{*},x^{*}),\ :y_0\in\{y_1,\cdots,y_k\} \\ rp\leftarrow_{\$}\mathcal{A} \\ y_0\leftarrow_{\$}\mathrm{CEval}_2(pp,rp,r,x^{*}) \\ y_1,\cdots,y_k\leftarrow_{\$}\mathcal{A}^{\mathrm{C}(\cdot)}(k) \end{array}\right]<\varepsilon$$

其中，$\mathcal{D}$ 表示 x 的分布，d 表示分布 $\mathcal{D}$ 的最小熵，k 表示攻击者对 y 进行有限次查询的次数，$C(\cdot)$表示模拟客户端的谕示图灵机，对每一次查询，输出$(ch,r)\leftarrow\mathrm{CEval}_1(pp,ID^{*},x^{*})$。

3 基于口令和智能卡的双因素身份认证方案

本文构造了一类基于口令和智能卡的双因素认证方案，用户使用 ID 和一个低熵口令短语 PP 在密钥服务器上进行注册后获得一个经密钥服务器认证的智能卡，通过智能卡与用户持有的低熵口令短语 PP 共同生成一个高熵口令 PWU 作为 IBOPRF 的输入，避免了对低熵口令短语的字典攻击。在密钥服务器的辅助下，通过 IBOPRF 对高熵口令 PWU 进行强化，生成一个强化口令 pwd，使用 pwd 在云服务器进行注册登录。注册及登录过程中，用户仅需插入智能卡并输入 ID 和 PP，即可通过任意可接入智能卡的终端便捷地访问云上的数据。同时，本文方案使用云服务平台提供的通用注册登录接口进行注册登录，无需额外的计算服务；通过密钥服务器在发放智能卡时写入签名信息，实现了用户到密钥服务器的双向认证。

为了保证用户数据的安全性，本文方案使用 KEM-DEM 混合加密机制。基于 KEM 进行主密钥封装：通过智能卡生成加密主密钥，将加密主密钥封装后存储于密钥服务器，在恢复主密钥时，需由智能卡、密钥服务器与云服务器三方共同协作才能对加密主密钥解封。基于 DEM 进行数据封装：在智能卡内对用户数据进行加密封装后上传至云服务器，实现了盲云存储。

具体来说，本方案分为注册、数据上传和数据下载 3 个阶段，参与方包括密钥服务器 KS、云服务器 CS 和用户 U_i，其中密钥服务器 KS 持有私钥 msk，用户 U_i 持有唯一身份标识 ID_i 和口令短语 PP_i。

3.1 注册阶段

注册阶段分为注册到密钥服务器 KS 与注册到云服务器 CS 两个部分，详细流程如图 4 所示。注册到 KS 的步骤如下：

1)用户 U_i 选择随机数 α，计算 $PWU=h(ID_i\parallel PP_i\parallel\alpha)$，将 ID_i，PWU 发送给 KS 进行注册。

2)如果 U_i 是首次注册到 KS，则 KS 在列表 $\mathcal{L}^{\mathcal{K}}$ 中为其创建并维护一个条目 $\mathcal{L}^{\mathcal{K}}_{ID_i}$，同时生成一个随机数 β 存入 $\mathcal{L}^{\mathcal{K}}_{ID_i}$，否则注册失败。

3)KS 计算 $s=h(ID_i\parallel msk\parallel\beta)$，$k=s\oplus PWU$，$t=\beta\oplus h(ID_i\parallel PWU)$，将$\mathbb{G}$，$h(\cdot)$，$p$，$t$，$k$ 写入智能卡并发放给用户，其中$\mathbb{G}$，p 为公共参数，p 是一个安全素数，$\mathbb{G}$是一个 p 阶乘法群。

4)U_i 收到智能卡后，计算 $P_1=\alpha\oplus h(ID_i\parallel PP_i)$，$P_2=t\oplus h(ID_i\parallel\alpha)$，将 P_1，P_2 存入智能卡并删除 t。

注册到 CS 的步骤如下：

1)智能卡随机选择 $r\in\mathbb{Z}_q^{*}$ 并计算 $a=H_1^{r}(PWU)$，将 ID_i，a 发送给 KS，通过 IBOPRF 机制生成登录凭据 pwd。

2)智能卡将 ID_i，pwd 发送至 CS，通过 AuthReg 算法在 CS 注册。同样地，如果 U_i 是首次注册到 CS，则 CS 在列表 $\mathcal{L}^{\mathcal{C}}$ 中为其创建并维护条目 $\mathcal{L}^{\mathcal{C}}_{ID_i}$，在 $\mathcal{L}^{\mathcal{C}}_{ID_i}$ 中存入 AuthReg 算法输出的存根 γ_{ID_i}，完成注册。

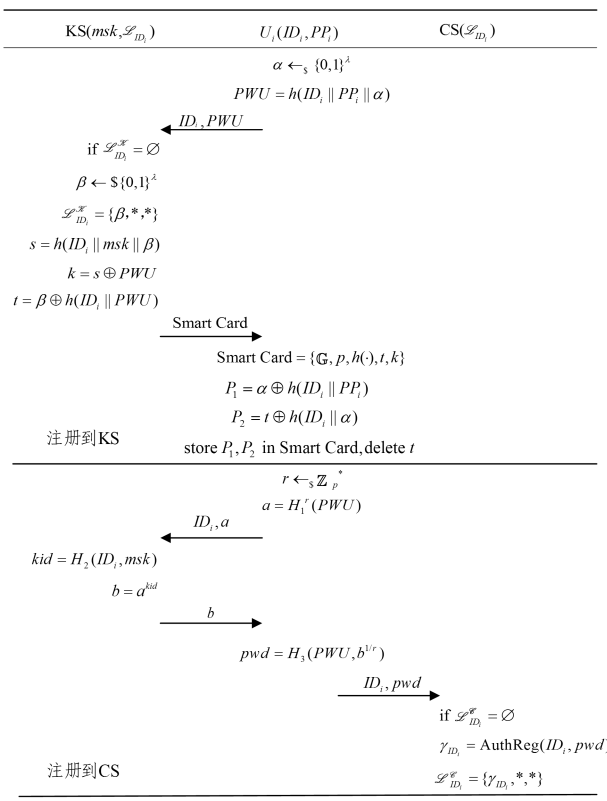


图 4 注册阶段

Fig. 4 Registration phase

注册完成后，智能卡内存储$\{\mathbb{G},h(\cdot),p,k,P_1,P_2\}$。在后续的操作中，用户$U_i$只需在终端插入智能卡，向智能卡输入$ID_i$和$PP_i$，由智能卡进行后续的运算与通信。

3.2　数据上传阶段

用户U_i通过智能卡生成一个主密钥mk，基于KEM机制对其加密后存入密钥服务器KS，基于DEM机制对需要存储的文件数据$Data$加密后存入云服务器CS，服务器各自维护相应的列表，如图5所示。具体步骤如下：

1)U_i在终端插入智能卡，输入ID_i和PP_i。

2)智能卡计算$\alpha=P_1\oplus h(ID_i\|PP_i)$，$PWU=h(ID_i\|PP_i\|\alpha)$。

3)智能卡随机选择$r\in\mathbb{Z}_q^*$并计算$a=H_1^r(PWU)$，将ID_i，a发送给KS，通过IBOPRF机制生成登录凭据pwd，将ID_i，pwd发送至CS并通过Login算法登录。

4)登录成功后，智能卡生成主密钥mk和一个随机种子sid，将sid发送至CS存入$\mathscr{L}_{ID_i}^{\mathscr{C}}$。

5)智能卡计算$k_1=KDF_1(k,PWU,sid)$，$k_2=KDF_2(k,PWU,sid)$，使用k_1对mk进行加密，即$ct=Enc_{k_1}(mk)$，使用k_2计算ct的哈希值用于完整性检验，即$\tau=H_4(ct,k_2)$。

6)智能卡计算$\beta=P_2\oplus h(ID_i\|\alpha)\oplus h(ID_i\|PWU)$，$\nu=H_5(k\oplus PWU,\beta)$，将$ID_i$，$ct$，$\tau$，$\nu$发送至KS；对$Data$加密，即$M=\text{DEM_Enc}(mk,Data)$，将$M$发送至CS存入$\mathscr{L}_{ID_i}^{\mathscr{C}}$。

7)KS收到ID_i，ct，τ，ν后，首先从$\mathscr{L}_{ID_i}^{\mathscr{K}}$中读取$\beta$，然后计算$s=h(ID_i\|msk\|\beta)$并验证$\nu=H_5(s,\beta)$是否成立，若成立，则用户身份验证通过，KS将$ct$，$\tau$存入$\mathscr{L}_{ID_i}^{\mathscr{K}}$。

数据上传阶段完成。

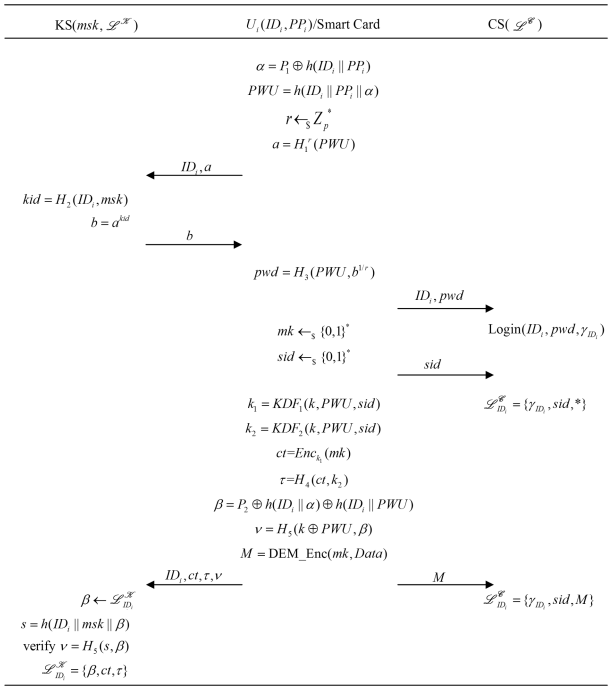


图5　数据上传阶段

Fig. 5　Data upload phase

3.3　数据下载阶段

用户U_i通过智能卡登录密钥服务器KS和云服务器CS，读取并恢复主密钥mk和文件数据$Data$，如图6所示。具体步骤如下：

1)U_i在终端插入智能卡，输入ID_i和PP_i。

2)智能卡计算$\alpha=P_1\oplus h(ID_i\|PP_i)$，$PWU=h(ID_i\|PP_i\|\alpha)$，随机选择$r\in\mathbb{Z}_q^*$并计算$a=H_1^r(PWU)$，$\beta=P_2\oplus h(ID_i\|\alpha)\oplus h(ID_i\|PWU)$，$\nu=H_5(k\oplus PWU,\beta)$，将$ID_i$，$a$，$\nu$发送给KS。

3)收到ID_i，a，ν后，KS从$\mathscr{L}_{ID_i}^{\mathscr{K}}$中读取$\beta$，计算$s=h(ID_i\|msk\|\beta)$，验证$\nu=H_5(s,\beta)$是否成立，若成立，则用户身份验证通过。而后KS从$\mathscr{L}_{ID_i}^{\mathscr{K}}$中读取$ct$，$\tau$，并计算IBOPRF的中间参数$b$，将$b$，$ct$，$\tau$发送给智能卡。

4)收到b，ct，τ后，智能卡首先利用b计算登录凭据pwd，然后将ID_i，pwd发送给CS，通过Login算法进行登录。

5)登录成功后，CS从$\mathscr{L}_{ID_i}^{\mathscr{C}}$中取出$M$，$sid$并发送给智能卡。

6)智能卡收到M，sid后，恢复密钥k_1，k_2并验证$\tau=H_4(ct,k_2)$，验证通过后恢复主密钥mk：$mk=Dec_{k_1}(ct)$，使用mk恢复文件数据$Data$：$Data=\text{DEM\_Enc}(mk,M)$。

数据下载阶段完成。

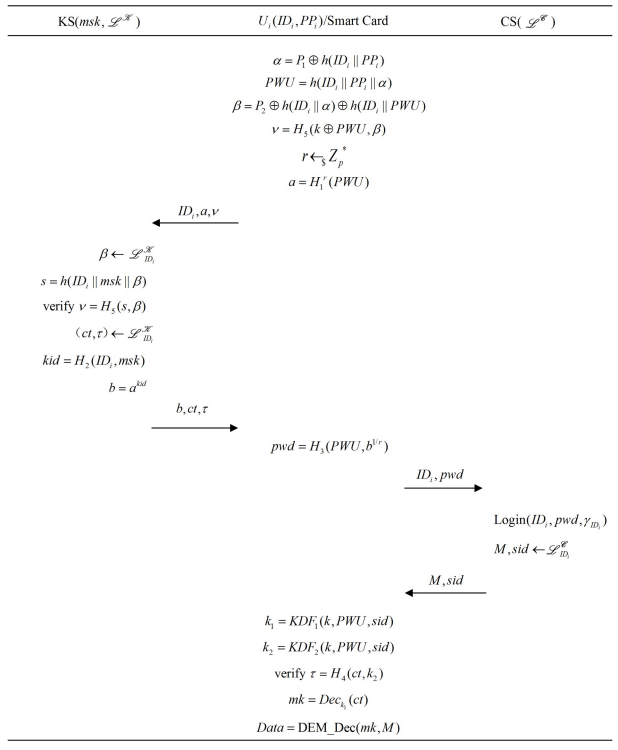


图6　数据下载阶段

Fig. 6　Data download phase

4　安全性分析

本章通过启发式安全性分析证明所提方案能够抵抗智能卡泄露攻击、口令泄露攻击、服务器腐化攻击等常见攻击，实现双因素认证安全、主密钥安全等安全目标。与PBCS系统[1]一样，本文方案适用于服务器持有PKI颁发的证书而客户端没有的运行环境，允许在诚实客户端与服务器之间建立经服务器认证的TLS安全通道，避免攻击者读取或更改诚实客户端与诚实服务器之间的通信。

4.1 双因素认证安全

对口令泄露攻击：假设攻击者 $\mathcal{A}$ 获取了某个用户 U_i 的口令短语 PP_i。一方面，攻击者 $\mathcal{A}$ 若想假冒用户 U_i 向密钥服务器 KS 进行认证，则需要计算认证元 $\nu=H_5(k\oplus PWU,\beta)$，其中 β,k 由 KS 生成，经混淆后存储在智能卡内，$PWU=h(ID_i\|PP_i\|\alpha)$由智能卡生成，哈希函数 $h(\cdot)$与生成 α 的相关信息同样存储在智能卡内。攻击者 $\mathcal{A}$ 在无法获取智能卡内信息的情况下，只能对 β,k 以及 PWU 进行猜测来构造认证元 ν。因此，在随机谕示模型下，攻击者 $\mathcal{A}$ 无法实现到 KS 的认证。

另一方面，攻击者 $\mathcal{A}$ 若想假冒用户 U_i 向云服务器 CS 进行认证，则需要计算 IBOPRF 生成的 $pwd=H_3(PWU,H_1^{H_2(ID_i,msk)}(PWU))$。由于 IBOPRF 的伪随机性和不经意性，攻击者 $\mathcal{A}$ 只能同时对 PWU 和 msk 进行猜测来构造 pwd。因此，在随机谕示模型下，攻击者 $\mathcal{A}$ 也无法实现到 CS 的认证。

对智能卡泄露攻击：假设攻击者 $\mathcal{A}$ 获得了智能卡内的所有信息，即$\{\mathbb{G},h(\cdot),p,k,P_1,P_2\}$，

其发起离线字典攻击的方式如下：

猜测用户口令短语 PP_i'，计算 $\alpha'=P_1\oplus h(ID_i\|PP')$，$PWU'=h(ID_i\|PP'\|\alpha')$，$\beta'=P_2\oplus h(ID_i\|\alpha')\oplus h(ID_i\|PWU')$；验证 $k=h(ID_i\|msk\|\beta')\oplus PWU'$ 是否成立。

在密钥服务器 KS 未被腐化的情况下，攻击者 $\mathcal{A}$ 无法获知 KS 持有的私钥 msk，只能对 msk 进行猜测来完成上述验证，其概率是可忽略的，因此本文方案具有对离线字典攻击的安全性。

其发起在线字典攻击的方式如下：

猜测用户口令短语 PP_i'，计算 $\alpha'=P_1\oplus h(ID_i\|PP')$，$PWU'=h(ID_i\|PP'\|\alpha')$，将 PWU' 发送给密钥服务器 KS，利用 IBOPRF 机制生成 pwd'，而后将(ID_i,pwd')提交给云服务器 CS 进行身份认证，则攻击者 $\mathcal{A}$ 成功的概率至多为 $O(B/|\mathcal{D}|)$，其中 B 表示至多允许用户对同一个身份标识 ID_i 进行认证请求的次数，$|\mathcal{D}|$ 表示用户口令短语的字典空间大小。

4.2 主密钥安全

对腐化的密钥服务器：假设攻击者 $\mathcal{A}$ 腐化了密钥服务器 KS，从而可以获取(msk,β,ct,τ)等信息，我们有以下结论：

1）腐化的密钥服务器 KS 无法破坏加密主密钥 mk 的完整性。否则用户 U_i 在数据下载阶段将得到 $\tau\neq H_4(ct,k_2)$，从而验证失败。

2）腐化的密钥服务器 KS 无法破坏加密主密钥的保密性。在随机谕示模型下，攻击者 $\mathcal{A}$ 若想利用 ct 恢复主密钥 mk，只能通过以下几种方式：

(1)猜测 PWU'，利用 IBOPRF 机制生成 pwd'，向云服务器 CS 提交(ID_i,pwd')进行认证以获取 sid。由于 IBOPRF 的不经意性，$\mathcal{A}$ 无法从之前的输出中获得任何关于正确的 pwd 或 PWU 的信息，只能进行盲目猜测，其成功的概率是一个可忽略的函数 $negl(\lambda)$，λ 为安全参数。

(2)猜测 PWU' 及 sid'，计算 $k'=h(ID_i\|msk\|\beta')\oplus PWU'$，$k_1'=KDF_1(k',PWU',sid')$，使用 k_1' 对 ct 解密以获取 mk。其成功的概率是可忽略的。

综上，本文方案具有对腐化的密钥服务器的安全性。

对腐化的云服务器：假设攻击者 $\mathcal{A}$ 腐化了云服务器 CS，从而可以获得(sid,M)等信息。其破坏主密钥安全的唯一途径是伪装成用户 U_i 从密钥服务器 KS 获取 ct，因此需要构造认证元 ν。在无法获取智能卡和 PP_i 的情况下，攻击者只能同时对 k,PWU,β 这 3 个参数(或 $msk/s,\beta$ 这两个参数)进行猜测，其成功的概率是可忽略的。因此，在随机谕示模型下，本文方案对腐化的云服务器具有安全性。

5 方案效率比较

本章对所提方案与 Chen 等的 PBCS 方案[1]进行了效率与安全性两方面的比较。仿真实验环境为 VMware，Intel © Core™ i7-9700 CPU @ 3.00 GHz，15.6 GB 内存，Ubuntu 18.04.6 LTS 64 位操作系统，实验代码使用 Python 编写，并基于 Charm，Crypto 等库设计完成。

我们选取了不同的安全参数 λ、不同大小的文件数据，对两种方案的运行效率进行了测试，测试结果为 100 次运行的平均时间，如图 7、图 8 所示。

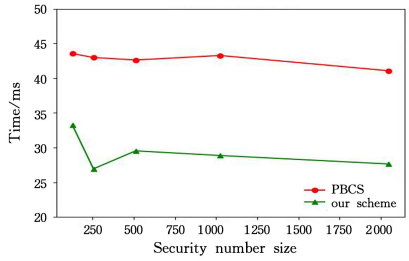


图 7 文件大小为 1MB 时不同安全参数下的运行效率比较

Fig. 7 Efficiency comparison with different security parameters when file size is 1MB

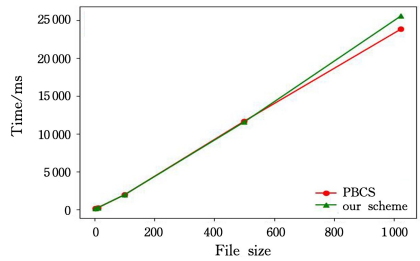


图 8 安全参数为 1024 时不同文件大小下的运行效率比较

Fig. 8 Efficiency comparison under different fille sizes while security parameter is 1024

进一步地，我们对方案中各个参与方在不同阶段的计算效率与通信轮次进行了比较，结果如表 1 所列。其中，T_H 表示进行 1 次单向哈希运算的时间开支，T_E 表示 1 次加/解密运算的时间开支，T_M 表示 1 次 HMAC 运算的时间开支(我们将密钥派生函数也视作此类运算)，T_L 表示调用 1 次登录/注册算法的时间开支，T_I 表示 1 次模幂运算的时间开支。据 Wang 等[9]的研究，上述运算时间存在 $T_E\gg T_I\gg T_M>T_H$ 的不等式关系，T_L 对于不同的云服务器平台存在不同程度的差异，难以进行对比，但不应该被忽视。其余运算如按位异或运算、并联运算等仅需要较小的时间开支，在此忽略不计。

表 1　与 PBCS 方案的计算与通信效率比较

Table 1　Computation and communication efficiency comparison with PBCS

方案	运行阶段	用户/智能卡	密钥服务器	云服务器	轮次
PBCS	注册阶段	$2T_H+1T_M+1T_I$	$1T_H+1T_L$	$1T_L$	5
	数据上传阶段	$2T_H+4T_M+2T_E+1T_I$	$1T_H+1T_L$	$1T_L$	7
	数据下载阶段	$2T_H+4T_M+2T_E+1T_I$	$1T_H+1T_L$	$1T_L$	6
总计		$9T_H+9T_M+4T_E+6T_L+3T_I$			
本文方案	注册阶段	$5T_H+1T_I$	$3T_H$	$1T_L$	5
	数据上传阶段	$6T_H+4T_M+2T_E+1T_I$	$2T_H+1T_M$	$1T_L$	6
	数据下载阶段	$6T_H+4T_M+2T_E+1T_I$	$2T_H+1T_M$	$1T_L$	4
总计		$24T_H+10T_M+4T_E+3T_L+3T_I$			

根据图 7、图 8 的实验测试结果，本文方案在运行效率上较 PBCS 方案有小幅提升，但总体差距不大。根据表 1 的量化分析结果，本文方案相比 PBCS 方案增加了哈希运算与 HMAC 运算的次数，但减少了 3 个通信轮次和 3 次登录/注册操作，并且没有增加加解密运算和模幂运算的次数，而这两项运算的时间开支在方案整体运行过程中占据了较大的比重[9]。综上所述，总体而言，本文方案与 PBCS 方案在运行效率上基本相当。

表 2 列出了在安全性方面的比较结果。与 PBCS 方案一样，本文方案也具备对腐化的密钥服务器和腐化的云服务器的安全性。同时，由于增加了智能卡这一认证因素，本文方案还提供了用户到密钥服务器的双向认证和多因素安全的保障，因此具有更高的安全性。

表 2　与 PBCS 方案的安全性比较

Table 2　Security comparison with PBCS

方案	安全模型	对腐化密钥服务器的安全性	对腐化云服务器的安全性	多因素安全	双向认证
PBCS	随机预言模型	是	是	否	否
本文方案	随机预言模型	是	是	是	是

结束语　针对用户在使用商业云存储服务中存在的盲存储与可移植性之间的矛盾问题，同时为了弥补单一口令认证方式的安全性缺陷，本文在已有的通用盲云存储方案的基础上，提出了一种面向用户-云服务器-密钥服务器三方架构的基于智能卡和口令的双因素身份认证与盲云存储方案。基于启发式分析方法分析了方案的安全性，并在实验环境下与原方案进行了运行效率和安全性方面的比较。结果表明，本文方案在不显著增加额外的计算和通信开销的情况下，实现了更高的安全目标。

参 考 文 献

[1] CHEN L, LI Y N, TANG Q, et al. End-to-Same-End Encryption: Modularly Augmenting an App with an Efficient, Portable, and Blind Cloud Storage[C]// Proceedings of the 31st USENIX Security Symposium. Boston: USENIX Association, 2022: 2353-2370.

[2] WANG D, WANG P. On The Implications of Zipf's Law in Passwords[C]// Computer Security — ESORICS 2016. Heraklion: Springer International Publishing, 2016: 111-131.

[3] CHANG C C, WU T C. Remote Password Authentication with Smart Cards[J]. Computers and Digital Techniques, IEEE Proceedings, 1991, 138(3): 165-168.

[4] WANG C, WANG D, XU G, et al. A Lightweight Password-Based Authentication Protocol Using Smart Card[J]. International Journal of Communication Systems, 2017, 30(16): e3336.

[5] TURKANOVIĆ M, BRUMEN B, HÖLBL M. A Novel User Authentication and Key Agreement Scheme for Heterogeneous Ad Hoc Wireless Sensor Networks, Based on The Internet of Things Notion[J]. Ad Hoc Networks, 2014, 20: 96-112.

[6] CHANG C C, LE H D. A Provably Secure, Efficient, and Flexible Authentication Scheme for Ad Hoc Wireless Sensor Networks[J]. IEEE Transactions on Wireless Communications, 2016, 15(1): 357-366.

[7] WANG D, GU Q, CHENG H, et al. The Request for Better Measurement: A Comparative Evaluation of Two-Factor Authentication Schemes[C]// Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. Xi'an: ACM, 2016: 475-486.

[8] WANG D, HE D, WANG P, et al. Anonymous Two-Factor Authentication in Distributed Systems: Certain Goals Are Beyond Attainment[J]. IEEE Transactions on Dependable and Secure Computing, 2015, 12(4): 428-442.

[9] WANG D, WANG P. Two Birds with One Stone: Two-Factor Authentication with Security Beyond Conventional Bound[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 15(4): 708-722.

[10] WANG D, LI W, WANG P. Measuring Two-Factor Authentication Schemes for Real-Time Data Access in Industrial Wireless Sensor Networks[J]. IEEE Transactions on Industrial Informatics, 2018, 14(9): 4081-4092.

[11] WANG D, WANG P. On The Anonymity of Two-Factor Authentication Schemes for Wireless Sensor Networks: Attacks, Principle and Solutions[J]. Computer Networks, 2014, 73: 41-57.

[12] SRINIVAS J, DAS A K, KUMAR N, et al. Cloud Centric Authentication for Wearable Healthcare Monitoring System[J]. IEEE Transactions on Dependable and Secure Computing, 2020, 17(5): 942-956.

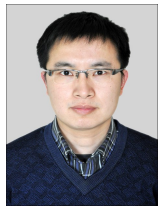
[13] LIU R, WANG X, WANG C. An Efficient Two-Factor Authentication Scheme Based on Negative Databases: Experiments and Extensions[J]. Applied Soft Computing, 2022, 119: 108558.

[14] FAN C I, CHAN Y C, ZHANG Z K. Robust Remote Authentication Scheme with Smart Cards[J]. Computers & Security, 2005, 24(8): 619-628.

[15] RAMASAMY R, MUNIYANDI A P. New Remote Mutual Authentication Scheme Using Smart Cards[J]. Transactions on

Data Privacy, 2009, 2: 141-152.
[16] LEE Y C, HSIEH Y C, LEE P J, et al. Improvement of the ElGamal Based Remote Authentication Scheme Using Smart Cards [J]. Journal of Applied Research and Technology, 2014, 12(6): 1063-1072.
[17] GIRI D, MAITRA T, AMIN R, et al. An Efficient and Robust RSA-Based Remote User Authentication for Telecare Medical Information Systems [J]. Journal of Medical Systems, 2014, 39(1): 145.
[18] KUMARI A, JANGIRALA S, ABBASI M Y, et al. ESEAP: ECC Based Secure and Efficient Mutual Authentication Protocol Using Smart Card [J]. Journal of Information Security and Applications, 2020, 51: 102443.
[19] KUMARI A, ABBASI M Y, ALAM M. A Smartcard-Based Key Agreement Framework for Cloud Computing Using ECC [C]// 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV). Tirunelveli: IEEE, 2021: 43-48.
[20] SHOHAIMAY F, ISMAIL E S. Improved and Provably Secure ECC-Based Two-Factor Remote Authentication Scheme with Session Key Agreement [J]. Mathematics, 2023, 11(1): 5.
[21] XIE Q, WONG D S, WANG G, et al. Provably Secure Dynamic ID-Based Anonymous Two-Factor Authenticated Key Exchange Protocol with Extended Security Model [J]. IEEE Transactions on Information Forensics and Security, 2017, 12(6): 1382-1392.
[22] AMIN R, BISWAS G P. Design and Analysis of Bilinear Pairing Based Mutual Authentication and Key Agreement Protocol Usable in Multi-Server Environment [J]. Wireless Personal Communications, 2015, 84(1): 439.
[23] AMIN R, ISLAM S H, BISWAS G P, et al. A More Secure and Privacy-Aware Anonymous User Authentication Scheme for Distributed Mobile Cloud Computing Environments [J]. Security and Communication Networks, 2016, 9(17): 4650.
[24] SURESHKUMAR V, AMIN R, ANITHA R. An Enhanced Bilinear Pairing Based Authenticated Key Agreement Protocol for Multi-Server Environment [J]. International Journal of Communication Systems, 2017, 30(17): e3358.
[25] SURESHKUMAR V, AMIN R, OBAIDAT M S, et al. An Enhanced Mutual Authentication and Key Establishment Protocol for TMIS Using Chaotic Map [J]. Journal of Information Security and Applications, 2020, 53: 102539.
[26] KUMAR A, OM H. An Enhanced and Provably Secure Authentication Protocol Using Chebyshev Chaotic Maps for Multi-Server Environment [J]. Multimedia Tools and Applications, 2021, 80(9): 14163-14189.
[27] KWON J O, JEONG I R, LEE D H. Three-Round Smart Card-Based Key Exchange Scheme [J]. IEICE Transactions on Communications, 2007, E90-B(11): 3255-3258.
[28] YOON E J, YOO K Y. Enhanced Three-Round Smart Card-Based Key Exchange Protocol [C]// Autonomic and Trusted Computing. Berlin, Heidelberg: Springer, 2008: 507-515.
[29] YANG H, ZHANG Y, ZHOU Y, et al. Provably Secure Three-Party Authenticated Key Agreement Protocol Using Smart Cards [J]. Computer Networks, 2014, 58: 29-38.
[30] KATZ J, LINDELL Y. Introduction to Modern Cryptography [M]. 2nd ed. Boca Raton, US: CRC Press, 2015: 389-398.
[31] CRAMER R, SHOUP V. Design and Analysis of Practical Public-Key Encryption Schemes Secure Against Adaptive Chosen Ciphertext Attack [J]. SIAM Journal on Computing, 2003, 33(1): 167-226.
[32] ZHAO Z, FAN T, PENG T, et al. Key Encapsulation Mechanism from Lattice in Standard Model [J]. Journal of Frontiers of Computer Science and Technology, 2019, 13(4): 629-638.
[33] JARECKI S, LIU X. Fast Secure Computation of Set Intersection [C]// Security and Cryptography for Networks. Berlin, Heidelberg: Springer, 2010: 418-435.
[34] CHEN H, HUANG Z, LAINE K, et al. Labeled PSI from Fully Homomorphic Encryption with Malicious Security [C]// Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18). Toronto: ACM, 2018: 1223-1237.
[35] CONG K, MORENO R C. Labeled PSI From Homomorphic Encryption with Reduced Computation and Communication [C]// Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. Virtual Event: ACM, 2021: 1135-1150.
[36] FREEDMAN M J, ISHAI Y, PINKAS B, et al. Keyword Search and Oblivious Pseudorandom Functions [C]// Theory of Cryptography. Berlin, Heidelberg: Springer, 2005: 303-324.
[37] AMANDA C, DAVI R, DIEGO F A. Faster Unbalanced Private Set Intersection [J]. Journal of Internet Services and Applications, 2018, 9(1): 1-18.

WANG Yi, born in 1994, postgraduate. Her main research interests include password authentication and privacy protection.



HU Xuexian, born in 1982, Ph.D, associate professor, Ph.D supervisor. His main research interests include big data security, applied cryptography and network security.

(责任编辑:喻藜)