文章编号:1671-4229(2024)01-0001-11

一种基于多因素认证与密钥协商的数据密钥管理方案

朱恩强[1], 张 宇[1], 江观华[1], 许宇光[2*]
(1. 广州大学 计算科技研究院, 广东 广州 510006; 2. 西安科技大学 计算机科学与技术学院, 陕西 西安 710054)

摘 要: 隐私数据远程存储技术为用户存储数据带来便捷的同时,也增加了敏感数据在传输过程中遭受拦截攻击的风险。为了提高数据的安全性,需要对上传到远程设备的敏感数据进行加密。因此,高效可靠的密钥管理是确保数据安全的关键。多因素认证是保证数据安全传输的关键技术之一,在安全领域具有广泛的应用,如隐私数据保护、访问权限管理和在线支付等。鉴于此,提出了一种基于多因素认证的密钥存储策略来加强密钥管理:①通过确定的设备身份信息对密钥进行Shamir(2,3)分割;②对设备身份信息进行公钥加密,然后利用用户私有登录口令和生物特征来隐藏密钥的Shamir分割份额和公钥加密的私钥;③对获得的密钥相关信息进行一系列计算处理并分别存储到相应的设备中。理论分析表明,所提方案具有认证灵活,密钥管理高效、可靠以及通信安全等优势。此外,为了进一步说明方案的有效性,进行了BAN逻辑分析和启发式安全分析。分析结果表明,框架能够安全地协商会话密钥并抵抗多种已知攻击。

关键词: 隐私数据保护; 多因素认证; 加密; BAN逻辑
中图分类号: TN918.4 **文献标志码:** A

A data key management scheme based on multifactor authentication and key agreement

ZHU En-qiang[1], *ZHANG Yu*[1], *JIANG Guan-hua*[1], *XU Yu-guang*[2*]
(1. Institute of Computing Science and Technology, Guangzhou University, Guangzhou 510006, China;
2. College of Computer Science & Technology, Xi'an University of Science and Technology, Xi'an 710054, China)

Abstract: While remote storage technology for private data can provide convenience for users, the risk of interception attacks on private data during transmission is also increased. To improve security, sensitive data should be encrypted before uploading to remote devices. So, how to manage the secure key efficiently and reliably is very significant to data security. Multi-factor authentication is one of the key technologies to ensure the security of data transmission, and it has been widely applied in security fields, such as privacy data protection, access rights management, and online payment, etc. To address these problems, a key storage strategy based on multi-factor authentication is proposed to enhance secure key management. First, the key is split using chamir (2,3) based on the indentified device indentity information. Second, it encrypts the identity information of the devices using public key encryption, and concels a share of the secure key derived from Shamir's secret sharing, as well as

收稿日期: 2023-08-02; **修回日期:** 2023-09-20
基金项目: 国家自然科学基金资助项目(61872101);广州市基础研究计划市校(院)联合资助项目(202201020180)
作者简介: 朱恩强(1983—),男,教授,博士. E-mail:zhuenqiang@gzhu.edu.cn
* 通信作者. E-mail:xuyuggmw@xust.edu.cn
引文格式: 朱恩强, 张宇, 江观华, 等. 一种基于多因素认证与密钥协商的数据密钥管理方案[J]. 广州大学学报(自然科学版), 2024,23(1):1-11.

the private key used in public key encryption, through the user's private login password and biometric feature. Finally, all of the above information related to the secure key is processed by a series of computations and then are stored in the designated devices, respectively. Analysis in theory shows that our framework possesses the advantages of flexible authentication, efficient and reliable key management, and secure communication. Moreover, to further illustrate the effectiveness of the approach, experiments on BAN logic analysis and heuristic security analysis were carried out. The experimental results show that the proposed framework can negotiate session keys securely and resist various known attacks.

Key words: privacy data protection; multi-factor authentication; public key encryption; BAN Logic

近年来,随着无线信息通信等互联网技术的快速发展,远程隐私大体量数据备份存储得到了极大普及[1],这为用户提供了灵活、经济且非常便捷的存储服务。然而,远程服务器并不完全可信[2],例如2020年微博上亿用户数据泄漏①。为了保证用户数据文件的机密性,需要在数据文件上传之前对其进行加密,从而保证在用户口令等相关因素被泄漏时攻击者无法通过获取解密密钥来恢复明文[3]。对称加密是一种常用的大体量数据加密方法,用户只需安全管理密钥,即可保证数据安全[4-5]。然而,密钥的高复杂性使得它很难被人们记住,因此,其通常被存储在智能设备或个人笔记本电脑中。但是,这些存储设备可能丢失或被攻击,密钥存在被盗取的风险,从而导致数据泄露或无法恢复[6]。如何管理和保护加密数据的密钥对保障数据安全极其重要。

1 相关工作

2015年,Chang等[7]提出了一种面向服务器的数据保护方案。该方案基于Shamir秘密共享[8-9],主要实现原理是用户选择一份密钥并将其分成3份存储在智能卡设备、笔记本电脑和服务器中。根据Shamir秘密共享的理论安全特性[10],用户应该加强与服务器之间的身份认证。一个安全有效的身份认证方案应具有以下4个特征:①能够保护用户隐私且具备较高的安全性以抵御多种已知攻击;②允许用户选择和更新他们的个人口令;③能够提供用户和服务器之间相互的身份认证;④为后续通信生成安全会话密钥。为满足上述需求,多种认证方法被相继提出,主要包括传统认证方案可信执行环境[11]和结合生物特征的多因素认证技术[12]。与传统方案相比,后者安全协议中安全认证的级别更高、更实用[13]。目前,用户常用的智能设备采用的几乎都是结合生物特征的认证技术,它已经成为用户认证领域的主流。

结合用户名和口令的传统用户认证是由Lamport[14]在1981年提出的,它是一种单因素认证协议[15]。通常用户选择的口令长度较短且不是随机生成的,因此,容易被简单的字典攻击和窥视攻击破译[16]。2017年,Wang等[17]指出许多人倾向于把自己的口令存储在设备上,并在多个账户中重复使用同一个口令[16],这意味着单纯依靠口令的认证是不安全的。

在之后的研究中出现了双因素身份认证[18],这种方案增强了身份认证的安全性,其中一个因素是已知的数据,如口令,另一个因素是物理设备,如智能卡、身份令牌等。尽管双因素认证方案能实现一系列安全目标,但它只依赖于智能卡和口令,容易遭受字典攻击和拒绝服务攻击,因此,系统存在安全隐患[19]。此外,它很容易受到中间人攻击。

由于双因素认证的安全缺陷,结合指纹、声纹或人脸等生物特征也相继被用于身份认证,它提供了潜在的高熵信息,同时不易被窃取或遗忘。基于生物识别的多因素认证协议结合了口令、智能卡和生物识别的优点,提高了数据信息的保障能力[20],这是多因素认证的一种实现方案,已得到广泛研究。

2017年,Liu等[21]设计了一种以用户为中心的多因素认证方案,即私钥由数据所有者保存而非从服务器获得。在该方案中,由用户生成密钥

① https://baijiahao.baidu.com/s?id=1661670005134865024&wfr=spider&for=pc

并对敏感数据进行加密,从而解决了以服务器为中心的身份认证方案中存在的服务器不可信与密钥恢复复杂等问题。

2019 年,Hu 等[22]发现,在 Liu 等的方案中,一旦参与会话方之间传输的敏感信息被攻击者截获,就会遭受一系列的恶意攻击,包括离线口令猜测、用户模拟和服务器模拟等。这些都是出现在用户与服务器之间相互认证过程中的安全问题[23]。此外,Hu 等还发现恶意攻击者可以在口令或者生物特征更新阶段随意更改用户的口令或生物特征。为解决上述问题,Hu 等提出了一种基于用户为中心的增强型数据保护方案,从而优化了 Liu 等的方案。2018 年 Roy 等[24]发现以服务器为中心的多因素认证方案容易遭受服务器冒用和用户冒用攻击,提出了一种基于混沌映射的轻量级三因素远程认证,并验证了所提方案可以抵抗各种已知攻击。

虽然 Hu 等改进了 Liu 等的方案,但是仍存在易遭受侧信道攻击和弱安全密钥管理等安全问题。

基于以用户为中心的设计模型,本文提出了一种以哈希和异或运算为主的轻量级多因素认证密钥管理方案,并通过 BAN 逻辑分析[25],证明了方案可以正确、安全地执行。此外,本文对所提方案进行了启发式安全性分析,即通过静态和动态相结合的方式,分析程序执行过程中可能遭受的安全威胁,并针对可能出现的安全问题进行方案执行过程反编译。

2　准备工作

2.1　模糊提取器

为了实现生物识别技术,本文使用模糊提取器[26]从收集生物特征的设备中提取有效的生物特征。

模糊提取器由两个概率多项式时间算法 Gen 和 Rep 组成。

(1)Gen(B) = (R,P):该算法将生物特征 B 作为输入,输出提取的随机字符串 R 和辅助字符串 P。

(2)Rep(B',P) = R:该算法接受一个新值 B' 和一个字符串 P 作为输入,如果新输入的生物特征 B' 与先前的生物特征模板 B 相对类似 $dis(B, B') < t$(t 是可接受的容错范围值),则模糊提取器从输入的生物特征中返回随机字符串 R。

(3)模糊提取器在本方案中的使用:在模糊提取器处理过的生物特征的输出字符的任意位置插入一个口令,通过安全的散列函数进行处理,从而得到需要的一个生物特征的输出,即使恶意用户获取了用户的指纹或者人脸等信息,也无法猜测或者更新方案中使用的特征值。从模糊提取器中获取到生物特征 B 后,随即将生物特征与一个口令 PW_0 通过一个散列函数生成 $Bio = h(B \| PW_0)$ 作为一个生物特征输入。

2.2　系统模型

所提模型主要包含以下 5 个部分(图 1):

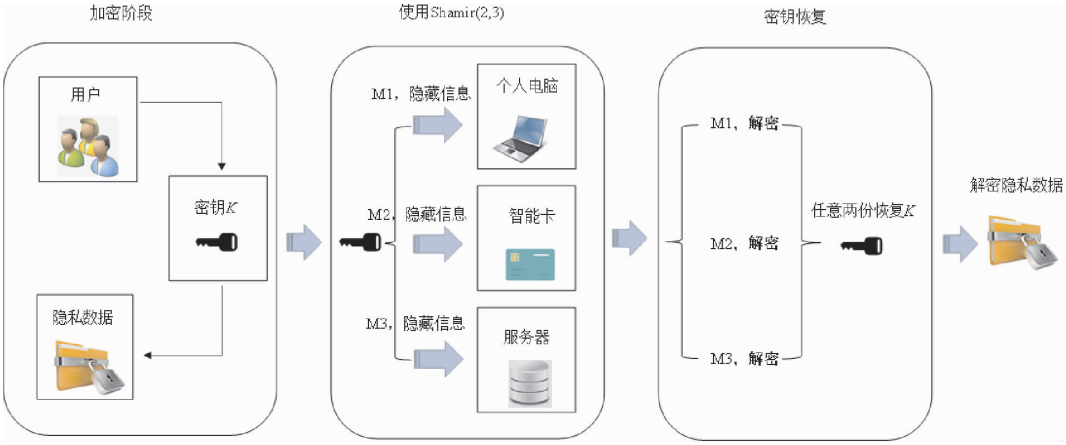

图 1　系统模型

Fig. 1　System model

(1)加密过程:用户选择一个对称密钥 K,用于加密备份的敏感数据。

(2)隐藏用户个人信息:用于增强数据的保密性。

(3)生成共享份额:用户使用 Shamir(2,3)门限秘密共享方案将密钥分成3个部分后,分别分发给服务器、智能卡和电脑存储,三者共享一份密钥 K。

(4)密钥 K 的重建:用户通过服务器、智能卡和个人电脑三者存储的份额进行身份验证,然后使用拉格朗日差值公式恢复密钥 K。

(5)解密备份的敏感数据:用户使用重建的密钥,通过远程服务器存储的备份敏感数据进行解密。

2.3 恶意用户的攻击能力

(1)恶意用户可以通过开放通信信道篡改、截获、删除或插入恶意信息。

(2)恶意用户可以通过某种手段盗用用户的个人设备。

(3)当用户的个人电脑或者智能卡被恶意用户窃取时,攻击者可以通过窥视用户输入口令知晓用户的口令,或者通过侧信道攻击获取用户的生物特征,但是不能两者兼而有之,同时,本文认为恶意用户不能同时得到用户的个人电脑和用户的智能卡。

(4)如果恶意用户获取了用户的口令或者生物特征,那么恶意用户无法得到用户的个人电脑和智能卡。

2.4 安全目标

(1)攻击者即使通过某种手段获取了用户的口令、智能卡或者个人电脑,也无法获取用户的生物特征。

(2)即使攻击者获取了用户的个人电脑或智能卡,也无法获取智能卡或者电脑中存储的有效信息。

(3)即使攻击者获取了用户的生物特征和用户的电脑,也无法获取用户的口令。

(4)即使攻击者获取了用户的生物特征和用户的智能卡,也无法获取用户的口令和智能卡中的有效信息。

3 具体方案

本文提出的基于多因素认证的隐私数据保护密钥加强框架包含4个部分:注册过程、身份认证过程、恢复过程和更新过程,设计方案所用符号见表1。

表1 符号说明

Table 1 Symbol description

符号	符号说明
PW	用户密码
Bio	经过处理的用户生物特征
U	用户
S	服务器
ID_{usr}	用户身份信息
ID_{sc}	智能卡身份信息
ID_{ser}	服务器身份信息
$E_{SK}()/D_{SK}()$	使用 SK 加解密
K	对称加密密钥
γ	RSA 加密的公钥
λ	RSA 加密的私钥
$E_{\gamma}()/D_{\lambda}()$	使用 γ 加密,λ 解密
$\oplus$	异或操作
$\|\|$	连接操作
$h(\cdot)$	安全的散列函数
US	恶意用户
ID_{usr}^{new}	更新的用户身份信息
PW^{new}	更新的用户密码
Bio^{new}	经过处理的更新的用户生物特征

3.1 注册阶段

通过注册阶段,用户 U 的个人电脑和智能卡将会与服务器建立连接,需要执行以下步骤:

(1)用户首先选择自己的身份信息 ID_{usr}、ID_{sc}、个人口令 PW,获取服务器 ID_{ser} 和处理过的生物特征信息 Bio。

(2)用户选择一个对称密钥 K 来加密备份数据,使用多项式 $f(x)=ax+K$(其中,a 是随机数)生成3个共享份额 $g_{sc}=f(ID_{sc})$、$g_{usr}=f(ID_{usr})$ 和 $g_{ser}=f(ID_{ser})$,随后用户销毁用户私钥 K。此步骤确保了恢复密钥阶段的安全性。

(3)使用基于因子分解(RSA)的非对称加密对身份信息参数进行加密。选择一个公钥 γ,对应的私钥为 λ,将3份身份信息进行加密得到:$C_{sc}=E_{\gamma}^{*}(ID_{sc})$,$C_{ser}=E_{\gamma}^{*}(ID_{ser})$,$C_{usr}=E_{\gamma}^{*}(ID_{usr})$。

(4)用户生成 $S_{ser}=g_{ser}\oplus h(C_{ser}\|PW\|Bio)$,$S_{sc}=g_{sc}\oplus h(C_{sc}\|PW\|Bio)$,$S_{usr}=g_{usr}\oplus h(C_{usr}\|PW\|Bio)$。

(5)用户计算身份认证消息 $V=h(C_{sc}\|PW\|C_{usr}\|N_1)$($N_1$ 是随机值),$C_1=h(PW\|Bio\|C_{usr})$,$C_2=h(PW\|Bio\|C_{ser})$,$H=h(PW\|Bio\|C_{usr})\oplus N_1$,$C_3=h(PW\|Bio\|C_{ser})\oplus\lambda$。

(6)将 $m_1=\{S_{ser},S_{sc},C_{ser},C_1,C_2,C_3\}$ 转发给服务器,此消息用来后续对服务器进行身份认证。

(7)服务器在接收到消息 m_1 之后,选择随机值 X_1 和 X_2 生成 $Y_1=h(C_{ser}\|X_1)$,$Y_2=h(C_{ser}\|X_2)$,$Z_1=Y_1\oplus C_1$ 和 $Z_2=Y_2\oplus C_2$ 并存储值 $m_2=(S_{sc},S_{ser},C_{ser},C_3,X_1,X_2)$。

(8)服务器向用户发送消息 $m_3=(Z_1,Z_2)$。

(9)用户在笔记本电脑中存储 $m_4=\{S_{usr},C_{ser},C_{usr},Z_1,Z_2,C_3,\gamma\}$。

(10)用户将 $m_5=\{S_{sc},C_{sc},H,V,Z_1,Z_2,\gamma\}$ 发送到智能卡,智能卡存储 m_5。

3.2　认证过程

如果用户希望恢复私钥 K 以解密加密的敏感数据,需使用服务器或智能卡执行身份认证过程,这个过程中可能出现以下 3 种情况。

3.2.1　同时持有智能卡和笔记本电脑

(1)用户与服务器建立会话密钥并且相互认证。

1)用户将 PW 和 Bio 输入到个人笔记本电脑中。读取存储在电脑中的份额之后,个人电脑计算 $Y_1=Z_1\oplus C_1$,$Y_2=Z_2\oplus C_2$,$m_6=Y_1\oplus T_u$,$VC_1=h(Y_2\|T_u)$,其中,T_u 是随机的临时值。

2)用户向服务器发送请求消息(VC_1,m_6)。

3)在接收到请求消息(VC_1,m_6)之后,服务器计算 $m_7=Y_1\oplus m_6$,$Y_1=h(C_{ser}\|X_1)$,$Y_2=h(C_{ser}\|X_2)$,之后服务器检查 $h(Y_2\|m_7)=VC_1$ 是否成立。如果判断条件成立,服务器认证用户通过,服务器计算 $SK_s=h(VC_1\|Y_1\oplus m_6\|T_s)$,$m_8=T_s\oplus Y_1\oplus Y_2$,$VC_2=h(SK_s\|Y_2\|T_s)$,其中 T_s 是随机的临时值;否则,程序停止。服务器向用户发送消息(m_8,VC_2)。

4)用户计算 $T_s=m_8\oplus Y_1\oplus Y_2$,$SK_u=h(VC_1\|T_u\|m_8\oplus Y_1\oplus Y_2)$之后,检查 $h(SK_u\|Y_2\|m_8\|\oplus Y_1\oplus Y_2)=VC_2$ 是否成立。如果判断条件成立,用户认证服务器并共享会话密钥 $SK=SK_u$;否则,程序将停止。

5)用户计算 $M=h(SK_u\|Y_1\|Y_2\|VC_1\|VC_2\|T_u\|\oplus Y_1\oplus Y_2)$。用户向服务器发送确认消息 $m_9=E_{SK}(M)$。

6)服务器计算 $D_{SK}(M)$,如果 $h(SK_S\|Y_1\|Y_2\|VC_1\|m_6\|Y_1\|T_s)=M$,确认共享会话密钥为 $SK=SK_u=SK_s$。

7)服务器计算 $m_9=E_{SK}(M)$,并向用户发送 $m_{10}=E_{SK}(S_{ser})$,用户计算 $S_{ser}=D_{SK}(m_{10})$,计算真正共享 $g_{ser}=S_{ser}\oplus h(C_{ser}\|PW\|Bio)$。

(2)用户与智能卡之间相互认证以及恢复密钥 K 过程。

1)用户与智能卡之间建立安全物理连接。

2)用户将 $C_{usr}=E_\gamma^*(ID_{usr})$,$PW$ 和 Bio 输入到自己的智能卡中,然后智能卡计算 $N_1'=h(PW\|Bio\|C_{usr})\oplus H$,$V'=h(C_{sc}\|PW\|C_{usr}\|N_1')$。

3)智能卡检查 V' 与 V 是否相等,如果相等,则向用户发送消息 S_{sc},C_{sc}。用户计算真正共享

$$g_{sc}=S_{sc}\oplus h(C_{sc}\|PW\|Bio),$$

$$g_{usr}=S_{usr}\oplus h(C_{usr}\|PW\|Bio)。$$

4)用户通过与远程设备建立连接后恢复三元组(ID_{usr},g_{usr}),(ID_{ser},g_{ser}),(ID_{sc},g_{sc})3 份中的两份,便可以轻松恢复备份数据的密钥 K。例如假设已知(ID_{usr},g_{usr})和(ID_{sc},g_{sc}),那么解密的密钥为

$$K=\frac{g_{sc}\cdot ID_{usr}}{ID_{usr}-ID_{sc}}-\frac{g_{usr}\cdot ID_{sc}}{ID_{usr}-ID_{sc}}。$$

3.2.2　智能卡丢失和更新

如果用户丢失了智能卡,那么用户需要提供其个人电脑中存储的有效口令 PW、生物特征 Bio 和存储在个人电脑中的有效份额,并通过服务器的身份认证。该过程预期实现两个目标:①用户与服务器之间的相互认证,目的是让用户获得存储在服务器中的 S_{ser},此时可以轻松恢复密钥 K;②密钥恢复以后,用户可以选择更新智能卡。更新智能卡的具体步骤如下:

(1)用户输入个人口令 PW 与生物特征 Bio。

(2)用户计算恢复密钥 $\lambda=C_3\oplus h(PW\|Bio\|S_{ser})$,$ID_{usr}=D_\lambda^*(C_{usr})$,$ID_{ser}=D_\lambda^*(C_{ser})$。

(3)恢复密钥 K。

(4)用户生成智能卡 ID'_{sc}的新标识和一个新的多项式 $f_2(x)=a_2x+K$(a_2 是一个随机值)。笔记本电脑计算 3 个真实共享 $g'_{sc}=f_2(ID'_{sc})$,$g'_{ser}=f_2(ID'_{ser})$,$g'_{usr}=f_2(ID'_{usr})$,然后计算 $C'_{sc}=E_\gamma(ID'_{sc})$,$C'_{ser}=E_\gamma(ID'_{ser})$,$C'_{usr}=E_\gamma(ID'_{usr})$,$S'_{sc}=g'_{sc}\oplus h(C'_{sc}\|PW\|Bio)$,$S'_{usr}=g'_{usr}\oplus h(C'_{usr}\|PW\|Bio)$,并选择一个随机值 N'_2,计算得到

$$H'=h(C'_{usr}\|Bio\|PW)\oplus N'_2,$$

$$V'=h(C'_{sc}\|PW\|C'_{usr}\|N'_2),$$

$$C'_1=h(PW\|Bio\|C'_{usr}),$$

$$C'_2=h(PW\|Bio\|C'_{ser}),$$

$$Y'_1=h(C'_{ser}\|X_1),Y'_2=h(C'_{ser}\|X_2),$$

又因为 $Z_1=Y_1\oplus C_1$,$Z_2=Y_2\oplus C_2$,所以,计算可

得 $Z'_1 = Y_1 \oplus C'_1$, $Z'_2 = Y_2 \oplus C'_2, C'_3 = \lambda \oplus h(PW \| Bio \| S'_{ser})$ 此过程 ID_{ser}, ID_{usr} 的值没有变化,C'_{ser}, C'_{usr} 也与原值保持一致。

(5)用户更新存储的份额,同时保存原来存储在个人电脑中的参数 $S_r = C_{ser}$,然后向服务器发送消息请求 $m_{11} = E_{SK}(S'_{ser}, S_r, S'_{sc})$。

(6)服务器计算 $D_{SK}(m_{11}) = (S'_{ser}, S_r, S'_{sc})$,计算 $C_{ser} = S_r$ 是否成立,如果成立,则使用 $S'_{ser}, C'_{ser}, S'_{sc}$ 替换服务器存储的 S_{ser}, C_{ser}, S_{sc}。

(7)用户向智能卡发送 $(C'_{sc}, S'_{sc}, H', V', Z'_1, Z'_2, \gamma)$,智能卡保存此消息。

3.2.3 丢失笔记本电脑和更新

用户即使丢失个人电脑也可以轻松通过智能卡和新的个人电脑与远程服务器建立新的连接,目的是为了获取存储在服务器端的 C_{ser} 和 C_3,用户在恢复密钥 K 以后,可以选择更新重建个人电脑,具体步骤如下:

(1)用户输入个人口令 PW 与生物特征 Bio 和 ID_{usr}。

(2)用户计算恢复密钥 $C_{usr} = E^*_{\gamma}(ID_{usr})$, $ID_{ser} = D_{\lambda}{}^*(C_{ser})$。

(3)向智能卡发送认证请求,智能卡计算:

$$N'_3 = h(PW \| Bio \| C_{usr}) \oplus H,$$

$$V' = h(C_{sc} \| PW \| C_{usr} \| N'_3)。$$

(4)智能卡检查 V' 与 V 是否相等,如果相等,则向用户发送消息 S_{sc}, C_{sc}。

(5)向服务器发送请求 $m_{12} = E_{SK}(S_{sc})$,服务器计算 $D_{SK}(m_{12})$ 与服务器存储的 S_{sc} 是否相等,相等则向用户发送消息 $m_{13} = E_{SK}(S_{ser}, C_{ser}, C_3)$,用户拿到消息 m_{13} 解密后恢复密钥 K。

(6)用户生成新电脑中的身份 ID'_{usr} 的新标识和一个新的多项式 $f_3(x) = a_3 x + K$,其中,a_3 是一个随机值。笔记本电脑计算 3 个真实共享 $g'_{sc} = f_3(ID_{sc})$, $g'_{ser} = f_3(ID_{ser})$, $g'_{usr} = f_3(ID_{usr})$,然后计算 $C'_{sc} = E_{\gamma}(ID_{sc})$, $C'_{ser} = E_{\gamma}(ID_{ser})$, $C'_{usr} = E_{\gamma}(ID'_{usr})$,随即计算得到

$$S'_{ser} = g'_{ser} \oplus h(C'_{ser} \| PW \| Bio),$$

$$S'_{usr} = g'_{usr} \oplus h(C'_{usr} \| PW \| Bio),$$

$$S'_{sc} = g'_{sc} \oplus h(C'_{sc} \| PW \| Bio),$$

之后选择一个随机值 $N'_4, H' = h(C'_{usr} \| Bio \| PW) \oplus N'_4, V' = h(C'_{sc} \| PW \| C'_{usr} \| N'_4), C'_1 = h(PW \| Bio \| C'_{usr}), C'_2 = h(PW \| Bio \| C'_{ser})$,又因为 $Z_1 = Y_1 \oplus C_1, Z_2 = Y_2 \oplus C_2$,计算得到 $Z'_1 = Y_1 \oplus C'_1$, $Z'_2 = Y_2 \oplus C'_2, C'_3 = \lambda \oplus h(PW \| Bio \| S'_{ser})$。此过程 ID_{ser}, ID_{sc} 的值没有变化 C'_{ser}, C'_{sc} 也与原值保持一致。

(7)用户保存更新后的值。智能卡更新信息并保存更新后的值。

(8)用户用 $S_r = S_{sc}$ 记录智能卡的信息并向服务器发送消息请求 $m_{14} = E_{SK}(S_r, S'_{ser}, S'_{sc}, C'_3)$。

(9)服务器接收信息之后检查 $D_{SK}(m_{14})$,检查 S_r 是否与服务器保存的 S_{sc} 相等,相等则用 S'_{ser}, S'_{sc}, C'_3 替换服务器保存的值 S_{ser}, S_{sc}, C_3。

3.3 更新阶段

3.3.1 更新口令

用户需要定期更新口令以避免恶意用户猜测攻击或者通过侧道攻击获取用户口令。此过程需要将原来的口令 PW 更新为 PW'。

(1)用户将自己掌握的智能卡连接至个人电脑。

(2)用户输入其自身的 $C_{usr} = E^*_{\gamma}(ID_{usr})$,旧的 PW 以及生物特征 Bio。

(3)智能卡读取到用户输入的信息之后计算检查 V'_1 与 V 是否相等,其中 $V'_1 = h(C_{sc} \| PW \| C_{usr} \| N'_1), N'_1 = h(PW \| Bio \| C_{usr}) \oplus H$,如果 V'_1 与 V 相等,那么计算过程继续,否则,停止程序。

(4)用户选择一个新的随机值 N'_5 和新的 PW^{new},然后计算下面的值:

$$H^{new} = h(PW^{new} \| Bio \| C_{usr}) \oplus N'_2,$$

$$V^{new} = h(C_{sc} \| PW^{new} \| C_{usr} \| N'_2),$$

$$g_{sc} = S_{sc} \oplus h(C_{sc} \| PW \| Bio),$$

$$g_{usr} = S_{usr} \oplus h(C_{usr} \| PW \| Bio),$$

$$S^{new}_{sc} = g_{sc} \oplus h(C_{sc} \| PW^{new} \| Bio),$$

$$S^{new}_{usr} = g_{usr} \oplus h(C_{usr} \| PW^{new} \| Bio)。$$

因为 $C_1 = h(PW \| Bio \| C_{usr}), C_2 = h(PW \| Bio \| C_{ser})$ $C^{new}_1 = h(PW^{new} \| Bio \| C_{usr}), C^{new}_2 = h(PW^{new} \| Bio \| C_{ser})$,所以 $Z^{new}_1 = C_1 \oplus C^{new}_1 \oplus Z_1, Z^{new}_2 = C_2 \oplus C^{new}_2 \oplus Z_2$。

(5)认证服务器,用户向服务器发出请求,服务器读取到用户注册登录的信息之后,用户向服务器发送 $E_{SK}(PW^{new}, Bio, S^{new}_{sc})$,服务器更新的值包括 $D_{SK}E_{SK}(PW^{new}, Bio, S^{new}_{sc}), g_{ser} = S_{ser} \oplus h(C_{ser} \| PW \| Bio), S^{new}_{ser} = g_{ser} \oplus h(C_{ser} \| PW^{new} \| Bio), C_3{}^{new} = \lambda \oplus h(PW^{new} \| Bio \| S^{new}_{ser})$ 计算的值均需要更新。服务器向用户发送消息 $m_{15} = E_{SK}(C_3{}^{new})$,用户收到消息后,计算 $D_{SK}(m_{20})$,用户和智能卡更新存储的值。

3.3.2　更新生物特征

生物特征的更新需要用到模糊提取器和注册生物特征时提供的口令 PW_0,用户执行模糊提取器获得新的生物特征 B^{new}后重新输入一个保障生物特征安全的口令 PW'_0,计算得到 $Bio^{new}=h(B^{new}\|PW'_0)$。

(1)用户将智能卡与个人电脑连接。用户输入其自身的 $C_{usr}=E^*_\gamma(ID_{usr})$、$PW$ 以及旧的生物识别的特征 Bio。

(2)智能卡读取用户输入的信息检查 V'_1与 V 是否相等,其中, $V'=h(C_{sc}\|PW\|C_{usr}\|N_1)$, $N_1=h(PW\|Bio\|C_{usr})\oplus H$, $C_xc=E^*_\gamma(ID_{sc})$,如果 V' 与 V 相等,那么,计算过程继续,否则停止程序。

(3)用户选择一个新的随机值 N'_6和新的 Bio^{new}。然后计算下面的值:

$$H^{new}=h(PW\|Bio^{new}\|C_{usr})\oplus N'_6,$$
$$g_{sc}=S_{sc}\oplus h(C_{sc}\|PW\|Bio),$$
$$g_{usr}=S_{usr}\oplus h(C_{usr}\|PW\|Bio),$$
$$S_{sc}^{new}=g_{sc}\oplus h(C_{sc}\|PW\|Bio^{new}),$$
$$S_{usr}^{new}=g_{usr}\oplus h(C_{usr}\|PW\|Bio^{new}),$$

又因为 $C_1=h(PW\|Bio\|C_{usr})$, $C_2=h(PW\|Bio\|C_{ser})$计算可得 $C_1^{new}=h(PW\|Bio^{new}\|C_{usr})$, $C_2^{new}=h(PW\|Bio^{new}\|C_{ser})$, $Z_1^{new}=C_1\oplus C_1^{new}\oplus Z_1$, $Z_2^{new}=C_2\oplus C_2^{new}$。

(4)用户向服务器发出发消息请求服务器读取到用户注册登录的信息之后,用户向服务器发送 $m_{16}=E_{SK}(Bio^{new},S_{sc}^{new})$,服务器接收到消息之后计算下面的值。

$D_{SK}(m_{16})$, $g_{ser}=S_{ser}\oplus h(C_{ser}\|PW\|Bio)$, $S_{ser}^{new}=g_{ser}\oplus h(C_{ser}\|PW\|Bio^{new})$, $C_3^{new}=\lambda\oplus h(PW\|Bio^{new}\|S_{ser}^{new})$服务器更新所存储的值($S_{sc}^{new}$, S_{ser}^{new}, C_{ser}, C_3^{new}, X_1, X_2)。

(5)服务器向用户发送消息 $m_{17}=E_{SK}(C_3^{new})$,用户收到消息后计算 $D_{SK}(m_{17})$,用户更新存储的值,智能卡更新存储的值。

3.3.3　更新智能卡与个人电脑中的信息

与 3.2 节中智能卡和个人电脑丢失情况的重建过程相同,这里不再重复叙述。

4　相互认证安全性分析

BAN 逻辑分析[26]被广泛应用于用户与服务器之间的相互认证分析,本小节采用 BAN 逻辑对方案相互认证的安全性进行分析,BAN 逻辑的符号和相关规则如表 2 所示。

表 2　BAN 逻辑的符号和相关规则

Table 2　Notation and rules for BAN logic

符号	说明
P,Q	通信双方
X,Y	参数
F	密钥
$\#(X)$	X 是新鲜值
$P\lhd X$	P 收到包含 X 的消息
$P\|\sim X$	P 发送包含 X 的消息
$P\|\equiv X$	P 相信 X
$P\overset{Y}{\Leftrightarrow}Q$	P 和 Q 共享秘密 Y
$P\overset{F}{\leftrightarrow}Q$	P 和 Q 共享密钥 F
$<X>_Y$	X 包含秘密 Y
$\{X\}_F$	对 X 使用 F 加密
$P\Rightarrow X$	P 对 X 有正确与否的判决权
$\dfrac{P\|\equiv P\overset{F}{\leftrightarrow}Q,P\lhd\{X\}_F}{P\|\equiv Q\|\sim X}$或$\dfrac{P\|\equiv P\overset{Y}{\Leftrightarrow}Q,P\lhd<X>_Y}{P\|\equiv Q\|\sim X}$	消息意义规则
$\dfrac{P\|\equiv\#(X),P\|\equiv Q\|\sim X}{P\|\equiv Q\|\equiv X}$	随机数验证规则
$\dfrac{P\|\equiv Q\Rightarrow X,P\|\equiv Q\|\equiv X}{P\|\equiv X}$	仲裁规则
$\dfrac{P\|\equiv(X,Y)}{P\|\equiv X}$	附加规则

根据 BAN 逻辑程序分析的要求,如果所提方案能实现下列目标,那么,所提的本方案正确实现了相互认证与会话密钥的协商。

G1: $S|\equiv U|\equiv(U\overset{SK}{\longleftrightarrow}S)$, G2: $S|\equiv(U\overset{SK}{\longleftrightarrow}S)$

G3: $U|\equiv S|\equiv(U\overset{SK}{\longleftrightarrow}S)$, G4: $U|\equiv(U\overset{SK}{\longleftrightarrow}S)$

用户与服务器通常情况下认证和会话密钥建立的信息传输形式如下:

消息 M1: $U\rightarrow S<VC_1, m_6>$,

消息 M2: $S\rightarrow U<m_8, VC_2>$,

消息 M3: $U\rightarrow S<m_9>$。

理想化的消息形式:

消息 M1: $U\rightarrow S<VC_1,m_6>_{T_u}$。

消息 M2:$S\rightarrow U<m_8,VC_2>_{T_s}$

消息 M3:$U\rightarrow S<m_9>_{T_u}$

方案的初始化假设:

A1:$S|\equiv U\overset{T_u}{\Leftrightarrow}S$

A2:$S|\equiv\#(T_u)$

A3:$S|\equiv U\Rightarrow<VC_1,\ m_6>$

A4:$U|\equiv U\overset{T_s}{\Leftrightarrow}S$

A5:$U|\equiv\#(T_s)$

A6:$U|\equiv S\Rightarrow<m_8,VC_2>$

A7:$U|\equiv S\Rightarrow(U\overset{SK_s}{\longleftrightarrow}S)$

A8:$S|\equiv U\overset{SK_u}{\longleftrightarrow}S$

A9:$S|\equiv U\Rightarrow<m_9>$

A10:$S|\equiv U\Rightarrow(U\overset{SK_u}{\longleftrightarrow}S)$

S1:从消息 M1,得到

$S\lhd<VC_1,m_6>_{T_u}$

S2:根据 A1,S1,应用消息意义规则,得到

$S|\equiv U|\sim<VC_1,m_6>$

S3:根据 A2,S2,应用随机数认证规则,得到

$S|\equiv U|\equiv<VC_1,m_6>$

S4:根据 A3,S3,应用仲裁规则,得到

$S|\equiv<VC_1,m_6>$

S5:根据 S4,应用附加规则,得到

$S|\equiv VC_1,S|\equiv m_6$

S6:从消息 M2,得到

$U\lhd<m_8,VC_2>_{T_s}$

S7:根据 A4,S6,应用消息意义规则,得到

$U|\equiv S|\sim<m_8,VC_2>$

S8:根据 A5,S7,应用随机数认证规则,得到

$U|\equiv S|\equiv<m_8,VC_2>$

S9:根据 A6,S8,应用仲裁规则,得到

$U|\equiv<m_8,VC_2>$

S10:根据 S5,S9,以及 $SK_s=h(VC_1\|Y_1\oplus m_6\|T_s)$,得到

$U|\equiv S|\equiv(U\overset{SK_s}{\longleftrightarrow}S)$

S11:根据 A7,S10,应用仲裁规则,得到

$U|\equiv(U\overset{SK_s}{\longleftrightarrow}S)$

S12:根据 S9,应用附加规则,得到

$U|\equiv VC_2,S|\equiv m_8$

S13:从消息 M3,得到

$S\lhd<m_9>_{T_u}$

S14:根据 A8,S13,应用消息意义规则,得到

$S|\equiv U|\sim<m_9>$

S15:根据 A2,S14,应用随机数认证规则,得到

$S|\equiv U|\equiv<m_9>$

S16:根据 A9,S15,应用仲裁规则,得到

$S|\equiv<m_9>$

S17:根据 S5,S12,S16,以及 $SK_u=h(VC_1\|T_u\|m_8\oplus Y_1\oplus Y_2)$,得到

$S|\equiv U|\equiv(U\overset{SK_u}{\longleftrightarrow}S)$

S18:根据 A10,S17,应用仲裁规则,得到

$S|\equiv(U\overset{SK_u}{\longleftrightarrow}S)$

S19:根据 S16,得到

$SK=SK_u=SK_s$

S20:根据 S10,S11,S17,S18 和 S19,得到

$U|\equiv S|\equiv(U\overset{SK}{\longleftrightarrow}S),U|\equiv(U\overset{SK}{\longleftrightarrow}S)$

$S|\equiv U|\equiv(U\overset{SK}{\longleftrightarrow}S),S|\equiv(U\overset{SK}{\longleftrightarrow}S)$

5 启发式安全分析

本节采用启发式安全性分析的方式分析一些重要的安全目标。

5.1 抵抗离线口令攻击

即使恶意攻击者 *US* 在开放的信道中获取了存储在智能卡中的 $m_5=(S_{sc},C_{sc},H,V,Z_1,Z_2,\gamma)$ 或者获取用户向服务器发送的请求(VC_1,m_6)信息,攻击者也不能对其口令进行离线猜测,恶意用户可通过包含在智能卡中的 H 和 V,以及在开放信道中的请求信息(VC_1,m_6)猜测用户口令,然而 $V=h(C_{sc}\|PW\|C_{usr}\|N_1)$(其中,$N_1$ 是随机值),$H=h(PW\|Bio\|C_{usr})\oplus N_1$,$m_6=Y_1\oplus T_u$ 和 $VC_1=h(Y_2\|T_u)$,其中,T_u 是随机的临时值,这些值都不能让恶意用户以离线猜测的方式获得口令。

假设 *US* 获取了智能卡中存储的份额,但是无法得到 C_{usr},N_1,PW,Bio 这几个值,那么猜测口令不可能成功。如果 *US* 通过猜测计算得到 $m_7=Y_1\oplus m_6$,$Y_1=h(C_{ser}\|X_1)$和$Y_2=h(C_{ser}\|X_2)$,之后,服务器将检查 $h(Y_2\|m_7)=VC_1$ 是否正确,但是 *US* 无法得到 Y_1 和 Y_2 的值,因此,以此方式不可能通过认证服务器猜测而得到口令。还有一种情

况,在用户注册阶段向服务器发送消息时,消息 m_1 被截获,*US* 通过开放信道获取了智能卡的份额,但即使是这样也无法猜测口令,因为这需要用户处理过后的生物特征,在本方案中,不可能被 *US* 得到。以上这两种情况,恶意用户均不可能通过猜测攻击得到用户的口令。

5.2　抵抗在线口令猜测攻击

恶意用户 *US* 即使获取了 $m_5=(S_{sc},C_{sc},H,V,Z_1,Z_2,\gamma)$值,也无法获取用户口令。智能卡只有 3 次由远程用户身份认证方案提供的有限请求机会,此外,恶意用户 *US* 无法得到用户经过处理后的生物特征,当恶意用户 *US* 3 次或 3 次以上输入错误的身份认证信息时,智能卡将发现恶意用户 *US* 的非法行为。因此,本文提出的方案可以抵抗在线口令猜测方式。

5.3　抵抗生物特征更新和生物特征猜测以及侧道攻击

从模糊提取器中获取到生物特征 B 后,随即将生物特征与一个口令 PW_0 通过一个散列函数生成 $Bio=h(B\|PW_0)$,在本方案中,*Bio* 作为一个生物特征的输入。恶意用户即使通过侧道攻击[27]得到了真正的用户生物特征,也无法得到方案所需的经过处理后的生物特征 *Bio*,恶意用户同样无法通过猜测得到方案中使用的处理过后的生物特征 *Bio*。

5.4　抵抗用户和服务器模拟攻击

恶意用户 *US* 不可能模拟用户或者服务器进行攻击。要模拟合法用户,*US* 必须持有 $Y_1=Z_1+C_1$ 和 $Y_2=Z_2+C_2$,它们是服务器身份认证的重要消息。假设 *US* 拥有 $C_1=h(C_{usr}\|PW\|Bio)$,$C_2=h(C_{ser}\|PW\|Bio)$,即截获了用户注册阶段向服务器的消息也无法计算 Y_1 和 Y_2,因为它未持有任何有关用户口令或生物特征的信息。因此,攻击者无法通过计算伪造的请求消息 VC_1 和 m_6 来通过服务器进行身份认证。要模拟合法服务器,就要计算 m_8 和 VC_2 以通过用户的认证,其中,$m_8=T_u\oplus Y_1\oplus Y_2$,$VC_2=h(SK_s\|Y_2\|T_s)$,$h(VC_1\|T_u\|m_8\oplus Y_1\oplus Y_2)$,其中,$T_u$ 是恶意用户生成的随机值,T_s 是服务器生成的随机值。*US* 还要获得有效值 Y_1,Y_2 和 T_u,但是,无法获得这些值。*US* 即使可以截获开放通道的消息,也无法获取有效的信息,并通过用户和服务器模拟方式模拟服务器和用户。

基于以上说明,本方案具备强安全密钥的管理能力和抵抗多种已知攻击的能力。

6　性能比较

本节主要介绍功能性和效率分析两方面。本文中的安全性分析基于第 2.3 节和第 2.4 节所示的假设前提下进行,通过 BAN 逻辑[26]和启发式安全性分析对所提方案进行了正确性和安全证明,结果表明,所提方案是正确和安全的,与相关方案的功能比较见表 3。

表 3　功能比较表

Table 3　Functional comparison

攻击类别	Hu 等[22]	Liu 等[21]	本方案
智能卡盗用攻击	√	×	√
个人电脑盗用攻击	√	×	√
侧信道攻击[27]	×	×	√
离线猜测攻击	√	√	√
在线猜测攻击	√	√	√
用户模拟攻击	√	×	√
服务器模拟攻击	√	×	√
更新口令攻击	√	×	√
更新生物特征攻击	√	×	√
安全的多因素认证	√	×	√
强安全密钥	×	×	√

本方案与文献[21-22]在注册、认证、恢复和更新阶段的计算效率对比见表 4,结果由各方案各阶段实现流程而得。T_H、T_P 分别表示散列函数运算、多项式计算。由于异或运算和连接运算的时间复杂度可以忽略不计,因此,不予考虑,由表 4 可知,注册阶段、认证阶段及计算量均小于文献[21-22],恢复阶段计算量小于文献[18],更新阶段计算量均大于文献[21-22]。

表 4　效率分析比较

Table 4　Comparison of efficiency analysis

文献	注册阶段	认证阶段	恢复阶段	更新阶段
文献[18]	$5T_H+8T_P$	$9T_H+4T_P$	$3T_H+3T_P$	$5T_H+4T_P$
文献[22]	$3T_H+8T_P$	$9T_H+4T_P$	$3T_H+6T_P$	$3T_P+4T_P$
本方案	$3T_H+5T_P$	$6T_H+5T_P$	$3T_H+5T_P$	$5T_H+6T_P$

仿真时间如表5所示,其中,多项式生成、拉格朗日插值、哈希函数SHA256和AES是在Window 10 Professional上借助Crypto++ Library[28]进行评估的,该Windows 10 Professional采用运行在3.0 GHz、RAM 16G PC上的AMD Ryzen 5 4600H CPU。每个结果都是执行相同程序1 000次后的平均值。此外,智能卡中的模拟时间参见文献[29]。

表5 性能仿真时间

Table 5 Performance simulation time ms

实体	T_p	T_h
3.0 GHz PC	0.049	0.031
36MHzHiPerSmartTM	1.2	0.172

7 结束语

设计多因素认证来保护用户数据的安全是近年来研究的热点,本文回顾近年来多因素认证的发展历程,提出了基于多因素认证的隐私数据密钥加强方案,在保证数据安全的同时兼顾了使用的便捷性。在方案中,远程存储设备无法主动获取用户的敏感数据,用户存储的敏感数据经过加密后存储在远程设备中,要使用这些数据,用户需要通过多因素认证身份认证后才能从远程设备获取加密数据,之后在本地进行解密获取明文。因此,在方案中加密数据的密钥保存非常重要,本文就如何安全地保存密钥做了详细说明。

参考文献:

[1] Venkatesh A, Eastaff M S. A study of data storage security issues in cloud computing[J]. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2018, 3(1): 1741-1745.

[2] Yan H, Li X, Wang Y, et al. Centralized duplicate removal video storage system with privacy preservation in IoT[J]. Sensors, 2018, 18(6): 1814-1823.

[3] Abdullah A M. Advanced encryption standard (AES) algorithm to encrypt and decrypt data[J]. Cryptography and Network Security, 2017(16): 1-11.

[4] Selent D. Advanced encryption standard[J]. Rivier Academic Journal, 2001, 6(2):1-4.

[5] Zhou J, Cao Z, Dong X, et al. Security and privacy for cloud-based IoT: Challenges[J]. IEEE Communications Magazine, 2017, 55(1):26-33.

[6] 施迅, 叶思海. 保护密钥授权数据的方法、设备和TPM密钥管理中心:CN104618096B[P]. 2018-10-30.

[7] Chang C C, Sun C Y, Chou Y C. Novel and practical scheme based on secret sharing for laptop data protection[J]. IET Information Security, 2015, 9(2):100-107.

[8] Dawson E, Donovan D. The breadth of Shamir's secret-sharing scheme[J]. Computers & Security, 1994, 13(1):69-78.

[9] Fatima S, Ahmad S. Secure and effective key management using secret sharing schemes in cloud computing[J]. International Journal of e-Collaboration, 2020, 16(1):1-15.

[10] Abdel Hakeem S A, Kim H W. Centralized threshold key generation protocol based on shamir secret sharing and HMAC authentication[J]. Sensors, 2022, 22(1): 331-340.

[11] Lee D, Kohlbrenner D, Shinde S, et al. Proceedings of the Fifteenth European Conference on Computer Systems, April 15, 2020[C]. New York: ACM,2020.

[12] Abhishek K, Roshan S, Kumar P, et al. A comprehensive study on multifactor authentication schemes[M]. Advances in Computing and Information Technology. Berlin: Springer, 2013.

[13] Oke B A, Olaniyi O M, Aboaba A A, et al. Multifactor authentication technique for a secure electronic voting system[J]. Electronic Government, an International Journal, 2021, 17(3): 312-338.

[14] Lamport L. Password authentication with insecure communication[J]. Communications of the ACM, 1981, 24(11): 770-772.

[15] Wang X M, Zhang W F, Zhang J S, et al. Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards[J]. Computer Standards & Interfaces, 2007, 29(5):507-512.

[16] Ahvanooey M T, Zhu M X, Li Q, et al. Modern authentication schemes in smartphones and IoT devices: An empirical survey[J]. IEEE Internet of Things Journal, 2021, 9(10): 7639-7663.

[17] Wang D, Cheng H, Wang P, et al. Zipf's Law in passwords[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(11):2776-2791.

[18] Sengupta A, Singh A, Kumar P, et al. A secure and improved two factor authentication scheme using elliptic curve and bilinear pairing for cyber physical systems[J]. Multimedia Tools and Applications, 2022, 81(16): 22425-22448.

[19] Aguboshim F C. Strategies for enhancing ICT innovations system security in Nigeria[J]. International Journal of Advances in Engineering and Management (IJAEM), 2021, 3(2): 37-43.

[20] Lee E, Seo Y D, Oh S R, et al. A survey on standards for interoperability and security in the internet of things[J]. IEEE Communications Surveys & Tutorials, 2021, 23(2): 1020-1047.

[21] Liu Y, Zhong Q, Chang L, et al. A secure data backup scheme using multi-factor authentication[J]. IET Information Security, 2017, 11(5): 250-255.

[22] Hu H, Lin C, Chang C C, et al. Enhanced secure data backup scheme using multi-factor authentication[J]. IET Information Security, 2019, 13(6):649-658.

[23] Mirzadi K, Mohasefi J B. An Ultra-lightweight mutual authentication protocol based on LPN problem with distance fraud resistant[J]. Wireless Personal Communications, 2021, 117(3): 2225-2251.

[24] Roy S, Chatterjee S, Das A K, et al. Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing internet of things[J]. IEEE Internet of Things Journal, 2017, 5(4): 2884-2895.

[25] Canetti R, Fuller B, Paneth O, et al. Reusable fuzzy extractors for low-entropy distributions[J]. Journal of Cryptology, 2021, 34(1): 1-33.

[26] Burrows M, Abadi M, Needham R. A logic of authentication[J]. ACM Transactions on Computer Systems (TOCS), 1990, 8(1):18-36.

[27] Patel C, Joshi D, Doshi N, et al. An enhanced approach for three factor remote user authentication in multi-server environment[J]. Journal of Intelligent & Fuzzy Systems, 2020, 39(6): 8609-8620.

[28] Wei D. Crypto++ Library 5.6.3[OL/EB]. (2015-11-20)[2023-05-20]. http://www.cryptopp.com, accessed at 20 May 2022.

[29] Jiang Q, Khan M K, Lu X, et al. A privacy preserving three-factor authentication protocol for e-health clouds[J]. The Journal of Supercomputing, 2016, 72(10): 3826-3849.

【责任编辑:陈　钢】