软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2024,35(10):4859−4875 [doi: 10.13328/j.cnki.jos.006966]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

以用户为中心的双因子认证密钥协商协议*

杨 雪[1], 刘怡静[1], 姜 奇[1], 王金花[2], 李兴华[1]

[1](西安电子科技大学 网络与信息安全学院, 陕西 西安 710119)

[2](保密通信重点实验室, 四川 成都 610041)

通信作者: 姜奇, E-mail: jiangqixdu@gmail.com

摘 要: 当前基于用户名和口令的认证协议已难以满足日益增长的安全需求. 具体而言, 用户选择不同口令访问不同在线服务, 极大地增加了用户记忆负担; 此外, 口令认证安全性低, 面临许多已知攻击. 为了解决此类问题, 基于PS (Pointcheval-Sanders) 签名提出一个以用户为中心的双因子认证密钥协商协议 UC-2FAKA. 首先, 为防止认证因子泄露, 基于 PS 签名构造口令和生物特征双因子凭证, 并以零知识证明的方式向服务提供商 (service provider, SP) 验证身份; 其次, 采用以用户为中心的单点登录 (single sign on, SSO) 架构, 用户可以通过向身份提供商 (identity provider, IDP) 注册请求身份凭证来向不同的 SP 登录, 避免 IDP 和 SP 跟踪或链接用户; 再次, 采用 Diffie-Hellman 密钥交换认证 SP 身份并协商通信密钥, 保证后续的通信安全; 最后, 对所提出协议进行全面的安全性分析和性能对比, 结果表明所提出协议能够抵御各种已知攻击, 且所提出协议在通信开销和计算开销上表现更优.

关键词: 口令; 认证; 凭证; 双因子

中图法分类号: TP309

中文引用格式: 杨雪, 刘怡静, 姜奇, 王金花, 李兴华. 以用户为中心的双因子认证密钥协商协议. 软件学报, 2024, 35(10): 4859–4875. http://www.jos.org.cn/1000-9825/6966.htm

英文引用格式: Yang X, Liu YJ, Jiang Q, Wang JH, Li XH. User-centric Two-factor Authentication Key Agreement Protocol. Ruan Jian Xue Bao/Journal of Software, 2024, 35(10): 4859–4875 (in Chinese). http://www.jos.org.cn/1000-9825/6966.htm

User-centric Two-factor Authentication Key Agreement Protocol

YANG Xue[1], LIU Yi-Jing[1], JIANG Qi[1], WANG Jin-Hua[2], LI Xing-Hua[1]

[1](School of Cyber Engineering, Xidian University, Xi'an 710119, China)

[2](Science and Technology on Communication Security Laboratory, Chengdu 610041, China)

Abstract: The current authentication protocol based on username and password has been difficult to meet the increasing security requirements. Specifically, users choose different passwords to access different online services, which greatly increases the user's memory burden. In addition, password authentication has low security and faces many known attacks. To solve such problems, this study proposes a user-centric two-factor authentication key agreement protocol UC-2FAKA based on the Pointcheval-Sanders signature. Firstly, to prevent the leakage of authentication factors, passwords, and biometric two-factor credentials are constructed based on the Pointcheval-Sanders signature. The identity is authenticated to the service provider (SP) in a zero-knowledge proof manner. Secondly, using a user-centric single sign on (SSO) architecture, users can request identity credentials by registering with an identity provider (IDP) to log in different SPs to avoid IDP or SP tracking or linking users. Thirdly, the Diffie-Hellman key exchange is used to authenticate SP identities and negotiate communication keys to ensure subsequent communication security. Finally, comprehensive security analysis and performance comparison of the proposed protocol are carried out. The results show that the proposed protocol can resist various known attacks, and the proposed protocol performs better in communication overhead and computational overhead.

Key words: password; authentication; certificate; two-factor

* 基金项目: 国家自然科学基金 (62072352, 62125205, 92167203, 62072359); 陕西省重点产业链项目 (2020ZDLGY09-06)
收稿时间: 2022-11-08; 修改时间: 2022-12-14; 采用时间: 2023-05-11; jos 在线出版时间: 2023-10-25
CNKI 网络首发时间: 2023-10-26

随着移动计算与互联网应用的快速发展, 多种多样的互联网在线服务应运而生, 如网络通信、在线交易、远程办公、健康监测等, 覆盖了人们日常生活的方方面面. 用户与各类服务提供商的交互过程往往涉及大量敏感信息. 由于网络的开放性, 其交互过程可能遭受各种攻击, 轻则泄露用户身份, 重则造成财产损失等. 作为保护用户隐私和安全的第 1 道防线, 身份认证对在线服务而言是必不可少的安全机制.

在网络攻击日渐复杂的背景下, 传统的口令认证已不能满足要求. 究其原因, 人类的记忆力是有限的. 而对于每个在线服务, 用户均要记住一个安全口令, 这将导致用户在口令的选择上趋于简单化、单一化, 难以抵御离线字典攻击. 诸多文献提出结合两个及以上认证因子 (口令、指纹、虹膜、硬件令牌等) 以实现高安全性的多因子身份认证[1]. 仅当敌手攻破所有的认证因子, 才可伪装为合法用户.

随着在线服务数量不断增加, 用户需向每个服务提供商 (service provider, SP) 提供口令、生物特征等认证因子. 由此带来了两方面问题, 一是大大增加了用户管理身份的负担, 二是导致用户身份隐私泄露风险急剧上升. 单点登录 (single sign on, SSO) 可缓解上述问题, 仅登录一次就可访问不同 SP 且无需向 SP 提供口令等认证信息, 一定程度上保护了用户的身份隐私.

当前的 SSO 可分为两类, 一类以身份提供商 (identity provider, IDP) 为中心, 最具代表性且部署最广泛的是 OpenID Connect[2]. 用户在 IDP 处进行单点注册和认证, 进而访问不同的 SP. 但该方案存在隐私泄露问题. 首先, 登录某些 SP 的行为本身就会泄露有关用户的敏感信息. 例如为心理疾病患者提供在线聊天室, 或为特殊障碍人群提供帮助的网站. 其次, IDP 参与了注册和认证过程, 不仅可以追踪用户登录的 SP, 还可以通过已验证的认证因子链接到唯一标识的用户[3]. 而允许 IDP 获取用户登录信息必须高度信任 IDP 有能力保护隐私数据免受内部和外部人员攻击, 显然最佳方案是能够减少对这种高度信任的需求. 另一类是以用户为中心的 SSO[4]. 用户在 IDP 注册, 申请凭证并保存; 之后, 用户利用凭证向 SP 进行身份认证. 认证过程中不涉及与 IDP 的交互, IDP 无法跟踪和链接用户, 大大减少了用户信息泄露的可能性.

综上所述, 迫切需要研究以用户为中心的多因子认证方案. 尽管已有相关研究[5,6], 但现有方案仍不尽完善. 文献 [5] 提出了一种以用户为中心的基于生物特征的三因子认证解决方案, 用户可以通过设备向不同的 SP 进行认证, 而无需在交易中涉及 IDP. 该方案基于生物特征和用户提供的秘密构造多因子凭证, 并与 SP 执行认证协议. 然而, 在注册过程中, 用户将口令和生物特征模板以明文形式发送给 IDP 来构造凭证, 容易引发内部人员攻击. 文献 [6] 以承诺的形式传输认证因子以实现双因子认证, 保证了即使敌手得到承诺值也不能实现身份伪造攻击, 但是该方案在设备处以明文形式存储认证因子, 无法抵御在线猜测攻击.

本文提出了一个以用户为中心的双因子 (口令和生物特征) 认证密钥协商 UC-2FAKA 协议, 其以用户为中心体现在用户可以独立管理和控制其个人信息的使用, 自己保存身份隐私信息, 而不是由 SP 或 IDP 保存; 而且在认证过程中不涉及与 IDP 的交互, 尽可能降低认证因子泄露的风险. 具体来说, 在注册时, 首先, 用户利用承诺构造双因子凭证; 其次, 用户与 IDP 执行零知识证明协议, 证明用户拥有口令和生物特征; 最后, IDP 对承诺进行盲签名颁发双因子凭证. 在此过程中, IDP 无法获取用户的认证信息和身份, 实现对用户的隐私保护, 并确保 IDP 对用户的不可跟踪性. 在认证过程中, 用户将凭证随机化发送给 SP, 防止共谋的 SP 链接同一用户, 实现 SP 对用户的不可链接性; 然后用户使用签名的知识证明向 SP 证明凭证的真实性, 有效防止了认证因子泄露.

本文第 1 节讨论相关工作. 第 2 节介绍 UC-2FAKA 协议中使用的相关技术知识. 第 3 节详细描述 UC-2FAKA 协议的具体流程. 第 4 节提出针对 UC-2FAKA 协议的安全性分析. 第 5 节描述 UC-2FAKA 的实验以及开销对比. 第 6 节进行总结.

1 相关工作

认证是防止非法访问数据或敏感应用程序的基本保障, 用户认证方式分为单因子认证和多因子认证, 组成架构分为独立认证系统和单点登录系统. 下面从多因子认证协议和单点登录系统两方面来回顾相关研究.

1.1 多因子认证协议

口令认证是应用最广泛的单因子认证, 然而当前口令认证存在问题[7]. 第一, 用户从口令集中选择口令, 可能

会导致敌手的在线猜测攻击. 第二, 敌手能根据口令字典预先计算出一个查询表, 一旦成功腐化服务器就能找到口令. 综上, 口令认证协议存在的主要问题是服务器必须要存储口令相关的信息, 由此导致的内部攻击是难以避免的.

硬件的引入一定程度上弥补了口令认证的安全性问题. 文献 [8] 提出基于智能卡和口令的双因子协议, 将口令与智能卡结合加强了认证的安全性, 并声称可以抵挡离线字典攻击. 然而, 文献 [9] 证明了文献 [8] 在假设敌手能够窃取智能卡的情况下, 容易受到离线字典攻击. 随后, 文献 [10] 给出一个基于口令和智能卡的双因子认证协议, 将“蜜词”[11]与“模糊验证器”相结合引入到双因子加密协议设计中, 协议可以及时检测到智能卡的损坏, 从而阻止在线猜测攻击. 针对服务器腐化问题, 文献 [12] 提出了一种抵抗服务器腐化的基于口令和设备的双因子认证方案. 但是, 其依赖于公钥基础设施, 不能阻止服务器访问明文口令, 这导致了密钥泄露模拟 (KCI) 攻击. 文献 [13] 提出了一个针对文献 [12] 改进的双因子认证方案 OpTFA, 以解决所有上述漏洞. 然而, OpTFA 过于复杂且效率低下.

生物特征因其便捷性和唯一性在多因子认证协议中受到众多研究学者的关注. 文献 [14] 在共同参考串模型下设计了一个基于口令、生物特征和私钥的三因子认证协议. 但是, 服务器与用户同时明文存储了用户的认证因子, 没有考虑用户身份隐私问题. 文献 [15] 提出一个三因子认证协议, 结合口令、生物特征和设备构造三因子凭证 $Z = H^{(\alpha+\beta+\gamma)}$ (α,β,γ 分别代表口令、生物特征和设备) 以实现认证. 尽管该方案在 Bellare-Pointcheval-Rogaway (BPR) 模型 [16]中被证明是安全的, 但其仍然受到 KCI 攻击. 服务器需要额外存储用户认证凭证, 带来了凭证泄露风险. 文献 [17] 提出基于口令的凭证的概念, 其中用户通过生成指定的可验证的身份验证令牌, 使用口令加密的证书进行身份验证. 尽管攻击者可以窃取受口令保护的证书, 但方案仍可以抵抗离线字典攻击.

综合上述文献, 多因子认证提供了更高的安全性, 然而身份隐私问题亟待解决. 为此, 本文将重点关注保护隐私的多因子认证协议.

1.2 单点登录系统

随着网络技术的不断发展, 用户通过访问多个服务器获取不同服务. 然而在传统的认证方案中, 获取服务需要多次注册、多次认证. SSO 系统[18]使得用户在一个中心机构注册, 便能访问多个服务器. 本文按照以 IDP 为中心和以用户为中心将认证系统分类. 以 IDP 为中心的认证系统中应用最广泛的身份认证协议是 OpenID Connect[2], 其是一种在网络中委托身份认证的协议. 由于 SP 依赖 IDP 实现用户认证, IDP 能够跟踪用户并获取用户信息, 隐藏着巨大的身份隐私危机. 为了克服 OpenID Connect 存在的隐私问题, 文献 [19] 提出解决方案. 使用 SP 的假名和用户假名以及身份转换算法确定用户唯一账号实现认证过程. 以用户为中心的代表性实践为 FIDO 的 UAF 协议[20]. 在 UAF 协议中, 用户在 SP 处注册 UAF 设备, 并生成特定于 SP 的密钥对. 在注册阶段, 选择服务器支持的本地认证机制, 如指纹识别, 输入 PIN 码, 行为识别等. 服务器也可以保留口令认证, 将口令和生物特征相结合, 增强认证安全性.

匿名认证是保障 SSO 系统隐私的坚实基础, 其将 IDP 生成的凭证与传输给 SP 的凭证解耦, 使 IDP 无法链接用户登录的 SP. 虽然研究人员已对匿名认证协议进行大量研究[21−25], 但仍存在不足之处. 例如, 文献 [22] 的协议无法实现所声称的离线口令猜测攻击, 且未实现用户匿名性和前向安全性; 文献 [23] 的协议同样不能抵抗离线口令猜测攻击, 且不能提供匿名性, 对两种破坏前向安全性的攻击是脆弱的; 文献 [24] 的协议不能抵抗所声称的用户仿冒攻击和离线口令猜测攻击, 且无法实现用户不可追踪性. 与上述协议不同, EL PASSO[4]是一个以用户为中心的基于匿名凭证的 Web 单点登录系统, 该系统可以仅向 SP 展示登录所需的信息, 而无需透露其他无关信息. 另外, 该系统支持多设备部署、设备盗窃恢复和隐私保护的身份验证. DAMFA[25]是一个匿名双因子认证方案, 使用不经意伪随机函数 (oblivious pseudorandom function, OPRF) 将用户口令和生物特征结合, 使用假名和零知识证明实现了用户不可追踪性, 用户认证过程不再依赖可信第三方.

综合上述文献, 以用户为中心的认证系统在一定程度上保证了用户的身份隐私, 防止了 IDP 跟踪用户. 本文将重点研究以用户为中心的匿名认证协议.

2 基础知识

本节主要介绍所提出协议中使用到的相关概念和技术, 分别包括零知识证明协议、PS 签名以及模糊提取器.

2.1 零知识证明

零知识证明协议[26]的参与方包括证明方 P 和验证方 V. 协议的目标是在 P 未向 V 提供秘密的明文值的情况下, 使 V 确信 P 拥有秘密, 以此保护 P 的身份隐私. 在所提出协议中, 参考文献 [27], 零知识证明用于证明 Pedersen 承诺[28] $C=g^r h^s \bmod p$ 中的秘密 s 和盲化因子 r 的知识.

如图 1 所示, 首先, 证明方 P 选择两个随机数 t_1,t_2, 并构造辅助承诺 $T=g^{t_1}h^{t_2} \bmod p$ 将其发给 V; 然后 V 选择随机数 e 发送给 P; P 构造 $s_1=r\cdot e+t_1 \bmod q$, $s_2=s\cdot e+t_2 \bmod q$ 并发送给 V, 其中 r 和 s 是 P 的秘密; 最后, V 验证 $g^{s_1}h^{s_2}=C^eT$, 如果成立, P 成功向 V 证明其拥有秘密, 否则验证失败.

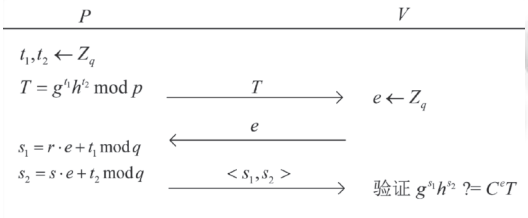

图 1 零知识证明协议

2.2 PS 签名

PS 签名是 Pointcheval 和 Sanders 提出的多消息签名方案[29], 该方案允许一个签名方对多个消息进行签名, 并采用双线性对映射实现签名验证. PS 签名还可以结合承诺实现消息的隐式传输. 用户使用承诺的零知识证明向签名方证明其拥有消息 m, 当签名方验证成功后, 签名方将使用其私钥对承诺进行签名. 基于承诺的 PS 签名的具体流程如下.

(1) 生成参数和公私钥对

给定安全参数 k, 生成两个双线性群 G_1 和 G_2, g 和 $\tilde{g}$ 分别是 G_1 和 G_2 的生成元. 在 Z_p^* 中选择随机数 x 和 y, 计算 $X=g^x$, $\tilde{X}=\tilde{g}^x$, $Y_i=g^{y_i}$, $\tilde{Y}_i=\tilde{g}^{y_i}$. 其中 X 是签名方的私钥, $\{\tilde{X},Y_i,\tilde{Y}_i\}$ 是签名方的公钥, i 表示消息的个数.

(2) 生成签名

用户构造承诺 $C=g^r\prod Y_i^{m_i} \bmod p$, 并将其发送给签名方. 用户与签名方执行零知识证明协议. 验证成功后, 签名方随机选择了一个随机数 u, 并使用私钥 X 进行签名, 得到签名 $\sigma'=(\sigma'_1,\sigma'_2)=(g^u,(XC)^u)$ 并发给用户, 用户收到签名后计算 $\sigma \leftarrow (\sigma'_1,\sigma'_2/(\sigma'^r_1))$ 并存储.

(3) 验证签名

用户向验证方发送签名 σ, 验证方使用 $e(\sigma_1,\tilde{X}\cdot\prod Y_i^{m_j})=e(\sigma_2,\tilde{g})$ 验证签名.

2.3 模糊提取器

模糊提取器[30]可以从用户生物信息 (比如指纹和人脸图像) 中提取出安全的密钥, 并通过密码操作来实现认证且不需要保存密钥. 模糊提取器允许存在一定的干扰噪声, 只要生物模板的输入在一定误差范围内, 都可以得到一个相同的均匀字符串.

模糊提取器由一对概率多项式时间算法 $FE=(FE.Gen,FE.Rep)$ 组成, 具体过程如下.

- $(R,P) \leftarrow FE.Gen(w)$: 输入生物特征模板 w, 输出一个均匀字符串 R 和一个公开的辅助字符串 P.
- $R \leftarrow FE.Rep(P,w')$: 输入辅助字符串 P 和认证模板 w', 如果 $Dist(w,w')$ 不超过预定的阈值, 则该算法恢复字符串 R, 否则输出 $\perp$.

3 多服务器场景下的双因子认证协议

在本节中, 定义了多服务器场景下双因子认证协议的系统模型和安全模型, 并详细阐述了基于口令和生物特征的双因子认证密钥协商协议 UC-2FAKA 的初始化阶段、注册阶段以及认证与密钥协商阶段的流程.

3.1 系统模型

随着互联网的发展, 认证场景逐渐多元化, 本节针对多服务器场景下的认证系统展开研究. 如图 2 所示, 系统由 3 个参与实体构成: 用户、身份提供商 IDP 和服务提供商 SPs.

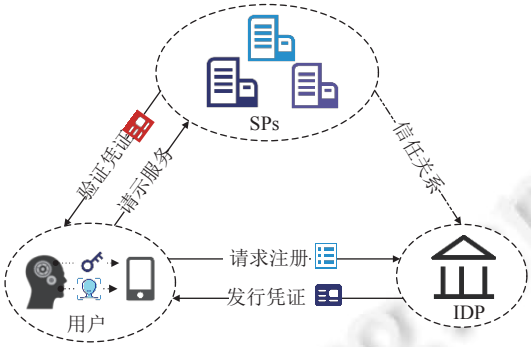

图 2 系统模型

- 用户 (User): 持有手机客户端的主体, 与 IDP 进行交互获取身份凭证, 与 SP 交互实现身份认证并获取服务.
- IDP: 身份提供商, 验证用户身份并为其颁发身份凭证.
- SPs: 服务提供商集合, 验证用户的身份凭证的真实性并为其提供服务.

在该场景中, IDP 利用 PS 签名向用户颁发一个与该用户口令和生物特征相关的身份凭证, 此后用户可以使用该凭证来向 SP 执行认证操作并协商会话密钥. 基于口令和生物特征的双因子认证密钥协商协议 UC-2FAKA 定义如下.

定义 1. UC-2FAKA. 基于口令和生物特征的双因子认证密钥协商协议由以下几个算法组成.

- $Setup(1^{\lambda}) \to pp$: 给定一个安全参数 $\lambda \in N$, 输出一系列公共参数 pp.
- $IDPKeyGen(pp) \to Isk, Ipk$: 输入安全参数 pp, 运行该算法, 为身份提供商 IDP 生成私钥 Isk 和公钥集合 $Ipk = \{Ipk_1, Ipk_2, \ldots, Ipk_r\}$.
- $SPKeyGen(pp) \to sk, pk$: 输入安全参数 pp, 运行该算法, 为服务提供商生成公私钥对 sk, pk.
- $Regist(pwd, Bio, Isk) \to \sigma$: 用户输入口令和生物特征向 IDP 注册凭证, 运行该算法, 输出一个关于口令和生物特征的凭证 σ.
- $Auth(pwd', Bio', P, \sigma, Ipk) \to SK$: 用户输入口令、新采样的生物特征、生物特征辅助数据以及凭证, 运行该算法, SP 与用户相互认证并协商会话密钥 SK.

3.2 设计思想

针对上述系统, 本文所提出的 UC-2FAKA 协议在认证安全和身份隐私方面的设计思想如下.

- 在 IDP 为用户颁发身份凭证之前需验证用户的个人信息来确定其有效性, 若直接将口令和生物特征发送给 IDP 会泄露用户身份隐私. 因此本文考虑将口令和生物特征以承诺的形式相结合, 并使用零知识证明协议向 IDP 证明其拥有口令和生物特征.
- 在 IDP 为用户颁发身份凭证时, IDP 为包含口令和生物特征的承诺进行盲签名操作, 这样用户得到了一个基于口令和生物特征的凭证, 只有该凭证的真正拥有者才能使用其来证明身份.
- 在用户向 SP 证明身份时, 用户对凭证进行随机化处理, 每次登录都使用不同的随机化后的凭证来验证用户身份, 确保 SP 对用户的不可链接性; 并且 SP 为验证凭证中的认证因子与用户执行零知识证明协议, 防止认证因子泄露.

3.3 安全模型

在本节参考文献 [6,10] 的安全模型对 UC-2FAKA 进行建模并给出安全性定义.

- 参与方: 参与方包括用户集合 U 和服务提供商集合 S, 其中用户和服务提供商不相交, 它们中的每一个都可

以有多个预言机实例参与不同的、可能并发的协议 π 的执行. 用 $\prod_U^i, \prod_S^j$ $(i, j \in Z)$ 分别表示客户端实例和服务提供商实例, 用 $\prod_P^i$ 表示任意类型实例. 另外, 每个用户均持有一个集合 $X = \{pwd, Bio\}$, 其中 pwd 表示从用户的口令字典 D 中选取的口令, Bio 表示用户的生物特征模板. 每个服务提供商均持有密钥对 $\{sk, pk\}$.

- 伙伴关系: 两个实例 $\prod_U^i$ (客户端实例) 和 $\prod_S^j$ (SP 实例), pid 表示实例的合作方标识. 当满足以下条件时, 称 $\prod_U^i$ 和 $\prod_S^j$ 是伙伴关系: 1) $\prod_U^i$ 的 pid 是 $\prod_S^j$, $\prod_S^j$ 的 pid 是 $\prod_U^i$; 2) 双方通信的会话文本一致; 3) 双方协议结束会话时, 均表示接受对方.

设置概率多项式时间 (probabilistic polynomial time, PPT) 敌手 A 与协议参与者之间的交互仅通过预言机查询发生, 查询预言机的动作模拟了真实攻击中的敌手能力, A 可以使用的查询类型如下所述.

- $Send(\prod_U^i, Start)$: 该查询表示对协议 π 进行初始化.
- $Send(\prod_P^i, m)$: 该查询模拟主动攻击, 即 A 拦截一条消息且伪造消息 m 后, 向实例 $\prod_P^i$ 发送 m 并获得该实例的响应.
- $Execute(\prod_U^i, \prod_S^j)$: 该查询模拟被动攻击, 它的输出包括实例 $\prod_U^i$ 和 $\prod_S^j$ 之间的所有通信记录.
- $Reveal(\prod_P^i)$: 该查询模拟对会话密钥的滥用, 即如果实例 $\prod_P^i$ 接受会话并生成会话密钥 SK, 则该实例用 SK 响应 A, 否则, 返回 $\perp$.
- $Corrupt(\prod_S^j)$: 该查询向 A 返回实例 $\prod_S^j$ 的长期私钥 sk_i.
- $Corrupt(\prod_U^i, a)$: A 可以揭露口令或生物特征中的任意一个, 但不能同时揭露这两个因子: 如果 $a = 0$, 该预言机输出实例 $\prod_U^i$ 对应的口令 pwd, 并将 $\prod_U^i$ 添加到 L_{pwd} 中; 如果 $a = 1$, 该预言机输出实例 $\prod_U^i$ 对应的生物特征 Bio, 并将 $\prod_U^i$ 添加到 L_{Bio} 中.
- $Test(\prod_P^i)$: 该查询模拟会话密钥的语义安全. 收到该查询后, 模拟抛掷一枚硬币 c, 如果 $b = 0$, 则预言机返回一个与 SK 等长的随机值; 如果 $b = 1$, 则预言机返回 SK, 如果实例 $\prod_P^i$ 没有生成 SK 则返回 $\perp$. 在敌手的执行时间内, 该查询可以在任何时候被调用但只能调用一次.

定义 2. CDH 假设. 计算 Diffie-Hellman 难题 (computational Diffie-Hellman problem, CDH) 给定输入 $g^a, g^b \in G$, 计算输出 $g^{ab} \bmod p$, 其中 $g \in G$ 是群 G 的生成元. CDH 假设表示任何概率多项式时间 (PPT) 敌手 A 成功解决 CDH 难题的优势 $Adv^{\mathrm{CDH}}(A) = \Pr[Succ^{\mathrm{CDH}}(A)] = \Pr[A(g^a, g^b, g^{ab}) = 1 : a, b \in Z_p^*]$ 是可忽略的.

定义 3. AKA-安全. 设 $Succ(A)$ 表示敌手 A 对一些新接受的实例进行 $Test(\prod_P^i)$ 查询, 并将 b' 作为猜测位于 $Test$ 查询所选择的位 b 相对照. 敌手破坏协议 π 的语义安全优势表现为 $Adv_\pi^{\mathrm{AKA}}(A) = |2\Pr[Succ^{\mathrm{AKA}}(A)] - 1| = |2\Pr[b' = b] - 1|$. 如果对于任何 PPT 敌手 A, 其优势 $Adv_{\mathrm{UC\text{-}2FAKA}}^{\mathrm{AKA}}(A)$ 可以忽略不计, 则 UC-2FAKA 实现 AKA 安全.

3.4 协议描述

在本文所提出的协议 UC-2FAKA 中, IDP 利用 PS 签名对用户颁发基于口令和生物特征的双因子身份凭证, 此后用户可以使用该凭证来向 SP 执行认证操作. 具体构造如下.

(1) 初始化阶段

$Setup(1^\lambda) \to pp$: 给定安全参数 λ, 生成公共参数 $pp = (p, q, G_1, G_2, G_T, g, \tilde{g})$, 其中 p, q 是两个大素数, 满足 $q | p - 1$, G_1, G_2 是阶为 p 的循环群, 存在映射关系 $G_1 \times G_2 \to G_T$, $g, \tilde{g}$ 分别是 G_1, G_2 的生成元. 选择 SHA-1 哈希函数 $H(\cdot)$ 和带密钥的消息验证函数 $MAC_k(\cdot)$.

(2) 密钥生成阶段

$IDPKeyGen(pp) \to Isk, Ipk$: IDP 选择随机数 $(x, y_1, y_2) \in Z_p^*$, 并计算 $(X, Y_1, Y_2) \leftarrow (g^x, g^{y_1}, g^{y_2})$ 和 $(\tilde{X}, \tilde{Y}_1, \tilde{Y}_2) \leftarrow (\tilde{g}^x, \tilde{g}^{y_1}, \tilde{g}^{y_2})$, 将 X 作为 IDP 的签名私钥, $(Y_1, Y_2, \tilde{X}, \tilde{Y}_1, \tilde{Y}_2)$ 作为 IDP 的公钥公开.

$SPKeyGen(pp) \to sk, pk$: SP 选择私钥 $s \leftarrow Z_p^*$, 计算公钥 $S = g^s$, 即 $pk = S, sk = s$ 并公开 pk.

(3) 注册阶段

$Regist(pwd, Bio, Isk) \to \sigma$: 该过程在安全信道上执行, 如图 3 所示.

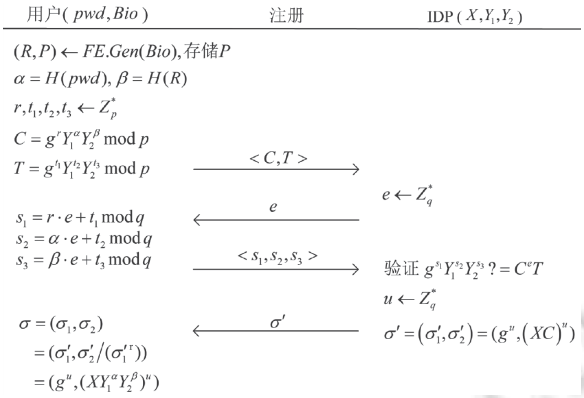

图 3 UC-2FAKA 注册阶段

1) 用户输入自己的生物特征 Bio 和口令 pwd, 执行模糊提取器 $(R,P) \leftarrow FE.Gen(Bio)$, 用户设备存储 P, 并计算口令哈希 $\alpha = H(pwd)$ 以及生物密钥哈希 $\beta = H(R)$, 选择随机数 $r \leftarrow Z_p^*$, 使用 IDP 的公钥 Y_1, Y_2 来构造承诺 $C = g^r Y_1^{\alpha} Y_2^{\beta} \bmod p$, 并将承诺 C 发送给 IDP.

2) 用户向 IDP 请求身份凭证, 并与 IDP 执行零知识证明协议.

a) 用户选择随机数 $t_1, t_2, t_3 \leftarrow Z_p^*$, 构造辅助承诺 $T = g^{t_1} Y_1^{t_2} Y_2^{t_3} \bmod p$ 并发送给 IDP.

b) IDP 收到 T 后, 选择随机挑战 e 发送给用户.

c) 根据零知识证明协议, 用户使用挑战 e 构造 $s_1 = r \cdot e + t_1 \bmod q$, $s_2 = \alpha \cdot e + t_2 \bmod q$ 和 $s_3 = \beta \cdot e + t_3 \bmod q$, 并将 $< s_1, s_2, s_3 >$ 发送给 IDP.

d) IDP 收到消息后, 使用 $g^{s_1} Y_1^{s_2} Y_2^{s_3} ? = C^e T$ 来验证承诺 C 中 $< r, \alpha, \beta >$ 的正确性, 验证成功则表示用户拥有口令 pwd 和生物密钥 R, 而无需向 IDP 揭示关于口令和生物密钥的明文值.

3) 验证成功后, IDP 选择了一个随机数 u 并使用私钥 X 对承诺进行签名, 获得盲签名 $\sigma' = (\sigma'_1, \sigma'_2) = (g^u, (XC)^u)$, 并将 σ' 发送给用户.

4) 用户收到 σ' 后, 使用随机数 r 对 σ' 进行去盲操作, 得到凭证 $\sigma = (\sigma_1, \sigma_2) = (\sigma'_1, \sigma'_2/(\sigma'^r_1)) = (g^u, (XY_1^{\alpha} Y_2^{\beta})^u)$ 并保存到数据库中.

(4) 认证与密钥协商阶段

$Auth(pwd', Bio', P, \sigma, Ipk) \rightarrow SK$: 如图 4 所示, 用户与 SP 相互认证并协商密钥. 用户拥有认证口令 pwd', 认证生物模板 Bio' 和辅助字符串 P 以及凭证 σ. SP 持有公共参数和 IDP 所有公钥. 首先, 用户通过向 SP 证明凭证的有效性来实现身份认证, 即用户与 SP 执行签名的知识证明协议, 在不揭露口令和生物特征的情况下, 向 SP 证明身份. 然后, SP 利用 Diffie-Hellman 密钥交换和私钥 sk 来向用户验证身份. 最后为保证后续安全通信, 用户与 SP 通过密钥派生函数协商一个会话密钥. 具体过程如下.

1) 用户选择随机数 t 和 v, 计算随机化凭证 $\sigma'' = (\sigma''_1, \sigma''_2) = (\sigma_1^v, (\sigma_2 \sigma_1'^t)^v) = (g^{uv}, (X \cdot g^t Y_1^{\alpha} Y_2^{\beta})^{uv})$, 目的是防止多个 SP 联合起来通过同一凭证链接到同一用户. 然后用户选择随机数 n_1, n_2, n_3 和 N, 构造辅助承诺 $T' = \tilde{g}^{n_1} \tilde{Y}_1^{n_2} \tilde{Y}_2^{n_3} \bmod p$, 并将 $< \sigma'', T', N >$ 发送给 SP.

2) SP 收到用户发送的消息后, 选择挑战 b, 并使用私钥 sk 计算 $A_1 = \tilde{g}^s \bmod p$, $A_2 = \tilde{Y}_1^s \bmod p$ 和 $A_3 = \tilde{Y}_2^s \bmod p$. 随后基于 Diffie-Hellman 密钥交换, 构造密钥 $K = T'^s \bmod p$, 根据文献 [31] 中的密钥派生函数计算 $K' = H(00 \| T' \| S \| K)$. 利用 N 和 K' 执行 MAC 操作, 生成 $B = MAC_{K'}(N)$, 并将消息 $< b, A_1, A_2, A_3, B >$ 发送给用户.

3) 用户收到 SP 发送的消息后, 计算口令哈希 $\alpha' = H(pwd')$, 利用模糊提取器的 $FE.Rep(P, Bio')$ 算法恢复生物密钥记为 R', 并对其进行哈希得到 $\beta' = H(R')$. 根据零知识证明协议, 用户使用挑战 b 构造 $s'_1 = t \cdot b + n_1 \bmod q$, $s'_2 = \alpha' \cdot b + n_2 \bmod q$ 和 $s'_3 = \beta' \cdot b + n_3 \bmod q$. 随后用户利用 Diffie-Hellman 密钥交换计算密钥 $K = A_1^{n_1} A_2^{n_2} A_3^{n_3} (\bmod p) =$

$T'^{s} \bmod p$, 并生成 $K' = H(00\|T'\|S\|K)$. 然后通过 $B? = MAC_{K'}(N)$ 验证与接收到的 B 是否一致, 验证一致则表示 SP 认证成功, 计算会话密钥 $K''_U = H(01\|T'\|S\|K\|\sigma'')$. 最终将 $< s'_1, s'_2, s'_3 >$ 发送给 SP.

4) SP 收到用户发送的消息后, 使用 IDP 公钥和公共参数来验证用户凭证 σ'' 的有效性. 具体而言, 利用 $e(\sigma''_1,\tilde{X})^b \cdot e(\sigma''_1,\tilde{g})^{s'_1} \cdot e(\sigma''_1,\tilde{Y}_1)^{s'_2} \cdot e(\sigma''_1,\tilde{Y}_2)^{s'_3} = e(\sigma''_2,\tilde{g})^b \cdot e(\sigma''_2,T')$ 来验证凭证 σ'' 中 α' 和 β' 的正确性. 验证成功则表示用户拥有凭证中对应的口令 pwd 和生物密钥 R, 完成了用户认证. 为保证后续的通信安全, 计算 $K''_{SP} = H(01\|T'\|S\|K\|\sigma'')$ 作为后续的通信密钥.

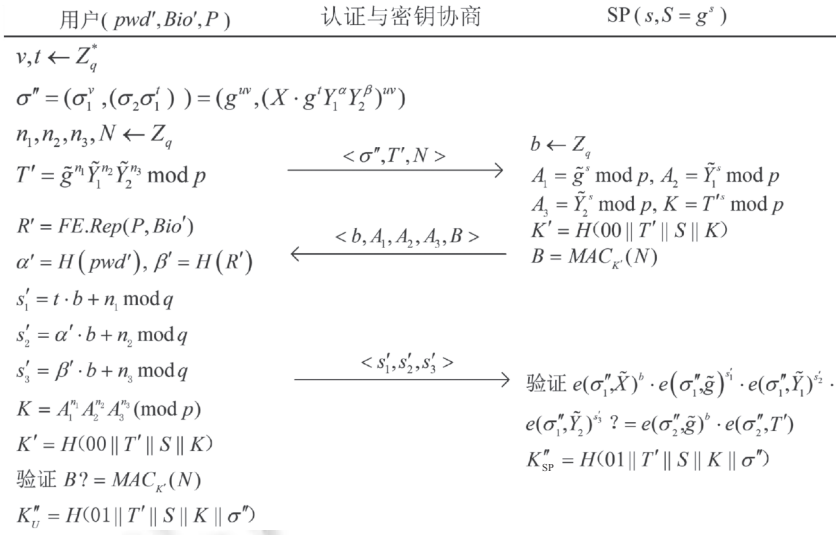

图 4 UC-2FAKA 认证与密钥协商阶段

3.5 方案正确性分析

(1) 用户认证服务提供商

标记由用户计算的 K 为 K_U, 由 SP 计算的 K 为 K_{SP}, 用户用从 SP 接收到的消息 $< b, A_1, A_2, A_3, B >$ 计算:

$$K_U = A_1^{n_1} A_2^{n_2} A_3^{n_3} \bmod p = \tilde{g}^{s \cdot n_1} \tilde{Y}_1^{s \cdot n_2} \tilde{Y}_2^{s \cdot n_3} \bmod p = T'^{s} \bmod p = K_{SP}.$$

(2) 服务提供商认证用户

SP 用从用户接收到的消息 $< s'_1, s'_2, s'_3 >$ 计算:

$$\begin{aligned} &e(\sigma''_1,\tilde{X})^b \cdot e(\sigma''_1,\tilde{g})^{s'_1} \cdot e(\sigma''_1,\tilde{Y}_1)^{s'_2} \cdot e(\sigma''_1,\tilde{Y}_2)^{s'_3} \\ &= e(g^{uv},\tilde{g}^x)^b \cdot e(g^{uv},\tilde{g}^x)^{t \cdot b + n_1} \cdot e(g^{uv},\tilde{g}^{y_1})^{\alpha' \cdot b + n_2} \cdot e(g^{uv},\tilde{g}^{y_2})^{\beta' \cdot b + n_3} \\ &= e(g,\tilde{g})^{uvb(x + t + \alpha' \cdot y_1 + \beta' \cdot y_2) + uv(n_1 + n_2 \cdot y_1 + n_3 \cdot y_2)}, \\ &e(\sigma''_2,\tilde{g})^b \cdot e(\sigma''_2,T') = e((X \cdot g^t Y_1^{\alpha} Y_2^{\beta})^{uv},\tilde{g})^b \cdot e(g^{uv},\tilde{g}^{n_1} \tilde{Y}_1^{n_2} \tilde{Y}_2^{n_3}) \\ &= e(g^{x + t + \alpha \cdot y_1 + \beta \cdot y_2},\tilde{g})^{uvb} \cdot e(g^{uv},\tilde{g}^{n_1 + n_2 \cdot y_1 + n_3 \cdot y_2}) \\ &= e(g,\tilde{g})^{uvb(x + t + \alpha \cdot y_1 + \beta \cdot y_2) + uv(n_1 + n_2 \cdot y_1 + n_3 \cdot y_2)}. \end{aligned}$$

因此, 当用户输入与注册时相同的口令和相近的生物特征, 即 $\alpha' = \alpha$, $\beta' = \beta$ 时, 有 $e(\sigma''_1,\tilde{X})^b \cdot e(\sigma''_1,\tilde{g})^{s'_1} \cdot e(\sigma''_1,\tilde{Y}_1)^{s'_2} \cdot e(\sigma''_1,\tilde{Y}_2)^{s'_3} = e(\sigma''_2,\tilde{g})^b \cdot e(\sigma''_2,T')$.

4 安全性分析

4.1 可证明安全分析

定理 1. 假设 $Adv^{AKA}_{UC\text{-}2FAKA}(A)$ 是 PPT 敌手 A 在有限时间 t 内成功破解协议语义安全的概率, 根据随机语言模拟器, 设 A 提出 q_h 次哈希查询, q_e 次 *Execute* 查询, q_s 次 *Send* 查询, 则:

$$Adv^{\mathrm{AKA}}_{\mathrm{UC\text{-}2FAKA}}(A) \leqslant \frac{q_h^2+2nq_s(nq_s-1)+4q_s}{2^l}+\frac{(q_s+q_e)^2}{p}+2(q_s+q_e)\left(Adv^{MAC}(A)+Adv^{\mathrm{PS}}(A)+\max\left\{\frac{1}{D},\frac{1}{2^{H_{\min}(X)}}+p_{\mathrm{false}}\right\}\right)$$
$$+2C'\cdot q_s^{s'}+2q_hAdv^{\mathrm{CDH}}(A) \tag{1}$$

其中, $|D|$ 是用户口令字典大小, l 是各函数输出值长度的最小值.

随机预言模拟器如下所示.

(1) *Hash* 查询: 对于哈希查询 $H(q)$ 或 $H'(q)$, 如果记录 $(q,r)\in L_H$, 则该预言机的响应为 r, 否则, 该预言机响应根据以下规则定义.

● 规则 $H^{(1)}$: 随机选择一个元素 $r\in\{0,1\}^{l_1}$, 并将 (q,r) 添加到 L_H 中. 如果查询是由敌手直接提出的, 则将 (q,r) 添加到 L_A 中.

(2) *Send* 查询 (对客户端)

1) *Send*$(\Pi_U^i, start)$ 查询: 该预言机按以下规则响应.

● 规则 $U1^{(1)}$: 选择 $v,t\leftarrow_R Z_q^*$, $n_1,n_2,n_3,N\leftarrow_R Z_q$, 计算 $\sigma''=(\sigma_1^v,(\sigma_2\sigma_1^t)^v)$, $T'=\tilde{g}^{n_1}\tilde{Y}_1^{n_2}\tilde{Y}_2^{n_3} \bmod p$.

然后查询的响应为 (σ'',T',N), 并且客户端实例进入预期状态.

2) *Send*$(\Pi_U^i,b,A_1,A_2,A_3,B)$ 查询: 如果客户端实例 Π_U^i 处于预期状态, 则该预言机按以下规则响应查询.

● 规则 $U2^{(1)}$: 计算 $R'=FE.Rep(P,Bio')$, $\alpha'=H(pwd')$, $\beta'=H(R')$, $s_1'=t\cdot b+n_1 \bmod q$, $s_2'=\alpha'\cdot b+n_2 \bmod q$, $s_3'=\beta'\cdot b+n_3 \bmod q$, $K=A_1^{n_1}A_2^{n_2}A_3^{n_3}(\bmod p)=T'^s \bmod p$, $K'=H(00\|T'\|S\|K)$, 验证 $B?=MAC_{K'}(N)$.

如果等式不成立, 则客户端实例不接受并终止, 不保存任何状态.

如果等式成立, 则计算会话密钥 $K''=H(01\|T'\|S\|K\|\sigma'')$.

然后该查询的响应为 (s_1',s_2',s_3'), 客户端实例接受并终止.

(3) *Send* 查询 (对服务器)

1) *Send*(Π_S^j,σ'',T',N) 查询: 该预言机按以下规则响应.

● 规则 $S1^{(1)}$: 选择 $b\leftarrow_R Z_q$, 计算 $A_1=\tilde{g}^s \bmod p$, $A_2=\tilde{Y}_1^s \bmod p$, $A_3=\tilde{Y}_2^s \bmod p$, $K=T'^s \bmod p$, $K'=H(00\|T'\|S\|K)$, $K''=H(01\|T'\|S\|K\|\sigma'')$ 和 $B=MAC_{K'}(N)$.

然后查询的响应为 (b,A_1,A_2,A_3,B), 并且服务器实例进入预期状态.

2) *Send*(Π_S^j,s_1',s_2',s_3') 查询: 如果服务器实例 Π_S^j 处于预期状态, 则该预言机按以下规则响应查询.

● 规则 $S2^{(1)}$: 验证 $e(\sigma_1'',\tilde{X})^b\cdot e(\sigma_1'',\tilde{g})^{s_1'}\cdot e(\sigma_1'',\tilde{Y}_1)^{s_2'}\cdot e(\sigma_1'',\tilde{Y}_2)^{s_3'}?=e(\sigma_2'',\tilde{g})^b\cdot e(\sigma_2'',T')$.

如果等式不成立, 则服务器实例不接受并终止, 不保存任何状态.

如果等式成立, 则计算会话密钥 $K''=H(01\|T'\|S\|K\|\sigma'')$.

然后服务器实例终止.

(4) *Execute*(Π_U^i,Π_S^j) 查询: 该查询是通过连续运行一系列 *Send* 查询来模拟的.

$(\sigma'',T',N)\leftarrow Send(\Pi_U^i,start)$, $(b,A_1,A_2,A_3,B)\leftarrow Send(\Pi_S^j,\sigma'',T',N)$, $(s_1',s_2',s_3')\leftarrow Send(\Pi_U^i,b,A_1,A_2,A_3,B)$, 并输出 $((\sigma'',T',N),(b,A_1,A_2,A_3,B),(s_1',s_2',s_3'))$.

(5) *Reveal*(Π_P^i) 查询: 只有在实例 Π_P^i 已经计算并接受会话密钥的情况下, 该查询返回由实例计算的会话密钥 K''.

(6) *Corrupt*(Π_U^i,a) 查询: 如果 $a=1$, 则返回口令 pw; 如果 $a=2$, 则返回生物特征 Bio. 因为本文证明中不考虑前向保密, 所以没有发生 *Corrupt*(Π_S^j)-查询.

(7) *Test*(Π_P^i) 查询: 首先从 *Reveal*(Π_P^i) 中获取会话密钥 K'', 然后模拟投掷一枚硬币. 如果 $c=1$, 则预言机返回 K''; 如果 $c=0$, 则返回一个与会话密钥等长的随机值 $K''\leftarrow\{0,1\}^{l_1}$.

定理 1 证明: 该证明过程由挑战者和敌手之间的一连串攻击游戏表示, 从模拟真实世界的攻击游戏 G_0 开始, 到 G_9 结束.

● S_i: 如果 $c=c'$, 则该事件就会发生, 其中 c 是测试查询中涉及的位, c' 是想要猜测 c 值的 A 的输出.

● $AskH_i$: 如果 A 在 $<00\|T'\|S\|K>$ 或 $<01\|T'\|S\|K\|\sigma''>$ 上进行哈希查询, 则该事件就会发生.

Game G_0: 该攻击游戏是在随机预言机下模拟真实攻击, 敌手 A 可以进行 *Send*、*Execute*、*Reveal*、*Corrupt* 和 *Test* 查询. 由定义 4 可得敌手优势为:

$$Adv_0(A) = |2\Pr[S_0] - 1| \tag{2}$$

假设如果任何一个游戏停止且 A 不输出 c', 则 c' 是随机选择的. 同样, 如果 A 在发送 q_s 次 *Send* 查询后没有完成游戏或者时间超过了预定义的时间 t, 则游戏停止, 并为 c' 选择一个随机值.

Game G_1: 该攻击游戏类似于 G_0, 不同之处在于 G_1 通过维护哈希列表 L_H 和 L_A 来模拟哈希预言机. 由于所有预言机都模拟真实攻击, 该攻击游戏无法与协议的实际执行情况区分, 因此:

$$\Pr[S_1] = \Pr[S_0] \tag{3}$$

Game G_2: 与 G_1 类似, 在攻击游戏 G_2 中模拟所有类型的查询, 但发生以下碰撞情况时模拟中止: 1) 哈希查询输出上的冲突; 2) 部分文本上的冲突: $(<\sigma'', T', N>, <b, A_1, A_2, A_3, B>, <s_1', s_2', s_3'>)$. 设 l_1 表示哈希函数的输出长度, 根据生日悖论, 有:

$$|\Pr[S_2] - \Pr[S_1]| \leqslant \frac{q_h^2}{2^{l_1+1}} + \frac{(q_s + q_e)^2}{2p} \tag{4}$$

Game G_3: 该攻击游戏与 G_2 不同之处在于, 如果存在非唯一随机值 N 或 b 的实例, 则攻击游戏中止. 最多有 nq_s 个随机值, 其中 n 为服务器和客户端所能接受的实例总数, 每个随机值都从 Z_q 中随机均匀选择, 长度为 l_0. 两个随机值相等的概率不超过 $nq_s(nq_s-1)/2^{l_0}$, 因此:

$$|\Pr[S_3] - \Pr[S_2]| \leqslant \frac{nq_s(nq_s-1)}{2^{l_0}} \tag{5}$$

Game G_4: 该攻击游戏与 G_3 不同之处在于 *MAC* 函数的输出被等长的随机值替代, 在这里将 *MAC* 函数看作黑盒, 因此:

$$|\Pr[S_4] - \Pr[S_3]| \leqslant (q_s + q_e) Adv^{MAC}(A) \tag{6}$$

Game G_5: 该攻击游戏与 G_4 不同之处在于模拟了 PS 签名方案, 将签名的输出替换为从方案范围内选取的一个均匀随机值. 敌手 A 可以访问签名和验证预言机, 在这里我们将 PS 签名方案看作黑盒, 因此:

$$|\Pr[S_5] - \Pr[S_4]| \leqslant (q_s + q_e) Adv^{\mathrm{PS}}(A) \tag{7}$$

Game G_6: 该攻击游戏与 G_5 不同之处在于避免敌手 A 幸运猜测出验证值 B 和密钥 K 的可能, 只有当 A 正确猜中 B 或 K 值且没有进行相应查询时才会发生这种情况. 设 *MAC* 函数输出长度为 l_2, K 值长度为 l_3, 因此:

$$|\Pr[S_6] - \Pr[S_5]| \leqslant \frac{q_s}{2^{l_2}} + \frac{q_s}{2^{l_3}} \tag{8}$$

Game G_7: 该攻击游戏与 G_6 不同之处在于考虑 *Corrupt* 查询, A 可以在 *Corrupt* 查询的帮助下获得用户口令或生物特征, 其概率分别用 $\Pr[AskWithCorr1]$ 和 $\Pr[AskWithCorr2]$ 表示.

1) A 查询 $Corrupt(\prod_U^i, 1)$, 表明用户口令被泄露, 假设口令遵循 Zipf 分布[32], 比传统的均匀分布更接近现实, 则:

$$\Pr[AskWithCorr1] \leqslant C' \cdot q_s^{s'} \tag{9}$$

其中, C' 和 s' 是取决于口令数据集的常数.

2) A 查询 $Corrupt(\prod_U^i, 2)$, 表明用户生物密钥被泄露, 生物特征由模糊提取器处理, 在此处模拟模糊提取器预言机. 假设模糊因子与口令拥有相同的字典空间, 设 $H_{\min}$ 表示模糊因子的最小熵, 随机变量 X 的最小熵可以定义为 $H_{\min}(X) = -\log_2 \max_{x \in X} \Pr[x]$. 因此, $\Pr[x] \leqslant 1/2^{H_{\min}(X)}, x \in X$. 此外, 设模糊提取器错误识别的概率为 p_{false}, 因此:

$$\Pr[AskWithCorr1] \leqslant (q_s + q_e) \cdot \max\left\{\frac{1}{|D|}, \frac{1}{2^{H_{\min}(X)}} + p_{\mathrm{false}}\right\} \tag{10}$$

所以有:

$$|\Pr[S_7] - \Pr[S_6]| \leqslant C' \cdot q_s^{s'} + (q_s + q_e) \cdot \max\left\{\frac{1}{|D|}, \frac{1}{2^{H_{\min}(X)}} + p_{\mathrm{false}}\right\} \tag{11}$$

Game G_8: 该攻击游戏与 G_7 不同之处在于如果 A 在 $<00\|T'\|S\|K>$ 或 $<01\|T'\|S\|K\|\sigma''>$ 上进行一次哈希

查询, 则该游戏终止. 用私有预言机 H' 代替随机预言机 H:

$$K' = H'(00\|T'\|S),$$

$$K'' = H'(00\|T'\|S\|\sigma'').$$

因此, K' 和 K'' 的值独立于 K, 除非事件 $AskH_8$ 发生, 否则 G_8 和 G_7 无法区分.

$$|\Pr[S_8]-\Pr[S_7]| \leqslant \Pr[AskH_8] \tag{12}$$

在 G_8 中, 会话密钥是用 A 不知道的私有哈希预言机计算的, 因此 $\Pr[S_8]=1/2$.

Game G_9: 在该攻击游戏中模拟执行 Diffie-Hellman 问题. 给定一个 CDH 实例 $\varphi=(A,B)$, 假设敌手能够通过预言查询得到 K, 则认为敌手能够解决 CDH 难题. $AskH_9$ 表示敌手 A 已经在 $<01\|T'\|S\|K\|\sigma''>$ 上查询了随机哈希预言机 H. 通过在 L_A 列表中随机抽取, 可以得到概率为 $1/q_h$ 的 DH 秘密值 $<T',(A_1,A_2,A_3),\mathrm{CDH}(T',\{A_i\}_{i=1}^3)>$. 然后在 L_A 和 L_B 列表中找到值 n_1,n_2,n_3 和 s, 使 $A_1=\tilde{g}^s \bmod p, A_2=\tilde{Y}_1^s \bmod p, A_3=\tilde{Y}_2^s \bmod p, T'=\tilde{g}^{n_1}\tilde{Y}_1^{n_2}\tilde{Y}_2^{n_3} \bmod p$ 即 $K=\mathrm{CDH}(T',\{A_i\}_{i=1}^3)=\mathrm{CDH}(\tilde{g}^{n_1}\tilde{Y}_1^{n_2}\tilde{Y}_2^{n_3},\tilde{g}^s\tilde{Y}_1^s\tilde{Y}_2^s)$.

由于并不具备计算 K 的条件, 因此在查询预言机时所有的返回值均为随机值. 因此:

$$\Pr[AskH_8]=\Pr[AskH_9] \leqslant q_h Adv^{\mathrm{CDH}}(A) \tag{13}$$

综上所述, 在公式 (2)–公式 (13) 的联合约束下, 设 $l=\min\{l_0,l_1,l_2,l_3\}$, 敌手优势为:

$$Adv_{\mathrm{UC\text{-}2FAKA}}^{\mathrm{AKA}}(A) \leqslant \frac{q_h^2+2nq_s(nq_s-1)+4q_s}{2^l}+\frac{(q_s+q_e)^2}{p}$$
$$+2(q_s+q_e)\left(Adv^{MAC}(A)+Adv^{\mathrm{PS}}(A)+\max\left\{\frac{1}{D},\frac{1}{2^{H_{\min}(X)}}+p_{\mathrm{false}}\right\}\right)+2C'\cdot q_s^{s'}+2q_hAdv^{\mathrm{CDH}}(A).$$

4.2 针对可能的攻击进行分析

(1) 抵御内部人员攻击

在本文协议的注册阶段, 用户向 IDP 发送口令和生物特征构造的双因子承诺 $C=g^rY_1^{\alpha}Y_2^{\beta} \bmod p$ 以及 $s_2=\alpha\cdot e+t_2 \bmod q$ 和 $s_3=\beta\cdot e+t_3 \bmod q$, 并没有直接发送口令和生物特征明文或是其哈希值 α 和 β. 由于 s_2 和 s_3 中的 t_2 和 t_3 是 IDP 未知的随机数, 且根据离散困难问题假设, 内部人员很难从等式 s_2 和 s_3 以及承诺 C 中计算出口令或生物特征. 同理, 在认证阶段, 内部人员也很难根据用户向 SP 发送的随机凭证 $\sigma''=(g^{uv},(X\cdot g^tY_1^{\alpha}Y_2^{\beta})^{uv})$ 以及 $<s_2',s_3'>$ 来恢复用户的隐私信息. 因此, 本文协议能够抵御内部人员攻击.

(2) 抵御离线口令猜测攻击

在本文提出的协议中, 假设敌手能够窃听公共信道上的所有消息, 并且能得到所有相关公共参数. 从以下两方面分析离线口令猜测攻击: 1) 当用户设备丢失时, 敌手可以获得存储在设备中的凭证值 $\sigma=(g^u,(XY_1^{\alpha}Y_2^{\beta})^u)$, 以及公共参数 g,Y_1,Y_2. 由于敌手不知道 pwd 值, 因此无法直接计算出 $\alpha=H(pwd)$. 假设敌手可以从口令字典中猜测到正确的候选口令, 在不知道 IDP 私钥、随机数 u 和生物密钥 R 的情况下, 即使敌手选择了错误的口令, 也可能计算出相同的凭证值, 因此无法猜测出用户口令. 2) 当敌手通过窃听以往会话拿到多个口令的相关值 s_2' 和参与 s_2' 计算的随机数 b. $s_2'=\alpha'\cdot b+n_2 \bmod q$, 在该等式中, 除口令的哈希值 α' 外, n_2 也是未知的, 因此无法直接计算出 α'. 并且每次通信中随机数都会发生变化, 即 s_2',b,n_2 在每次会话中都会更新, 因此也无法通过窃听多个等式计算出 α'.

综上, 本文提出的方案能够抵抗离线口令猜测攻击.

(3) 抵御重放攻击

当有敌手从早期会话中获取经过身份验证的消息并在新会话中使用来扮演合法的用户角色时, 就会发生重放攻击. 假设敌手通过某个公共信道获得了上一会话的验证消息 $<b,A_1,A_2,A_3,B>$, 则用户通过计算 K' 来验证 $B?=MAC_{K'}(N)$. 由于 N 和 B 在每次会话中都会更新, 无法在当前会话期间验证旧的 B 值. 同理, 若敌手向 SP 重放 $<\sigma'',T',N>$, 在当前会话中也不能验证旧的 σ''. 因此可以抵御重放攻击.

(4) 抵御用户伪装攻击

敌手想要伪装成用户达到认证成功的目的. 在注册过程中, IDP 将盲化后的凭证 $\sigma' = (g^u, (XC)^u)$ 发送给用户, 由用户来执行去盲操作 $\sigma = (\sigma'_1, \sigma'_2/\sigma'^r_1)$, 即只有构造承诺 C 、知道随机数 r 的用户才能执行去盲操作, 因此敌手无法实现用户伪装攻击. 在认证过程中, 假设敌手捕获了盲化后的凭证 σ''. 当敌手与 SP 执行后续协议时, 需要构造消息 $< s'_1, s'_2, s'_3 >$ 来通过认证. 敌手在不知道用户认证因子的情况下, 无法计算出能通过认证的消息. 因此, 本文方案可以抵抗用户伪装攻击.

(5) 抵御服务提供商伪装攻击

敌手想要伪装成服务提供商, 需要构造消息 $< b, A_1, A_2, A_3, B >$ 与用户进行认证. 由于敌手不知道服务提供商的私钥 s, 无法正确计算 A_1, A_2, A_3, 即敌手构造的消息将无法通过认证. 因此, 本文方案可以抵抗服务提供商伪装攻击.

4.3 安全属性分析

(1) 不可链接性

给定从用户登录中获取的信息, 不可链接性确保 SP 无法区分任何两次登录中收到的凭证是来自同一用户还是不同用户. 在本文方案的认证过程中, 用户选择随机数 t 和 v 将凭证进行随机化处理 $\sigma'' = (\sigma''_1, \sigma''_2) = (\sigma'^v_1, (\sigma_2\sigma'^t_1)^v) = (g^{uv}, (X \cdot g^t Y_1^{\alpha} Y_2^{\beta})^{uv})$, 使每次登录的 σ'' 都不相同, SP 无法区分收到的凭证. 因此该协议实现了 SP 对用户登录的不可链接性.

(2) 不可跟踪性

在注册过程中的不可跟踪性确保 IDP 无法通过保留任意两次或多次签名跟踪到请求签名的用户. IDP 发送出对用户承诺的盲签名 $\sigma' = (\sigma'_1, \sigma'_2) = (g^u, (XC)^u)$ 后, 必须利用该签名以及曾经签名过程中保留下的数据求解用户的秘密值 α 和 β, 才能够将该签名消息同自己先前的某次签名行为链接起来进而对用户的信息进行跟踪. 根据离散对数困难问题假设, 计算出 α 和 β 是不可行的, 因此本文方案满足 IDP 对用户的不可跟踪性.

在认证过程中的不可跟踪性确保敌手无法识别同一用户发起的任何两次过去的协议运行. 在认证阶段, 用户与 SP 之间传递的所有消息 $< b, A_1, A_2, A_3, B >$, $< \sigma'', T', N >$ 和 $< s'_1, s'_2, s'_3 >$ 均包含随机数或有随机数参与其运算, 会话中的所有消息都是动态更新而非不变的, 敌手无法通过在不同会话中捕获到的消息跟踪用户. 因此所提出的方案实现用户不可跟踪性.

(3) 双向认证

在用户与 SP 交互过程中, 首先, 用户发送凭证 σ'' 给 SP, 并通过零知识证明向 SP 证明凭证中的口令和生物密钥, SP 通过双线性映射来验证凭证的真实性和完整性, 以此实现用户认证. 其次, 用户发送时鲜值 N 和辅助承诺 T' 给 SP, SP 使用私钥 s 与 T' 生成密钥 K, 用 K 派生出的 K' 对时鲜值 N 加密返回给用户, 用户使用 SP 公钥生成 K, 通过派生 K' 验证密文来实现对 SP 的认证. 综上所述, 用户与 SP 实现了双向认证.

(4) 已知会话密钥安全

通信双方完成相互认证后, 需要建立会话密钥来保证后续通信数据的保密性和完整性. 在该协议中, 用户和 SP 都建立会话密钥 $K'' = H'(00||T'||S||K||\sigma'')$, 并且 K'' 对于每个会话都是独立、不同的. 因此, 尽管敌手能获得用户与 SP 之前建立的会话密钥, 也无法在以后的会话中使用它, 保证了已经会话密钥安全.

(5) 可扩展性

可扩展性表示协议能够根据认证场景的不同, 实现认证因子的灵活变动, 比如从双因子认证协议扩展到三因子认证协议. 协议利用承诺 $C = g^r Y_1^{\alpha} Y_2^{\beta} \bmod p$ 实现了口令和生物特征的组合认证, 当协议需要扩展到三因子认证时, 用户可以通过构造承诺 $C = g^r Y_1^{\alpha} Y_2^{\beta} Y_3^{\theta} \bmod p$ 来实现. 因此, 本协议能够实现可扩展性.

(6) 多因子安全

多因子安全指敌手在获取多个认证因子 (非全部) 的情况下, 不能计算出未知因子来通过身份认证, 从而保证通信的机密性. 在本文认证过程中, 需要用户构造关于口令因子 α 和生物特征因子 β 的消息 $< s'_1, s'_2, s'_3 >$ 发送给 SP 进行验证. 由第 4.2 节分析可得, 本文协议可以防止离线猜测攻击. 假设用户的生物特征被泄露, 敌手无法通过

离线猜测获得用户口令, 从而无法构造正确的 $<s_2'>$; 当用户的口令被泄露, 根据离散困难问题假设, 敌手无法通过保存在用户本地的消息计算出生物特征, 从而无法计算 $<s_3'>$. 因此, 敌手在只知道其中一个认证因子的情况下无法构造出正确的消息通过认证, 保证了协议的双因子安全. 同理, 当协议扩展为多因子认证时, 敌手无法通过保存在用户本地的信息离线出其他认证因子, 从而无法构造正确的认证消息, 保证了多因子安全.

4.4 安全特性对比

本节将协议 UC-2FAKA 与同类型协议进行安全特性对比, 以证明所提出协议在安全性上的优势. 如表 1 所示, 列出了协议 UC-2FAKA 与现有的多因子认证方案文献 [5] 和文献 [25] 之间的安全特性对比.

通过对比发现: 首先, 在文献 [5] 的注册阶段, 用户将口令和生物特征模板以明文形式发送给 IDP 来构造凭证, 不能抵御内部人员攻击; 在认证阶段, 用户直接将身份令牌发送给 SP, 多个 SP 能够根据身份令牌链接同一用户, 不能实现对用户的不可链接性; 且文献 [5] 中用户未验证 SP 身份, 不能实现其所声称的双向认证. 其次, 文献 [25] 仅考虑用户认证, 不能实现相互认证和已知会话密钥安全. 最后, 文献 [5,25] 中的认证因子是固定的, 均不能实现认证因子可扩展性.

表 1 协议 UC-2FAKA 安全特性对比

安全特性	UC-2FAKA	文献[5]	文献[25]
抵御内部人员攻击	√	×	√
抵御离线口令猜测攻击	√	√	√
抵御重放攻击	√	√	√
抵御伪装攻击	√	√	√
不可链接性	√	×	√
不可跟踪性	√	√	√
已知会话密钥安全	√	√	×
双向认证	√	×	×
可扩展性	√	×	×
多因子安全	√	√	√

注: “√”表示方案满足该安全特性, “×”表示方案不满足该安全特性

本文协议 UC-2FAKA 利用随机化凭证和零知识证明技术, 确保 SP 对用户凭证的不可区分, 保证了用户身份信息的隐私性和 SP 对用户的不可链接性; 协议 UC-2FAKA 采用以用户为中心的架构, IDP 不参与认证过程, 无法链接用户访问的 SP; 协议 UC-2FAKA 基于承诺形式组合认证因子, 实现了可扩展性.

5 协议实现与性能评估

本节首先介绍了协议实现的具体细节, 证明了所提出协议的可行性. 然后将本节将所提出协议与现有的认证方案文献 [5] 和文献 [25] 进行安全性对比、通信开销以及计算开销对比, 最终得出所提出协议在综合性能上具有较大优势.

5.1 实现环境

本文实验使用的笔记本电脑 (SP) 配置为 Lenovo、I5-07200 处理器、2.50 GHz CPU、8 GB 内存、Windows 10 系统. 移动客户端配置为 HUAWEI nova3、HiSilicon Kirin 970 处理器、6 GB 内存、Android 9.0. 实验基于 JPBC(2.0) 库, 使用 Java 和 Socket 实现了本文协议中的全部算法. 为和其他方案进行比较, 本文在相同开发环境下运行了所提出的方案、文献 [5] 和文献 [25] 中的相关密码原语, 测试出每一个具体操作的运行时间, 如后文表 2 所示, 最终结果是对 1 000 次运行总时间取平均值操作所得.

5.2 计算开销分析

在初始化阶段, 生成公共参数和 IDP 公私钥共耗时约 16 ms. 在注册阶段, 用户首先输入口令并获取生物密钥 R, 随后对口令和密钥进行哈希并构造关于口令和生物密钥的承诺 C 耗时约 110 ms. 用户与 IDP 执行零知识证明

协议的时间为约 200 ms. 验证成功后, IDP 生成用户凭证并发给用户所用时间约为 25 ms. 注册阶段总耗时约 335 ms.

表 2 给定运算操作的运行时间 (ms)

符号	描述	用户	SP
T_{H}	哈希(SHA-1)操作所用时间	0.061	0.009
T_{sd}	对称解密所用时间	0.092	0.022
$T_{t\text{-exp}}$	群 G_T 上指数运算时间	5.904	1.218
$T_{1\text{-exp}}$	群 G_1 上指数运算时间	22.134	12.642
$T_{t\text{-mul}}$	群 G_T 上乘法运算时间	0.062	0.019
$T_{1\text{-mul}}$	群 G_1 上乘法运算时间	23.147	12.527
T_{bp}	双线性对的运算时间	36.375	7.253
T_{PBKDF2}	PBKDF2生成密钥时间	24.124	12.012
$T_{\mathrm{RSA\text{-}Verify}}$	RSA验签	5.135	1.153
T_{mac}	消息认证码	0.071	0.017
T_{FE}	模糊提取器验证时间	7.012	0.826
T_{extract}	文献[5]的生物模板提取	13 825	—
T_{Predicte}	文献[5]的生成标签	120	—

注: “—”表示不执行该操作

在认证阶段, 用户首先对存储的凭证进行盲化处理, 共耗时约 90 ms, 随后用户与 SP 执行签名的知识证明协议来验证凭证的真实性, 共耗时约为 210 ms, 认证阶段总耗时约 300 ms. 从实验结果来看, 协议主要时间消耗来源于双线性对和指数运算.

如表 3 所示, 将本文协议与文献 [5]、文献 [25] 中的方案进行计算开销对比, 可以看出本文协议与文献 [5]、文献 [25] 对比有着较大的开销优势.

表 3 计算开销时间对比 (ms)

方案	用户	SP	总用时
UC-2FAKA	$2T_{\mathrm{H}}+6T_{1\text{-exp}}+3T_{1\text{-mul}}+T_{\mathrm{FE}}+T_{\mathrm{mac}}$ =209.45	$2T_{\mathrm{H}}+4T_{1\text{-exp}}+5T_{t\text{-exp}}+4T_{t\text{-mul}}+6T_{\mathrm{bp}}+T_{\mathrm{mac}}$ =100.287	309.737
文献[5]	$4T_{1\text{-exp}}+2T_{1\text{-mul}}+T_{\mathrm{PBKDF2}}+T_{\mathrm{sd}}+T_{\mathrm{extract}}+T_{\mathrm{Predicte}}$ =14 104.046	$7T_{1\text{-exp}}+3T_{1\text{-mul}}+T_{\mathrm{RSA\text{-}Verify}}$ =127.228	14 231.274
文献[25]	$3T_{\mathrm{H}}+12T_{1\text{-exp}}+4T_{1\text{-mul}}+T_{\mathrm{sd}}+3T_{\mathrm{bp}}+2T_{t\text{-mul}}+3T_{t\text{-exp}}$ =487.37	$7T_{1\text{-exp}}+3T_{1\text{-mul}}+5T_{t\text{-exp}}+4T_{t\text{-mul}}+5T_{\mathrm{bp}}+T_{\mathrm{sd}}$ =168.528	665.898

5.3 通信开销分析

为方便比较, 依据文献 [25], 假设 UC-2FAKA 协议中循环群 G_1, G_2 和 G_T 中元素长度均为 1 024 bit, q 是 160 bit, 口令和生物密钥进行 SHA-1 哈希后得到的位数是 160 bit, MAC 加密输出位数是 160 bit. 关于上述几个方案的通信开销如下.

在文献 [5] 的认证过程中, 传输的消息包括 $<IDT,d>$, $<e,a,b>$ 和 $<u,v>$, 其中随机数长度为 160 bit, 哈希后的长度为 160 bit, 总通信开销为 1024×8+160×3=8672 bit.

在文献 [25] 的认证过程中, 服务器向用户发送 $M_1=<Z,\sigma_s>$, 其中 Z 是循环群 G_1 上的元素为 1 024 bit, σ_s 是基于服务器私钥 y 的 Schnorr 签名, 大小为 1024+320=1344 bit. 用户向服务器发送 $M_4=<Nym_u^v,D,Hmac,\pi>$, 大小为 1024×10+160×2+320×3=11520 bit. 总通信开销为 13 888 bit.

在 UC-2FAKA 中, 传输的消息包括 $<\sigma'',T',N>$, $<b,A_1,A_2,A_3,B>$ 和 $<s'_1,s'_2,s'_3>$. 其中群中元素均为 1 024 位, σ'' 的比特数为 1024×2=2048 bit, T' 的比特数为 1 024 bit, 传输的总比特数为 1024×6+160×6=7104 bit.

从上面数据可以看出, 本文协议出于对用户的匿名性和凭证的安全性的考虑, 加入了凭证的随机化以及签名的知识证明, 因此增加了约 5 000 bit 的通信开销. 尽管如此, 本文协议仍比文献 [5] 通信开销略低; 对于达到同样安全水平的协议文献 [25], 所提出协议的通信开销具有较大优势.

6 结 论

针对口令认证安全性低、用户身份管理负担大以及身份隐私泄露风险高等问题, 本文提出一个以用户为中心的双因子认证密钥协商 UC-2FAKA 协议. 该协议由 IDP 对用户发送的承诺进行盲签名来向用户颁发双因子凭证, 并在认证时将盲化后的凭证发送给 SP 进行验证. 首先, 在整个过程中, IDP 和 SP 都无法获取用户的身份和认证信息, 实现对用户的隐私保护, 同时保证了 IDP 和 SP 对用户的不可跟踪性和不可链接性. 其次, 基于凭证构造的特殊性, 该协议可以进行多因子扩展. 最后, 通过对方案进行全面的安全性分析和性能比较, 表明本文协议能够抵御各种已经的攻击并提供更多的安全特性; 且与同类型协议相比, 本文方案在通信开销和计算开销方面更具优势.

References:

[1] Li ZP, Yang Z, Szalachowski P, Zhou JY. Building low-interactivity multifactor authenticated key exchange for industrial Internet of Things. IEEE Internet of Things Journal, 2021, 8(2): 844–859. [doi: 10.1109/JIOT.2020.3008773]

[2] Fett D, Küsters R, Schmitz G. The Web SSO standard openid connect: In-depth formal security analysis and security guidelines. In: Proc. of the 30th IEEE Computer Security Foundations Symp. Santa Barbara: IEEE, 2017. 189–202. [doi: 10.1109/CSF.2017.20]

[3] Hammann S, Sasse R, Basin D. Privacy-preserving OpenID connect. In: Proc. of the 15th ACM Asia Conf. on Computer and Communications Security. Taipei: ACM, 2020. 277–289. [doi: 10.1145/3320269.3384724]

[4] Zhang ZY, Król M, Sonnino A, Zhang LX, Rivière E. EL PASSO: Efficient and lightweight privacy-preserving single sign on. Proc. on Privacy Enhancing Technologies, 2021, 2021(2): 70–87. [doi: 10.2478/popets-2021-0018]

[5] Gunasinghe H, Bertino E. PrivBioMTAuth: Privacy preserving biometrics-based and user centric protocol for user authentication from mobile phones. IEEE Trans. on Information Forensics and Security, 2018, 13(4): 1042–1057. [doi: 10.1109/TIFS.2017.2777787]

[6] Murdoch SJ, Abadi A. A forward-secure efficient two-factor authentication protocol. arXiv:2208.02877, 2022.

[7] Bonneau J, Herley C, van Oorschot PC, Stajano F. Passwords and the evolution of imperfect authentication. Communications of the ACM, 2015, 58(7): 78–87. [doi: 10.1145/2699390]

[8] Wang YG. Password protected smart card and memory stick authentication against off-line dictionary attacks. In: Proc. of the 27th IFIP TC 11 Information Security and Privacy Conf. on Information Security and Privacy Research. Crete: Springer, 2012. 489–500. [doi: 10.1007/978-3-642-30436-1_40]

[9] Wang D, Wang P. Offline dictionary attack on password authentication schemes using smart cards. In: Proc. of the 16th Int'l Conf. on Information Security. Dallas: Springer, 2015. 221–237. [doi: 10.1007/978-3-319-27659-5_16]

[10] Wang D, Wang P. Two birds with one stone: Two-factor authentication with security beyond conventional bound. IEEE Trans. on Dependable and Secure Computing, 2018, 15(4): 708–722. [doi: 10.1109/TDSC.2016.2605087]

[11] Wang D, Ma CG, Wu P. Secure password-based remote user authentication scheme with non-tamper resistant smart cards. In: Proc. of the 26th Annual IFIP WG 11.3 Conf. on Data and Applications Security and Privacy XXVI. Paris: Springer, 2012. 114–121. [doi: 10.1007/978-3-642-31540-4_9]

[12] Shirvanian M, Jarecki S, Saxena N, Nathan N. Two-factor authentication resilient to server compromise using mix-bandwidth devices. In: Proc. of the 21st Annual Network and Distributed System Security Symp. San Diego: The Internet Society, 2014. 1–16. [doi: 10.14722/ndss.2014.23167]

[13] Jarecki S, Krawczyk H, Shirvanian M, Saxena N. Two-factor authentication with end-to-end password security. In: Proc. of the 21st IACR Int'l Conf. on Practice and Theory of Public-key Cryptography. Rio de Janeiro: Springer, 2018. 431–461. [doi: 10.1007/978-3-319-76581-5_15]

[14] Wei FS, Zhang G, Ma JF, Ma CG. Privacy-preserving multi-factor key exchange protocol in the standard model. Ruan Jian Xue Bao/Journal of Software, 2016, 27(6): 1511–1522 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5001.htm [doi: 10.13328/j.cnki.jos.005001]

[15] Zhang R, Xiao YT, Sun SZ, Ma H. Efficient multi-factor authenticated key exchange scheme for mobile communications. IEEE Trans. on

Dependable and Secure Computing, 2019, 16(4): 625–634. [doi: 10.1109/TDSC.2017.2700305]

[16] Bellare M, Pointcheval D, Rogaway P. Authenticated key exchange secure against dictionary attacks. In: Proc. of the 2000 Int'l Conf. on the Theory and Application of Cryptographic Techniques. Bruges: Springer, 2000. 139–155. [doi: 10.1007/3-540-45539-6_11]

[17] Zhang ZF, Wang YC, Yang K. Strong authentication without tamper-resistant hardware and application to federated identities. In: Proc. of the 2020 Network and Distributed Systems Security (NDSS) Symp. San Diego: The Internet Society, 2020. 23–26. [doi: 10.14722/ndss.2020.24462]

[18] Alaca F, van Oorschot PC. Comparative analysis and framework EVAluating web single sign-on systems. ACM Computing Surveys, 2021, 53(5): 112. [doi: 10.1145/3409452]

[19] Guo CQ, Lin JQ, Cai QW, Wang W, Li FJ, Wang QX, Jing JW, Zhao B. UPPRESSO: Untraceable and unlinkable privacy-preserving single sign-on services. arXiv:2110.10396, 2021.

[20] Machani S, Philpott R, Srinivas S, Kemp J, Hodges J. FIDO UAF architectural overview. 2018. https://media.fidoalliance.org/specs/fido-uaf-v1.2-id-20180220/FIDO-UAF-COMPLETE-v1.2-id-20180220.pdf

[21] Wang D, Li WT, Wang P. Crytanalysis of three anonymous authentication schemes for multi-server environment. Ruan Jian Xue Bao/Journal of Software, 2018, 29(7): 1937–1952 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5361.htm [doi: 10.13328/j.cnki.jos.005361]

[22] Wan T, Liu ZX, Ma JF. Authentication and key agreement protocol for multi-server architecture. Journal of Computer Research and Development, 2016, 53(11): 2446–2453 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2016.20150107]

[23] Amin R. Cryptanalysis and efficient dynamic ID based remote user authentication scheme in multi-server environment using smart card. Int'l Journal of Network Security, 2016, 18(1): 172–181.

[24] Reddy AG, Yoon EJ, Das AK, Odelu V, Yoo KY. Design of mutually authenticated key agreement protocol resistant to impersonation attacks for multi-server environment. IEEE Access, 2017, 5: 3622–3639. [doi: 10.1109/ACCESS.2017.2666258]

[25] Mir O, Roland M, Mayrhofer R. DAMFA: Decentralized anonymous multi-factor authentication. In: Proc. of the 2nd ACM Int'l Symp. on Blockchain and Secure Critical Infrastructure. Taipei: ACM, 2020. 10–19. [doi: 10.1145/3384943.3409417]

[26] Schnorr CP. Efficient signature generation by smart cards. Journal of Cryptology, 1991, 4(3): 161–174. [doi: 10.1007/BF00196725]

[27] Fiat A, Shamir A. How to prove yourself: Practical solutions to identification and signature problems. In: Proc. of the 1987 Conf. on the Theory and Application of Cryptographic Techniques. Santa Barbara: Springer, 1987. 186–194. [doi: 10.1007/3-540-47721-7_12]

[28] Pedersen TP. Non-interactive and information-theoretic secure verifiable secret sharing. In: Proc. of the 1992 Annual Int'l Cryptology Conf. on Advances in Cryptology. Santa Barbara: Springer, 1992. 129–140. [doi: 10.1007/3-540-46766-1_9]

[29] Pointcheval D, Sanders O. Short randomizable signatures. In: Proc. of the 2016 Cryptographers' Track at the RSA Conf. on Topics in Cryptology. San Francisco: Springer, 2016. 111–126. [doi: 10.1007/978-3-319-29485-8_7]

[30] Dodis Y, Reyzin L, Smith A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: Proc. of the 2004 Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Interlaken: Springer, 2004. 523–540. [doi: 10.1007/978-3-540-24676-3_31]

[31] Pointcheval D, Wang GL. VTBPEKE: Verifier-based two-basis password exponential key exchange. In: Proc. of the 2017 ACM on Asia Conf. on Computer and Communications Security. Abu Dhabi: ACM, 2017. 301–312. [doi: 10.1145/3052973.3053026]

[32] Wang D, Wang P. On the implications of Zipf's law in passwords. In: Proc. of the 21st European Symp. on Research in Computer Security. Heraklion: Springer, 2016. 111–131. [doi: 10.1007/978-3-319-45744-4_6]

附中文参考文献:

[14] 魏福山, 张刚, 马建峰, 马传贵. 标准模型下隐私保护的多因素密钥交换协议. 软件学报, 2016, 27(6): 1511–1522. http://www.jos.org.cn/1000-9825/5001.htm [doi: 10.13328/j.cnki.jos.005001]

[21] 汪定, 李文婷, 王平. 对三个多服务器环境下匿名认证协议的分析. 软件学报, 2018, 29(7): 1937–1952. http://www.jos.org.cn/1000-9825/5361.htm [doi: 10.13328/j.cnki.jos.005361]

[22] 万涛, 刘遵雄, 马建峰. 多服务器架构下认证与密钥协商协议. 计算机研究与发展, 2016, 53(11): 2446–2453. [doi: 10.7544/issn1000-1239.2016.20150107]

杨雪(1999－), 女, 博士生, 主要研究领域为多因子认证协议, 后量子密码学.

王金花(1995－), 女, 硕士, 主要研究领域为密码协议设计与分析.

刘怡静(1996－), 女, 硕士, 主要研究领域为多因子认证协议.

李兴华(1978－), 男, 博士, 教授, CCF 专业会员, 主要研究领域为无线网络安全, 隐私保护, 数据安全.

姜奇(1983－), 男, 博士, 教授, CCF 高级会员, 主要研究领域为密码协议, 物联网安全.