计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202440468
61(10): 2514−2525, 2024

基于区块链的高效无口令认证方案

柯宇泓[1] 林 超[2] 黄欣沂[2] 伍 玮[3] 陈雨婕[4]
[1](福建师范大学计算机与网络空间安全学院 福州 350117)
[2](暨南大学网络空间安全学院 广州 510632)
[3](香港科技大学(广州)教育科学学院 广州 511455)
[4](福建大数据信息安全建设运营有限公司 福州 350101)
(1163170349@qq.com)

An Efficient Passwordless Authentication Scheme Based on Blockchain

Ke Yuhong[1], Lin Chao[2], Huang Xinyi[2], Wu Wei[3], and Chen Yujie[4]
[1](*College of Computer and Cyber Security*, *Fujian Normal University*, *Fuzhou* 350117)
[2](*College of Cyber Security*, *Jinan University*, *Guangzhou* 510632)
[3](*College of Education Sciences*, *The Hong Kong University of Science and Technology* (*Guangzhou*), *Guangzhou* 511455)
[4](*Fujian Big Data Information Security Construction and Operation Co.*, *Ltd.*, *Fuzhou* 350101)

Abstract In the digital era, data have become a core asset for the functioning of society, and identity authentication credentials are among the most critical and sensitive data elements. Traditional password-based authentication methods require servers to store credential information such as usernames and passwords, which poses a serious risk of data leakage. Passwordless authentication technology based on public-key cryptography replaces traditional passwords with public-private key pairs. Users employ their private keys to compute signatures for authentication information, while servers only store public information like public keys, thus eliminating the issue of servers leaking private key information. However, existing passwordless authentication systems face challenges such as incompatibility across multiple platforms, high latency in online authentication, and the difficulty of recovering private keys when devices are lost. Moreover, the transparency and auditability of these systems need improvement. To address these problems, we propose an efficient, multi-platform compatible, passwordless identity authentication scheme based on blockchain technology. The scheme combines FIDO2 passwordless authentication with blockchain, allowing users to generate and upload multiple account public keys to the blockchain network for public verification by service providers. Through optimizations such as offline account pre-registration, pre-computation of signatures, and on-chain data synchronization, the scheme achieves interoperability, low overhead, and scalability for large-scale user authentication. The scheme also incorporates an encrypted backup mechanism, enabling users to recover backup data using stored encrypted keys even if their devices are lost. Furthermore, the scheme leverages the immutable data storage provided by blockchain, allowing all participants to query the status of authentication authorizations, thus enhancing system transparency. We comprehensively evaluate the security and performance of the proposed scheme. Theoretical analysis and experiments show that the proposed scheme reduces online computational overhead by 89.09% and communication overhead by 85.57% compared with similar schemes while maintaining low-latency responses under high-load conditions.

收稿日期: 2024-05-31; **修回日期**: 2024-07-15
基金项目: 国家自然科学基金项目(U21A20466, 62032005, 62102089, 62272104)
This work was supported by the National Natural Science Foundation of China (U21A20466, 62032005, 62102089, 62272104).
通信作者: 林超(cschaolin@163.com)

Key words identity authentication; FIDO2; blockchain; digital signature; data elements

摘 要 数字化时代，数据已成为社会运转的核心资产，而身份认证凭证是其中最关键且敏感的数据要素之一. 传统的口令认证方法需在服务器存储账号、口令等凭证信息，面临严重的数据泄露隐患. 基于公钥密码体系的无口令认证技术通过公私钥对替代传统口令，用户利用私钥计算认证信息的签名而实现认证，服务器仅存储公钥等公开信息，从而解决了服务器泄露私钥信息的问题. 然而，现有无口令认证系统存在多平台认证不互通、在线认证延迟高、设备丢失私钥难恢复等痛点，且系统的透明度和审计能力有待提高. 为解决这些问题，提出一种基于区块链的多平台兼容的高效无口令身份认证方案. 将 FIDO2 无口令认证技术与区块链技术相结合，允许用户生成并上传多个账户公钥至区块链网络，供服务方公开验证，同时通过账户线下预注册、签名预计算、链上数据同步等优化手段，实现了可互通、低开销、大规模用户适用的身份认证. 还设计了加密备份机制，即使用户设备遗失，也可利用保存的加密密钥恢复备份数据. 此外，利用区块链提供不可篡改的数据存证，所有参与方均可查询认证授权状态，提高了系统透明度. 最后，全面论述和评估了方案的安全性与系统性能，理论分析和实验结果表明，提出方案较同类型方案降低了 89.09% 的在线计算开销和 85.57% 的通信开销，且在高负载测试条件下仍能够保持低延迟的认证响应.

关键词 身份认证；FIDO2；区块链；数字签名；数据要素

中图法分类号 TP391

近年来，随着互联网技术的快速发展和信息化进程的不断深入，数据成为社会运转的核心资产和生产要素. 在海量的数据中，身份认证凭证是关键且敏感的数据要素，它作为识别个人或机构身份的依据，决定了各类数据资源的访问权限和使用权限. 一旦身份认证凭证被盗用或被窃取，将直接导致相应数据资源的非法访问、窃取和破坏. 这不仅严重侵犯个人隐私，更直接影响到各类数据资源的安全性和社会发展的稳定性.

传统的身份认证方法主要依赖口令认证技术，用户通过输入预设的口令来证明其身份的合法性. 然而，在传统的集中式认证架构下，用户的口令认证信息集中存储在后台服务器，这种集中存储的方式为数据安全带来了重大风险. 一旦服务器遭到攻击，攻击者将成功获取大量用户的认证数据. 近年来，服务器受攻击导致大规模口令泄露事件屡见不鲜，例如：在 2022 年知名口令管理服务 LastPass 遭遇的重大安全事件中，数百万用户的口令及敏感信息被非法获取[1]. 这种传统认证模式已逐渐难以满足物联网、区块链、人工智能等新兴技术发展所需的数据安全需求.

随着硬件设备计算能力的显著增强与通信速度的快速提升，基于公钥密码体系的无口令认证技术逐渐盛行. 该技术通过密钥对替代传统口令，用户在硬件设备上生成密钥对，并将公钥上传至服务端，私钥则安全地保存在用户的硬件设备中. 当用户尝试登录时，服务端将发起挑战，用户设备利用私钥计算并回复挑战的签名，服务端通过验证签名的有效性从而确认用户身份.

尽管基于公钥密码体系的无口令认证技术比口令认证技术拥有更高的安全性，但它在实际应用中仍然存在一些不足之处，亟待改进和完善：此类系统通常要求用户需要在每个网站独立注册、认证与撤销，认证过程不互通；用户需随身携带硬件设备，缺乏便携性；用户使用硬件设备认证过程操作繁琐，且签名计算开销较大，导致较高的在线延迟；若用户需要更换硬件设备，迁移和恢复认证凭证将比传统口令认证更加复杂，缺乏通用的迁移和备份机制；若用户一旦丢失存储密钥的硬件设备，则面临密钥无法恢复的风险，导致永久失去某些加密数据或服务的访问权限.

针对上述问题，本文提出一种基于区块链的多平台兼容的高效无口令身份认证方案，主要贡献包括 3 个方面：

1）在保证安全性的前提下，解决现有基于公钥密码体系的无口令认证系统中存在的认证不互通、设备丢失难以恢复等问题，大幅度简化用户在不同服务间的切换和认证流程，加强用户认证的便利性，提供良好的用户体验；

2）在提高系统安全性和用户便利性的同时，方案通过利用账户线下预注册、签名预计算以及链上数据同步机制，显著降低用户在线认证延迟与系统运行的通信和计算开销，确保在高负载情况下也能保持高效、低延迟的认证体验，有效支持大规模用户的认证需求；

3)方案结合区块链技术,将认证授权记录在公开的区块链中,实现了用户身份认证信息的不可篡改性和公开可验证性,所有参与方可以随时查询认证授权状态,为系统的审计和透明度提供强有力的保障.

1 相关工作

在信息安全领域中,身份认证是数据访问控制的关键环节.口令认证仍然为如今最流行的认证方案之一[2-3],这种认证方式对服务提供商来说简单快捷、成熟稳定,但面临公认的弱安全性[4].近年来,国内外学者提出了多种高安全的身份认证方案,以期替代传统的口令认证方式,其中包括多因素认证[5]、单点登录[6]、生物识别技术[7-8]、硬件令牌认证[9-10]、行为认证[11]、图形认证[12]等.然而,这些方案均未能得到广泛的采纳与应用,主要原因在于它们的可部署性受到诸多限制,且用户体验方面也存在明显不足[13].

对此,学术界和工业界不断探求新的无口令认证解决方案.2014年,文献[14]提出了一种无口令通用登录的解决方案,该方案为每个用户颁发一个证书,并将之存储在智能手机应用程序中,用于自动的网站身份认证,但该方案中服务方能够根据特定的公钥识别用户,违背身份认证的不可链接性.2017年,文献[15]开发了n-Auth系统,利用基于公钥密码系统的认证协议,使智能手机能够在网站上自动注册和登录,但该系统不允许用户在验证器丢失后恢复账户;文献[16]于同年提出基于标识符的Android客户端身份认证方案同样存在类似问题.

2019年,由包含苹果公司、谷歌公司、英特尔公司、微软公司等多家身份验证领域相关公司组成的线上快速身份(Fast Identity Online, FIDO)验证联盟正式发布FIDO2协议,其被誉为目前最先进的无口令认证标准协议.迄今为止,已有多种基于FIDO2协议的无口令认证方案被提出,下面对这些相关工作简要回顾,并总结在表1中.2019年,文献[17]基于SIM卡的可信平台模块提出一种安全便捷的FIDO2认证方案,但该方案要求移动服务提供商在其SIM卡上提供支持.2021年,文献[18]提出了FIDO2认证协议的第一个形式化模型,引入了无口令身份认证的概念,该概念在令牌使用常驻密钥的情况下模拟了WebAuthn协议,并且为基于PIN的访问控制引入了安全模型,试图正式定义在令牌与客户端之间执行的CTAP协议.同年,针对专用硬件不可用的情况,文献[19]提出一种使用QR码进行远程FIDO2身份验证的理论框架,但其安全性较弱.2022年,文献[20]提出一种基于电子身份证的无口令认证系统,主要利用电子身份证派生FIDO2密钥,实现了账户恢复的功能.然而,该系统依靠特定的电子身份证,无法在全球范围内通用,且该系统使用较为繁琐,在易用性方面存在较大提升空间.同年,文献[21]提出一种采用证书认证的无口令认证系统,其仅需要用户使用FIDO2密钥完成1次认证,即可通过自动发放证书的方式,代替用户完成后续的不同服务认证.然而,该系统中证书的发放与管理缺乏透明度,且需要用户频繁地与证书颁发机构进行通信以更新证书的有效性,导致性能较低.2023年,文献[22]提出另一种基于电子身份证的无口令认证系统,其对通用性进行了改进,不局限于特定的电子身份证件,还支持其他独立的匿名凭证系统,但该系统未涉及批量撤销处理机制.

Table 1 Comparison of Passwordless Authentication Systems Based on FIDO2

表1 基于FIDO2的无口令认证系统对比

方案	账户恢复	批量撤销	认证互通	高性能
文献[17]	×	×	×	×
文献[19]	×	×	√	×
文献[20]	√	×	×	×
文献[21]	√	√	√	×
文献[22]	√	×	×	×
本文方案	√	√	√	√

注:"√"表示符合所述特性,"×"表示不符合所述特性.

主流科技公司是FIDO2协议的有力推动者,提供了大量的技术支持:在2019年,谷歌公司允许用户在ARM Trustzone环境下的Android Keystore中存储FIDO2密钥,使用户能够使用手机及其PIN码或生物特征作为FIDO2虚拟身份验证器[23];微软公司则利用可信平台模块硬件为Windows系统的身份认证服务提供安全支持;在macOS和iOS平台上,FIDO2可借助苹果公司的Secure Enclave进行密钥管理,并利用FaceID和TouchID作为用户访问控制的手段[24].在2021年,谷歌、苹果及微软等公司提出了基于FIDO2的通行密钥(Passkey)跨设备无口令认证方案,旨在通过云同步多设备FIDO2密钥,实现跨设备无缝身份认证体验,并在设备丢失时提供恢复机制[25-26].然而,这些现有的解决方案局限于其各自的平台,尚未真正实现全平台的密钥同步和身份互通.

综上,基于公钥密码体系的无口令认证系统的研究已经取得了众多的成果,但现有方案大多在隐

私保护、密钥恢复、性能开销等方面存在不足，仍需进一步优化设计.

2 预备知识

2.1 FIDO2 认证协议

FIDO2 认证协议是一种开放的认证标准，旨在提供更安全且用户友好的身份认证解决方案，从根本上减少对传统口令认证的依赖. 该协议允许用户利用生物识别设备、移动设备、硬件安全密钥等各种身份验证器，以无需暴露私钥的方式进行认证请求，极大地降低网络钓鱼和重放攻击的风险.

FIDO2 由 WebAuthn 与 CTAP 两个标准组成. WebAuthn 是一种基于公钥密码学的 Web 身份验证标准，定义了客户端设备与 Web 服务器之间的通信，包括挑战-响应协议[27]. WebAuthn 允许用户使用硬件密钥、手机或其他设备计算认证凭证，而无需输入传统的口令. 这些凭证与用户的设备绑定，且全程驻留在设备内部，从而有效地规避了凭证被盗用的风险. CTAP 定义了用户设备与 USB 硬件密钥、NFC、蓝牙设备等外部验证器之间的通信协议[28]. CTAP 使得外部验证器能够与不同平台的 FIDO2 客户端互通，实现跨平台的无缝体验.

FIDO2 协议设计遵循“隐私保护”原则. 与其他需要收集个人身份信息(如手机号码)的认证方法不同，FIDO2 协议的认证过程中不涉及任何个人身份信息. FIDO2 协议涉及 4 种主要参与方：服务方(如在线服务)、身份验证器(如硬件密钥)、客户端(如网络浏览器或应用程序)和用户.

FIDO2 协议流程遵循挑战-响应协议，如图 1 所示，该流程从用户注册或请求身份验证开始(步骤①②). 随后，服务方生成一个随机挑战，该挑战通过客户端传输到身份验证器(步骤③④). 用户需要对身份验证器进行解锁(步骤⑤)，使其能够对挑战值进行签名. 最后，身份验证器生成签名并将其返回给服务方(步骤⑥⑦). 用户向服务方注册时，用户的身份验证器会产生唯一一对新的非对称密钥对，包括私钥和公钥. 私钥安全保存在用户的身份验证器中，并与该服务方的标识信息绑定，而公钥则被上传到服务方的数据库中. 在后续的认证过程中，服务方使用与认证账户关联的公钥来验证已签名的客户端响应. 在客户端使用私钥对挑战进行签名过程中，证明了其对私钥的控制权，使用户获得进入在线服务的权限. 该协议的安全性已经通过形式化证明[29].

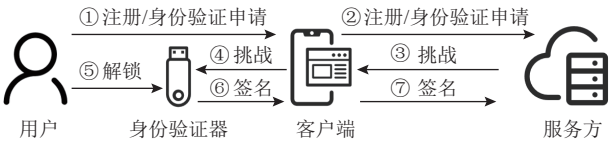


Fig. 1 Simplified identity authentication flow of FIDO2 authentication protocol

图 1 简化后的 FIDO2 认证协议的身份认证流程

2.2 Optimism 二层拓展网络

Optimism 是一种基于以太坊的二层网络扩展解决方案，旨在提供可扩展、低成本且安全的交易处理能力，将以太坊主网络的计算负载分流至二层网络，以缓解以太坊主网络拥塞和高昂燃气(Gas)费用问题，使得以太坊生态系统中的开发者和用户能够以更低的成本和更快的速度进行交易.

该方案的核心在于将多个交易汇总为以太坊网络上的一个批次数据，从而实现交易费用的显著降低和交易速度的大幅度提升. 传统的以太坊交易在被确认之前，需要通过共识机制进行链上验证，这种强一致性机制导致了低吞吐量问题. 而在 Optimism 方案中，交易通过一种称为“乐观汇总”的过程被打包成批次，每个批次中的交易在初始阶段都被网络假定为有效，这种乐观的假设使得交易能够迅速得到确认和执行. 为保证交易的有效性，Optimism 引入了一种称为欺诈证明的机制. 网络中的验证者如果在执行过程中发现某笔交易存在欺诈行为，则可以提出质疑并提供相关证据. 一旦欺诈行为被证实，相关交易将被回滚，并对欺诈者进行惩罚. 这种机制确保 Optimism 在保持高效性的同时，也具备强安全性.

Optimism 的大致工作流程如图 2 所示. 首先 Op-

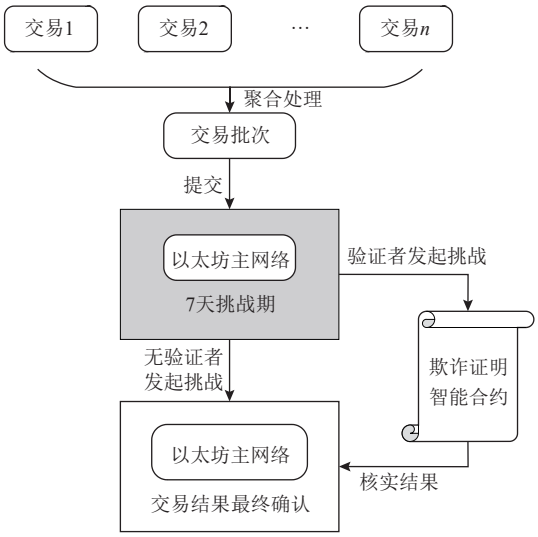


Fig. 2 Simplified Optimism workflow

图 2 简化后的 Optimism 工作流程

timism网络中的多个交易将被聚合处理，聚合后的交易结果(或称为状态更新)以批次的形式提交至以太坊主网络中. 这一过程基于乐观假设，即系统默认提交的结果是有效的，除非在7天的挑战期内，有验证者对结果进行检查并提出挑战. 一旦出现挑战，系统将进入验证阶段，以太坊主网络中的欺诈证明智能合约将负责核实结果的准确性. 根据验证结果，状态更新会被确认或回滚，并且交易结果将最终确定.

3 系统模型与安全模型

3.1 系统模型

本文所提出的系统包含用户、身份验证器、验证服务器、服务方和区块链网络5个实体，如图3所示.

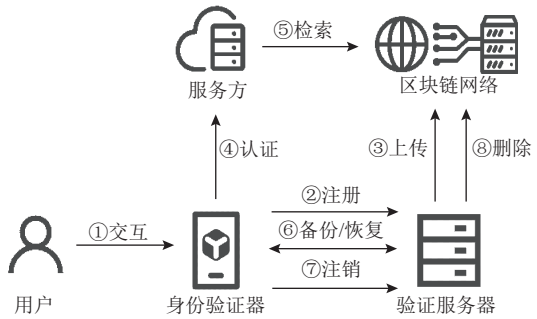


Fig. 3 The system model of our scheme

图3 本文方案的系统模型

1)用户. 该实体为系统服务使用者，通过与身份验证器进行安全交互(步骤①)，实现对服务方的快捷身份认证，以访问所需服务.

2)身份验证器. 该实体是系统的核心安全组件，运行在用户设备上，由用户控制，负责自动生成和管理公私钥对，并向验证服务器注册(步骤②)或注销(步骤⑦)公钥. 身份验证器在用户设备的硬件安全模块中安全存储私钥，以防止私钥泄露. 在身份认证过程中，身份验证器使用私钥对认证请求进行签名，并将签名提交给服务方作为用户的认证凭证(步骤④).

3)验证服务器. 该实体是系统的核心服务提供者，验证服务器在区块链网络中上传(步骤③)和删除(步骤⑧)用户的公钥，并维护它们之间的映射关系. 此外，验证服务器还处理用户数据的备份和恢复请求(步骤⑥).

4)服务方. 该实体是提供在线服务的平台，它们通过检索(步骤⑤)区块链网络中的公钥列表，验证用户提供的认证凭证合法性，进而允许用户登录并使用相关服务.

5)区块链网络. 该实体是整个身份认证系统的记录保证. 验证服务器通过在区块链上部署智能合约管理公钥的上传(步骤③)和删除(步骤⑧)，从而建立一个公开可检索的公钥列表. 在本系统中，区块链网络采用以太坊二层扩展网络Optimism作为底层技术框架，通过Optimistic Rollup优化技术实现链上数据高效计算和低费用存证，从而提高整体系统的性能和运行效率.

3.2 安全模型

本文将从攻击者的攻击主体出发，考虑3类不同的攻击者：

1)服务方攻击者$\mathcal{A}_1$. 攻击者$\mathcal{A}_1$可以通过腐化服务方，获取用户提供的账户公钥与签名，并试图利用获取到的信息通过其他未被腐化服务方的认证.

2)验证服务器攻击者$\mathcal{A}_2$. 攻击者$\mathcal{A}_2$可以通过控制或窃取验证服务器数据，获取用户上传的认证公钥、账户公钥以及备份数据，并试图利用获取到的信息伪装成为合法用户，以通过服务方的认证.

3)用户设备攻击者$\mathcal{A}_3$. 攻击者$\mathcal{A}_3$可以物理控制或窃取用户设备，并试图使用用户设备中的身份验证器通过服务方的认证.

除了各自能够直接获取的信息之外，这3类攻击者还可通过区块链网络查询到公开的公钥列表信息.

在系统的所有流程中，用户均通过身份验证器与验证服务器和服务方进行交互认证. 因此，为保障安全，用户在启动身份验证器时，需首先通过用户设备中的指纹识别、人脸识别等生物识别技术解锁身份验证器，方可进行安全交互. 身份验证器与验证服务器及服务方之间的所有通信均通过安全的TLS信道进行，确保数据传输过程中的保密性和完整性.

4 方案设计

本节主要详细介绍无口令认证方案的设计，该方案分为用户准备、用户认证、用户数据备份及用户注销4个部分.

1)用户准备部分包含用户初始化与用户账户预注册. 在此部分中，用户将通过完成对验证服务器的认证，得到被授权的认证公私钥对，并用其完成用户账户公私钥对的预注册.

2)用户认证部分包含用户向服务方注册和用户向服务方登录. 在此部分中，用户将利用预注册的账户公私钥对完成对服务方的注册认证与登录认证.

3)用户数据备份部分包含用户备份身份验证器

数据和用户恢复备份身份验证器数据. 在此部分中，用户可以将身份验证器中的密钥数据加密后备份至验证服务器，以备未来在原设备丢失、损坏的情况下能够在新设备上安全便携地恢复所有备份数据.

4)在用户注销部分中，用户可一次性注销先前所有认证过的认证公钥和账户公钥，防止泄露的密钥数据被攻击者利用. 以下为方案的详细介绍. 其中符号描述如表 2 所示.

Table 2 Symbols and Their Descriptions

表 2 符号及其描述

符号	描述	符号	描述
PK_{au}	用户认证公钥	UID	用户名
SK_{au}	用户认证私钥	SN	会话号
PK_{ac}	用户账户公钥	S_i	服务方 i
SK_{ac}	用户账户私钥	Enc	加密
σ	签名	$sKey$	秘密密钥
map	映射	$eKey$	对称密钥

4.1 用户初始化

在用户首次使用身份认证系统前，需通过 FIDO2 协议对验证服务器进行认证，以启动初始化流程. 用户完成 FIDO2 认证后，初始化流程将遵循 2 个步骤：

1)身份验证器将通过设备的硬件安全模块(如 TrustZone, Secure Enclave 等)的密钥管理机制，在其硬件防护的可信执行环境中，生成认证公私钥对(PK_{au},SK_{au}). SK_{au}将被安全存储在硬件安全模块中，以防止密钥泄露.

2)身份验证器向验证服务器提交初始化申请，并发送PK_{au}至验证服务器. 验证服务器通过智能合约将PK_{au}上传至区块链网络中的认证列表$auList$，表明该PK_{au}已得到授权.

4.2 用户账户预注册

为支持将来向多个服务方注册，身份验证器会预生成多组账户公私钥对. 该过程在用户设备的可信执行环境中进行，确保私钥的安全存储，具体流程为：

1)身份验证器在硬件安全模块中生成账户公私钥对(PK_{ac},SK_{ac})，并将其存储在模块内.

2)身份验证器使用在用户注册阶段获取的认证私钥SK_{au}，对账号公钥PK_{ac}进行签名，得到$\sigma(PK_{ac},SK_{au})$.

3)身份验证器将PK_{au},PK_{ac}及$\sigma(PK_{ac},SK_{au})$发送至验证服务器，申请注册该PK_{ac}.

4)验证服务器将根据区块链网络中的认证列表$auList$验证PK_{au}的合法性，若合法，则进一步通过PK_{au}验证$\sigma(PK_{ac},SK_{au})$正确性.

5)若$\sigma(PK_{ac},SK_{au})$正确，则证明该PK_{ac}由持有SK_{au}的用户发布，验证服务器将通过智能合约上传PK_{ac}至存储在区块链网络中的账户列表$acList$，同时在本地的账户公钥映射表$acmap$记录PK_{au}与PK_{ac}的映射关系.

身份验证器将在用户设备空闲时间内多次执行上述操作，预先注册多组账户公钥PK_{ac}，确保用户未来向服务方注册时，能够及时利用预注册的PK_{ac}进行注册，从而降低注册过程中的延迟.

4.3 用户向服务方注册

用户可使用已注册但未使用的账户公私钥对(PK_{ac},SK_{ac})向服务方进行注册，具体流程如图 4 所示.

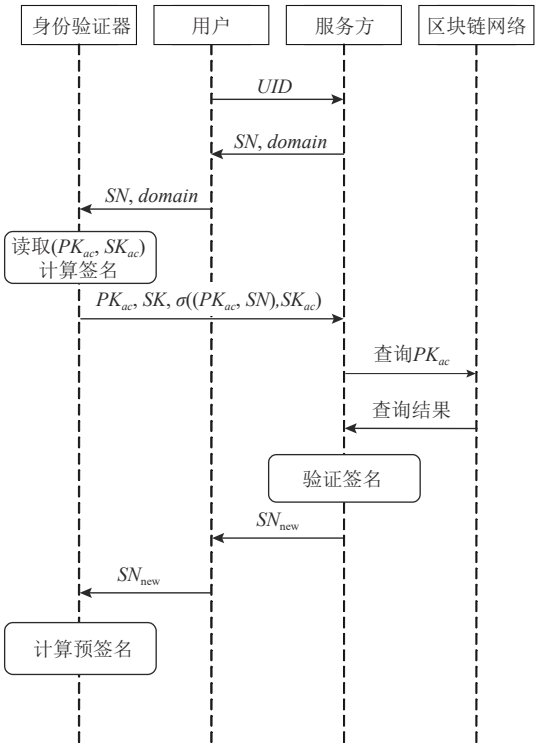


Fig. 4 Flow chart of user registrating with the service provider

图 4 用户向服务方注册流程图

1)用户向服务方S_i发送注册申请，提交用户名UID.

2)服务方S_i收到用户的注册申请后，利用伪随机数生成算法生成会话号SN，并将SN与服务方S_i的域名$domain$返回至用户.

3)用户收到SN与$domain$后，调用身份验证器并传入SN与$domain$.

4)身份验证器读取存储在硬件安全模块中未使用的(PK_{ac},SK_{ac})，并用SK_{ac}对(PK_{ac},SN)进行签名，得到签名$\sigma((PK_{ac},SN),SK_{ac})$.

5)身份验证器将PK_{ac}, SN, $\sigma((PK_{ac},SN),SK_{ac})$共同发送给服务方$S_i$.

6)服务方S_i收到PK_{ac}, SN, $\sigma((PK_{ac},SN),SK_{ac})$后,首先根据存储在区块链的账户列表$acList$验证$PK_{ac}$的合法性,若合法,则进一步通过$PK_{ac}$验证签名$\sigma((PK_{ac},SN),SK_{ac})$的正确性.

7)若签名$\sigma((PK_{ac},SN),SK_{ac})$正确,则证明用户拥有对应的$SK_{ac}$,服务方$S_i$将接受该用户的注册申请,随后生成新会话号$SN_{new}$,并将账户公钥$PK_{ac}$、新会话号$SN_{new}$与用户名$UID$关联存储至数据库的用户名列表$UIDList$中,最后将新会话号$SN_{new}$返回给用户.

8)身份验证器接收到新会话号SN_{new}后,将在用户空闲时间内,使用SK_{ac}对SN_{new}进行数字签名运算得到预签名$\sigma(SN_{new},SK_{ac})$,并存储至硬件安全模块中,同时利用服务方S_i的域名$domain$将其标记,以备未来认证时调用.

4.4 用户向服务方登录

当用户已经向服务方S_i完成注册后,可利用已注册的账户公钥PK_{ac}对服务方的快速认证完成登录,具体流程如图5所示.

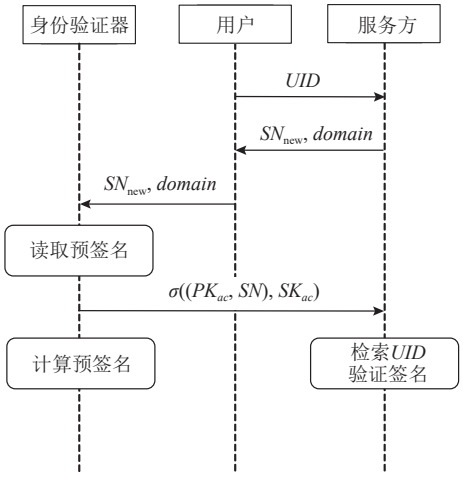


Fig. 5 Flow chart of user logging into the service provider

图5 用户向服务方登录流程图

1)用户在向服务方S_i发起登录请求,提交用户名UID.

2)服务方S_i收到登录请求后,利用伪随机数生成算法生成新的会话号SN_{new},并将SN_{new}与服务方S_i的域名$domain$共同返回给用户.

3)用户收到SN_{new}和$domain$后,调用身份验证器.身份验证器通过检索服务方域名$domain$读取对应的硬件安全模块中的预签名$\sigma(SN,SK_{ac})$并发送给服务方S_i.

4)服务方S_i收到$\sigma(SN,SK_{ac})$后,将根据用户名UID检索用户名列表$UIDList$中对应账户公钥PK_{ac}与会话号SN,并根据PK_{ac}和SN验证$\sigma(SN,SK_{ac})$的正确性,若$\sigma(SN,SK_{ac})$正确,则证明用户拥有对应的SK_{ac},服务方S_i通过用户的登录请求,用户完成登录认证.

5)服务方S_i通过用户的登录请求后,将用户名列表$UIDList$中对应的会话号更新为SN_{new}.

6)身份验证器将在用户空闲时间内,使用SK_{ac}对SN_{new}进行数字签名运算得到预签名$\sigma(SN_{new},SK_{ac})$,并将硬件安全模块中对应的预签名更新为$\sigma(SN_{new},SK_{ac})$.

4.5 用户备份身份验证器数据

用户可备份的身份验证器数据包括存储在硬件安全模块中的SK_{au}与所有SK_{ac}.当用户需要备份身份验证器中的密钥数据时,将具体执行6个步骤:

1)身份验证器在可信执行环境内生成秘密密钥$sKey$和对称密钥$eKey$.

2)身份验证器使用对称密钥$eKey$加密备份数据得到$Enc(eKey,(SK_{au}\|SK_{ac_1}\|\cdots\|SK_{ac_n}))$,同时使用秘密密钥$sKey$加密对称密钥$eKey$,得到$Enc(sKey,eKey)$.

3)用户访问验证服务器,并通过FIDO2协议向验证服务器认证.

4)验证服务器通过FIDO2认证后,用户向验证服务器上传加密后的PK_{au}、加密后的身份验证器备份数据$Enc(eKey,(SK_{au}\|SK_{ac_1}\|\cdots\|SK_{ac_n}))$与加密后的对称密钥$Enc(sKey,eKey)$.

5)验证服务器将加密后的身份验证器备份数据存储至服务器中,并与对应的FIDO2认证凭证相关联.

6)用户将秘密密钥$sKey$保存至安全存储媒介中,用于未来解密备份数据.

4.6 用户恢复备份身份验证器数据

当用户需要从验证服务器处恢复身份验证器备份数据时,将具体执行5个步骤:

1)访问验证服务器,并利用FIDO2协议向验证服务器认证.

2)验证服务器通过FIDO2认证后,根据FIDO2认证凭证读取关联的加密后的身份验证器备份数据和对称密钥,并将其返回给用户.

3)用户收到加密的备份数据和加密的对称密钥后,调用身份验证器,并传入加密后的备份数据和对称密钥.

4)用户在身份验证器中输入先前保存的秘密密钥$sKey$来解密对称密钥$eKey$.

5)身份验证器利用对称密钥$eKey$解密备份数据,得到SK_{au}与$(SK_{au}\|SK_{ac_1}\|\cdots\|SK_{ac_n})$,将其保存至硬件安全模块中供后续认证时读取.

4.7 用户注销

当用户发现或怀疑密钥泄露时，可向验证服务器申请注销PK_{au}. 与初始化阶段类似，用户需首先通过 FIDO2 协议向验证服务器进行认证. 验证服务器通过 FIDO2 认证后，将从认证列表*auList*中删除对应的PK_{au}. 同时，验证服务器将在本地存储的账户公钥映射表*acmap*检索所有与PK_{au}关联的PK_{ac}，并将这些账户公钥从账户列表*acList*中删除. 服务方将同步*acList*数据，并删除其存储的用户名列表*UIDList*中对应的PK_{ac}.

5 安全性分析

在本节中，将对系统安全性进行深入分析，以确保其在不同攻击场景下的可靠性. 在本文的具体方案设计中，采用椭圆曲线数字签名算法（ECDSA）与 AES 对称加密算法实现. 其中，ECDSA 的安全性依赖于椭圆曲线离散对数安全假设[30].

本文假设方案中使用的 TLS 信道是安全的，因此方案能够有效地抵御中间人攻击. 同时，假设硬件安全模块将有效抵抗物理和逻辑攻击，从而确保存储在其中的密钥无法被窃取. 此外，还假设身份验证器不会受到恶意软件的攻击，针对用户的任何攻击均需要通过物理访问用户设备实现. 在此基础上，本文展开对方案的安全性分析.

5.1 攻击者腐化服务方

首先，本文根据系统模型，定义安全模型和游戏，用于分析在攻击者腐化服务方的情况下本文方案的安全性. 在该游戏中，存在一个挑战者C和一个攻击者$\mathcal{A}$可以进行以下查询：

1）账户公钥查询谕言机. $\mathcal{A}$可以自适应地查询腐化服务方S_i上的任意账户公钥PK_{ac}^{i}与未被腐化服务方S_j上的任意账户公钥PK_{ac}^{j}. 对于每个这样的查询，C将模拟用户在服务方S_i或S_j上的注册过程，生成账户公私钥对. 随后，C将账户公钥PK_{ac}发送给$\mathcal{A}$，并保存相应的账户私钥SK_{ac}.

2）签名查询谕言机. $\mathcal{A}$可以自适应地查询任意消息m对应的PK_{ac}^{i}的签名σ. 对于每个这样的查询，C将使用相应的账户私钥SK_{ac}^{i}对消息m进行签名，得到签名σ，并将其发送给$\mathcal{A}$.

3）消息查询谕言机. $\mathcal{A}$可以自适应地选择一个PK_{ac}^{j}向C发起消息查询. C将随机生成一个从未在签名查询谕言机中查询过的消息m^*并返回.

4）验证谕言机. 在$\mathcal{A}$完成足够次数的上述查询后，将输出一个伪造的m^*的签名σ^*作为通过S_j验证的认证凭证. C将检查σ^*是否可被S_j有效验证. 如果能通过验证，则$\mathcal{A}$获胜，否则$\mathcal{A}$失败.

定义攻击者$\mathcal{A}$赢得上述游戏的优势为$Adv_{au}^{\mathrm{uf}}(\mathcal{A})$.

定义 1. 如果对于任何概率多项式时间攻击者$\mathcal{A}$，其优势$Adv_{au}^{\mathrm{uf}}(\mathcal{A})$都是可忽略的，那么称该方案在攻击者腐化服务方的情况下具有不可伪造性.

在定义 1 的安全模型的基础上，首先证明该方案在攻击者腐化服务方的情况下，其认证凭证具有不可伪造性，如定理 1 所示.

定理 1. 如果方案采用的签名方案是不可伪造的，则该方案在攻击者腐化服务方的情况下，其未腐化服务方的用户认证凭证是不可伪造的.

证明. 假设存在一个概率多项式时间攻击者$\mathcal{A}$，它可以在攻击者腐化服务方的情况下，以不可忽略的优势$Adv_{au}^{\mathrm{uf}}(\mathcal{A})$伪造一个有效的未被腐化服务方的认证凭证. 那么，可以构造另一个攻击者$\mathcal{B}$，它可以使用$\mathcal{A}$来破解 ECDSA 签名方案的存在不可伪造性.

给定$\mathcal{B}$签名谕言机$\sigma \leftarrow O_{\mathrm{sign}}(m, SK)$与测试谕言机$O_{\mathrm{test}}(m,\sigma,PK)$，并提供一个账户公钥列表$ListPK_{ac}$. $\mathcal{B}$的目标是在不具有私钥SK的情况下，伪造出消息m的有效签名σ. $\mathcal{B}$按如下方式模拟上述游戏中的谕言机：

1）账户公钥查询谕言机. 对于$\mathcal{A}$查询的账户公钥PK_{ac}^{i}或PK_{ac}^{j}，$\mathcal{B}$将从账户公钥列表$ListPK_{ac}$选择一个账户公钥PK_{ac}作为PK_{ac}^{i}或PK_{ac}^{j}，并将其发送给$\mathcal{A}$.

2）签名查询谕言机. 对于$\mathcal{A}$查询的消息m的签名σ，$\mathcal{B}$使用PK_{ac}^{i}查询其签名谕言机O_{sign}获得签名σ，并将σ发送给$\mathcal{A}$.

3）消息查询谕言机. 对于$\mathcal{A}$选择的未被腐化服务方S_j的账户公钥PK_{ac}^{j}，$\mathcal{B}$随机生成一个$\mathcal{A}$从未在签名查询谕言机中查询过的消息m^*，并返回给$\mathcal{A}$.

4）验证谕言机. 当$\mathcal{A}$输出伪造的签名σ^*时，$\mathcal{B}$将σ^*与$PK_{ac,j}$作为答案发送给测试谕言机O_{test}.

将$Adv_{\mathrm{ECDSA}}^{\mathrm{uf}}(\mathcal{B})$作为$\mathcal{B}$成功伪造 ECDSA 签名的优势，则有：

$$Adv_{\mathrm{ECDSA}}^{\mathrm{uf}}(\mathcal{B}) = Adv_{au}^{\mathrm{uf}}(\mathcal{A}).$$

这意味着如果$\mathcal{A}$对该方案的认证凭证有不可忽略的伪造优势，那么$\mathcal{B}$也对 ECDSA 签名有不可忽略的伪造优势. 而在 ECDSA 的 EUF-CMA 安全假设下，$\mathcal{B}$的优势$Adv_{\mathrm{ECDSA}}^{\mathrm{uf}}(\mathcal{B}) \leqslant negl(n)$，这与前文假设相矛盾. 综上所述，即使攻击者腐化了某个服务方，获得了用户在该服务方的账户公钥和所有签名消息对，也无

法伪造用户在其他未被腐化服务方上的签名.

此外，PK_{ac}均由验证服务器独立上传，不直接将PK_{ac}绑定用户的可识别信息.攻击者$\mathcal{A}$无法确定哪些PK_{ac}由特定用户上传，因而无法建立用户与PK_{ac}之间的直接关联.因此，$\mathcal{A}$仅能够知晓用户访问腐化服务方的PK_{ac}，而无法得知用户访问其他服务方所使用的PK_{ac}.即使存在攻击者能够成功腐化多个服务方，也无法将同一用户在不同服务方中使用的不同PK_{ac}相关联. 证毕.

5.2 攻击者腐化验证服务器

假设存在一个攻击者$\mathcal{A}$，能够成功腐化验证服务器.在此情况下，由于用户的SK_{au}与SK_{ac}均被安全地存储在硬件安全模块中，验证服务器不具有其访问权限.因此即使$\mathcal{A}$控制了验证服务器，也无法直接获取用户的SK_{au}与SK_{ac}.

$\mathcal{A}$可能会尝试从验证服务器中存储的用户备份数据中恢复出用户的SK_{au}与SK_{ac}.然而，在用户使用AES加密算法对备份数据进行加密且AES加密所基于的伪随机置换(PRP)存在的情况下[31]，对于任何概率多项式时间攻击者，其区分PRP和真随机置换函数的概率是可忽略的.因此$\mathcal{A}$无法通过尝试破解备份数据并获得用户的SK_{au}与SK_{ac}.

此外，验证服务器仅负责上传用户的账户公钥PK_{ac}，无法获取用户认证记录相关的关联信息.在用户后续的认证过程中，用户直接与服务方进行交互，验证服务器不参与认证流程.因此，即使$\mathcal{A}$控制了验证服务器，也无法将用户与认证的服务方相关联，从而无法推断出用户的活动轨迹或认证记录，有效防止跨站点追踪用户的行为.

5.3 攻击者窃取用户设备

若攻击者成功窃取用户设备并且能够访问身份验证器，但若无法绕过身份验证器软件中的生物识别锁，则其无法读取硬件安全模块中的SK_{au}与SK_{ac}，也无法利用身份验证器完成任何身份认证过程.

若攻击者有能力破解身份验证器软件上的生物识别锁，则可以在用户发现设备失窃之前，暂时使用该身份验证器进行身份认证.然而，一旦用户发现设备失窃并使用FIDO2密钥向验证服务器申请注销PK_{au}，验证服务器将从区块链网络中删除PK_{au}及其关联的所有账户公钥PK_{ac}，这意味着先前所有依赖这些PK_{ac}注册的服务方将不再接受对应的认证凭证，从而阻止攻击者进一步利用用户的身份验证器进行身份认证.

6 分析与评估

6.1 通信开销

为了估计验证服务器承担的负载，本文假设一个典型用户需要对50个不同的服务方进行身份验证，从而估计对验证服务器每年需要访问的API次数.本文与文献[21]中的证书颁发机构(CA)的通信开销进行分析对比，分析结果如表3所示.

Table 3 Communication Overhead

表3 通信开销

访问API	本文系统 每年通信次数	文献[21]系统 每年通信次数
FIDO2认证	1	1
注销	1	1
备份数据	1 825	1 825
会话数据同步		8 760
申请账户	50	2 425
总计	1 877	13 012

在表3的分析对比中，详细分析了本文的身份验证系统与文献[21]系统在处理通信开销方面的差异.文献[21]系统为维持会话撤销功能正常运行，需持续进行会话数据同步，且其系统中证书生命周期较短，要求用户必须频繁地与CA服务器进行通信以申请新的证书，因此其通信开销较大.在本文系统中，服务方可以通过区块链网络数据验证用户账户公钥是否被撤销，无需与验证服务器进行同步，大大降低了验证服务器的通信开销.且本文系统中账户公钥在长生命周期情况下仍能够保持高安全性，无需频繁申请新的账户公钥.结果表明，本文系统相比文献[21]系统，其服务器通信开销降低85.57%.

6.2 系统测试与评估

在本文的具体方案设计中，采用ECDSA secp256r1椭圆曲线用于实现椭圆曲线数字签名算法.为了优化区块链网络存储成本，本文方案实施了公钥的压缩存储策略，将公钥长度减少至33 B，而签名数据则维持64 B的长度.基于以上设计，本节从Gas消耗、服务方服务器性能方面分析讨论系统性能.

1)智能合约Gas消耗.为评估系统的智能合约Gas消耗，系统采用了Solidity 0.8.7版本编写智能合约，并在Optimism网络Gas价格为0.007 3 Gwei的情况下，将其部署至Optimism网络进行测试，得到Gas消耗情况.其中，据CoinMarketCap 2024年3月25日

行情，1 ETH 约等于 3 500 USD. 根据实际部署结果，合约部署消耗 Gas 最多，花费 0.000 007 7 ETH（约为 0.03 USD）. 上传压缩形式的公钥花费 0.000 000 9 ETH（约为 0.003 USD），删除公钥花费 0.000 000 4 ETH（约为 0.001 4 USD）. 从上述交易 Gas 消耗情况可以看出，系统的运行开销成本足够低，具有可观的经济可行性.

2）系统实际性能分析. 考虑到服务方读取区块链网络的数据时可能面临较高网络延迟，本文采取将区块链网络上的公钥数据同步至服务方本地数据库的优化方案，结合实时监听机制，捕获区块链网络上的数据更新，确保数据库中公钥数据的即时性和准确性. 在用户使用签名凭证对服务方进行认证的过程中，服务方将直接从高效率的本地数据库检索数据，以此验证公钥的合法性. 该设计有效消除了网络延迟的影响，从而显著缩短用户的认证时间.

为了分析实际性能，本文进一步使用智能手机作为客户端设备，本地物理机作为服务方服务器进行测评. 其中智能手机采用的配置为：型号 Redmi K40 Pro; 8 GB 内存；高通骁龙 888 处理器；Android 13 操作系统；使用 Java 编程语言进行开发. 本地物理机采用的配置为：Intel® Core™ Ultra 5 125H 1.20 GHz 处理器；内核数 14; 32 GB 内存；Windows 11 操作系统；使用 Golang 编程语言进行开发，版本号为 1.22.0. windows-AMD64; 选用 Badger 作为数据库系统，版本号为 v3.2103.5.

本文首先对客户端签名操作与服务器验证签名操作进行 1 000 次模拟测试，实验数据表明，客户端签名操作平均耗时为 9.8 ms，服务器验证签名操作平均耗时 1.2 ms. 由于本文方案采用预签名优化机制，客户端签名操作可以在离线状态中进行，在线认证过程中仅需执行服务器验证签名操作. 因此，本文方案与 FIDO2 标准流程相比，其在线计算开销降低 89.09%.

为模拟真实操作环境下的系统负荷，本文采用了广泛应用于性能测试的工具 Apache JMeter，版本号为 5.6.3. 在测试设定中，本文采用与文献 [21] 类似的用户假设，即假定系统需支持 200 万用户，每个用户平均拥有 50 条账户公钥信息，因此数据库中存储了总计 1 亿条账户公钥数据记录. 此次测试的主要目标在于评估系统在面对高并发请求时的响应性能.

测试过程中，本文配置 JMeter 以每秒 100 个用户的速率发起认证请求，以此来模拟 10 万用户进行认证操作的场景，并记录了服务端服务器的认证响应时间，结果如图 6 所示. 在该高负载测试条件下，系统表现出了优异的性能，平均认证响应时间为 2.2 ms，且 99% 的请求样本的响应时间低于 6 ms. 测试结果表明，即便在高强度并发请求的情况下，本文系统也能够维持高速响应，有效处理大量用户的认证请求，未出现明显的性能瓶颈.

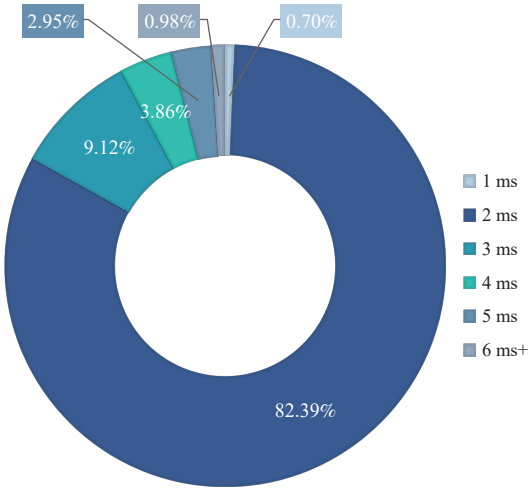


Fig. 6 Sample statistics of server authentication response time from service providers

图 6 服务方服务器认证响应时间样本统计

7 结 论

本文基于区块链提出了一种多平台兼容的高效无口令身份认证方案. 该方案融合了 FIDO2 无口令认证技术与区块链技术，在保证安全性的同时，解决了现有无口令认证系统面临的认证延迟高、身份验证器携带不方便、私钥丢失难恢复、认证不互通等问题，显著提升了用户体验. 方案主要通过构建账户线下预注册、签名预计算以及链上数据同步机制等优化措施，降低认证延迟与通信计算开销，在高负载情况下保持高效、低延迟的认证体验. 此外，方案通过结合区块链技术，实现了认证授权记录的公开透明，为系统的审计和透明度提供了强有力的保障. 理论分析和实验结果表明本文方案具有更高的实用性.

尽管本文方案通过优化设计减少了对区块链网络即时处理能力的依赖，但区块链网络的性能瓶颈仍可能成为限制因素. 在高并发请求的场景下，Optimism 网络可能面临网络拥堵、交易成本提高等问题，进而影响系统的认证与成本效益. 因此，在后续的工作中，将考虑在不同区块链网络中实现本文方案系统，以满足更多场景的应用需求，同时考虑继续优化区块链网络的数据存储结构，进一步压缩链上存储成本.

此外，本文方案所使用的传统公钥加密算法，其长期安全性受到量子计算机攻击的潜在影响. 为应

对量子计算机的攻击,未来工作将考虑采用后量子加密技术,如基于格的加密、多变量密码或哈希函数签名方案,以增强方案的安全性.

作者贡献声明:柯宇泓提出了方案框架并撰写论文;林超提出了优化思路并协助撰写论文;黄欣沂负责安全性证明;伍玮提出指导意见并修改论文;陈雨婕负责性能分析.

参 考 文 献

[1] Davey Winder. Why you should stop using LastPass after new Hack method update[EB/OL]. [2024-04-30]. https://www.forbes.com/sites/daveywinder/2023/03/03/why-you-should-stop-using-lastpass-after-new-hack-method-update/

[2] Wang Ping, Wang Ding, Huang Xinyi. Advances in password security[J]. Journal of Computer Research and Development, 2016, 53(10): 2173−2188 (in Chinese)
(王平, 汪定, 黄欣沂. 口令安全研究进展[J]. 计算机研究与发展, 2016, 53(10): 2173−2188)

[3] Bonneau J, Herley C, Van Oorschot P C, et al. Passwords and the evolution of imperfect authentication[J]. Communications of the ACM, 2015, 58(7): 78−87

[4] Lyastani S G, Schilling M, Neumayr M, et al. Is FIDO2 the kingslayer of user authentication? A comparative usability study of FIDO2 passwordless authentication[C]//Proc of 2020 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2020: 268−285

[5] Wang Ding, Wang Nan, Wang Ping, et al. Preserving privacy for free: Efficient and provably secure two-factor authentication scheme with user anonymity[J]. Information Sciences, 2015, 321: 162−178

[6] Kormann D P, Rubin A D. Risks of the passport single Signon Protocol[J]. Computer Networks, 2000, 33(1-6): 51−58

[7] Bud A. Facing the future: The impact of Apple FaceID[J]. Biometric Technology Today, 2018, 2018(1): 5−7

[8] Ross A, Shah J, Jain A K. From template to image: Reconstructing fingerprints from minutiae points[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(4): 544−560

[9] Stajano F. Pico: No more passwords![C]//Proc of Int Workshop on Security Protocols. Berlin: Springer, 2011: 49−81

[10] Yubico. Hardware authentication re-imagined[EB/OL]. [2024-04-30]. https://www.yubico.com/products/ease-of-use/

[11] Zheng Nan, Paloski A, Wang Haining. An efficient user verification system using angle-based mouse movement biometrics[J]. ACM Transactions on Information and System Security, 2016, 18(3): 1−27

[12] Chiasson S, Stobert E, Forget A, et al. Persuasive cued click-points: Design, implementation, and evaluation of a knowledge-based authentication mechanism[J]. IEEE Transactions on Dependable and Secure Computing, 2012, 9(2): 222−235

[13] Bonneau J, Herley C, Van P C, et al. The quest to replace passwords: A framework for comparative evaluation of Web authentication schemes[C]//Proc of 2012 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2012: 553−567

[14] Zhu Bo, Fan Xinxin, Gong Guang. Loxin – A solution to password-less universal login[C]//Proc of the 2014 IEEE Conf on Computer Communications Workshops. Piscataway, NJ: IEEE, 2014: 488−493

[15] Peeters R, Hermans J, Maene P, et al. n-Auth: Mobile authentication done right[C]//Proc of the 33rd Annual Computer Security Applications Conf. New York: ACM, 2017: 1−15

[16] Wang Yawei, Peng Changgen, Ding Hongfa, et al. Identity authentication scheme of Android client based on identifiers[J]. Chinese Journal of Network and Information Security, 2017, 3(4): 32−38 (in Chinese)
(王亚伟, 彭长根, 丁红发, 等. 基于标识符的 Android 客户端身份认证方案[J]. 网络与信息安全学报, 2017, 3(4): 32−38)

[17] Chakraborty D, Bugiel S. simFIDO: FIDO2 user authentication with simTPM[C]//Proc of the 2019 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2019: 2569−2571

[18] Barbosa M, Boldyreva A, Chen S, et al. Provable security analysis of FIDO2[C]//Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference. Berlin: Springer, 2021: 125−156

[19] Wagner P, Heid K, Heider J. Remote WebAuthn: FIDO2 authentication for less accessible devices[C]//Proc of the 7th Int Conf on Information Systems Security and Privacy. Setúbal Portugal: SciTePress, 2021: 368−375

[20] Schwarz F, Do K, Heide G, et al. FeIDo: Recoverable FIDO2 tokens using electronic IDs[C]//Proc of the 2022 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2022: 2581−2594

[21] Conners J, Derbidge S, Devenport C, et al. Let's authenticate: Automated certificates for user authentication[C]///Proc of the 26th Annual Network and Distributed System Security Symp. Reston, VA: The Internet Society, 2022: 24−28

[22] Yeoh W Z, Kepkowski M, Heide G, et al. Fast identity online with anonymous credentials (FIDO-AC)[C]//Proc of the 32nd USENIX Security Symp. CA: USENIX Association, 2023: 3029–3046

[23] Google. Hardware-backed Keystore[EB/OL]. [2024-04-30]. https://source.android.com/security/keystore

[24] Biometrics Research Group. Apple launches web authentication using FIDO standard with Touch ID or Face ID biometrics in Safari[EB/OL]. [2024-04-30]. https://www.biometricupdate.com/202006/apple-launches-web-authentication-using-fido-standard-with-touch-id-or-face-id-biometrics-in-safari

[25] Google. The beginning of the end of the password[EB/OL]. [2024-04-30]. https://blog.google/technology/safety-security/the-beginning-of-the-end-of-the-password/

[26] Apple. Spotlight on: Passkeys[EB/OL]. [2024-04-30]. https://developer.apple.com/news/?id=mgdnfp8w

[27] Jones J, Hodges J, Jones M, et al. Web authentication: An API for accessing public key credentials–Level 2[EB/OL]. [2024-04-30]. https://www.w3.org/TR/webauthn-2/

[28] Bradley J, Hodges J, Jones M, et al. Client to authenticator protocol (CTAP)[EB/OL]. [2024-04-30]. https://fidoalliance.org/specs/fido-v2.1-ps-20210615/fido-client-to-authenticator-protocol-v2.1-ps-20210615.html

[29] Bindel N, Cremers C, Zhao Mang. FIDO2, CTAP 2.1, and WebAuthn 2: Provable security and post-quantum instantiation[C]//Proc of 2023 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE,

2023: 1471−1490

[30] Miller V S. Use of elliptic curves in cryptography[C]//Advances in Cryptology—CRYPTO'85 Proceeding: 5th Annual Int Cryptology Conf. Berlin: Springer, 1985: 417−426

[31] Daemen J, Rijmen V. The Design Of Rijndael[M]. Berlin: Springer, 2020



Ke Yuhong, born in 2000. Master candidate. His main research interests include blockchain applications and identity authentication.

柯宇泓, 2000 年生. 硕士研究生. 主要研究方向为区块链应用、身份认证.



Lin Chao, born in 1991. PhD, professor. His main research interests include applied cryptography and blockchain technology.

林　超, 1991 年生. 博士, 教授. 主要研究方向为应用密码学、区块链技术.



Huang Xinyi, born in 1981. PhD, professor, PhD supervisor. His main research interests include applied cryptography and artificial intelligence.

黄欣沂, 1981 年生, 博士, 教授, 博士生导师. 主要研究方向为应用密码学、人工智能.



Wu Wei, born in 1981. PhD, senior lecturer. Her main research interests include cryptography and information security.

伍　玮, 1981 年生. 博士, 高级讲师. 主要研究方向为密码学、信息安全.



Chen Yujie, born in 1990. Bachelor. Her main research interests include digital government network security and cryptography applications.

陈雨婕, 1990 年生. 学士. 主要研究方向为数字政府网络安全、密码学应用.