

Blockchain-Based Authentication System For Protecting Computer Networks From Cyberattacks

Turg'unboyeva Adolatxon To'lqinjon qizi

UBS, Master's Program in Computer Networks and Their Software

Phone: +998 93 002 66 06

E-mail: adolatxonturgunboyeva5@gmail.com

Abstract:

Background. Traditional authentication systems in computer networks typically depend on a centralized server or identity management service, which creates a single point of failure and increases the risks of database breaches, credential theft, unauthorized access and other cyberattacks.

Methods. This study applies a design-oriented conceptual modelling approach combined with a qualitative threat analysis to develop a blockchain-based decentralized authentication architecture. The model incorporates public and private key pairs, digital signatures, challenge-response (nonce) verification, smart contracts and W3C Decentralized Identifiers (DID). **Results.** The proposed architecture comprises eight functional components and an eight-step registration and authentication workflow, a role-based access-control scheme implemented through a smart contract, and protection mechanisms against password theft, replay, man-in-the-middle, central-server compromise and administrator-privilege abuse. **Conclusion.** Integrating blockchain with cryptographic authentication can provide an additional security layer and reduce dependence on centralized infrastructures; however, private key management, scalability, transaction latency, smart contract security and privacy must be addressed before practical deployment.

Keywords: blockchain, cybersecurity, authentication, decentralized security system, computer networks, digital identity, cryptography, smart contract, decentralized identifier, DID, digital signature, information security, access control.

INTRODUCTION

As a result of the rapid development of digital technologies, the volume of communications between users, servers, IoT devices and various information systems in computer networks is increasing sharply

[1]. At the same time, the forms of cyberattacks aimed at unauthorized access to network resources are becoming more complex. Theft of user credentials, password cracking, phishing, session hijacking, credential stuffing, brute-force and other attacks pose a serious threat to authentication systems [2].

Authentication is the process of verifying who or what a user or other entity claims to be. OWASP describes authentication as the process of verifying the authenticity of a user, device or other entity on the basis of the authenticator it presents [3]. NIST guidelines likewise define authentication assurance levels according to the strength of the authenticators and the protocols used to prove possession of them [4].

In traditional computer networks, authentication data are often stored on a central server or in a database. One of the main problems of such a system is excessive dependence on a central component: if the central server is attacked or the database is breached, the accounts of many users may be put at risk simultaneously.

Blockchain technology offers a way to move the authentication architecture to a decentralized model. Blockchain forms a distributed, tamper-evident and tamper-resistant ledger in which records are protected through cryptographic linking [5]. Earlier studies have shown the applicability of blockchain and smart contracts to the Internet of Things, to lightweight security and privacy schemes for smart homes, and to token-based access control frameworks such as FairAccess. NIST has further examined blockchain for access-control systems and token design in blockchain networks. In parallel, the W3C has standardized Decentralized Identifiers and Verifiable Credentials, which allow digital identities to be created and verified without mandatory reliance on a central identity provider [6].

Despite these advances, a consolidated model that combines DID-based identification, challenge–response signature verification, smart-contract-based authorization and auditing for general-purpose computer networks, together with an explicit analysis of its resistance to common attacks, is still insufficiently described in the literature. This gap motivates the present study.

The **aim** of the study is to develop a conceptual model of a blockchain-based decentralized authentication architecture that increases the security of user and device identification and authentication in computer networks [7]. To achieve this aim, the following objectives were set:

- to analyze the security problems of traditional authentication systems;
- to identify the capabilities of blockchain technology in the authentication process;
- to develop a decentralized identification model in which users and devices are identified through cryptographic keys;
- to develop a model for managing access rights using a smart contract;
- to analyze the system from the point of view of the main cyberattacks;
- to identify the advantages and limitations of the proposed architecture.

METHODS

The study follows a design-oriented conceptual modelling approach. First, the security weaknesses of centralized password-based authentication were identified from normative and technical sources. Second, the functional components of a decentralized alternative were defined. Third, the registration, authentication and authorization workflow was modelled step by step. Finally, the resulting model was evaluated qualitatively against a set of common attacks and compared with a traditional centralized scheme.

The analysis relied on technical standards and guidelines of NIST specifications for Decentralized Identifiers and Verifiable Credentials, OWASP security recommendations (OWASP, n.d.-a, n.d.-b), and peer-reviewed studies on blockchain-based access control and smart contract security.

In many information systems, authentication based on a username and password is still widely used. However, the password-based mechanism depends on the strength of the password chosen by the user, as well as on the security of its storage and transmission. If a password database is breached, an attacker may obtain users' authentication data, and using the same password across several services increases the effectiveness of credential stuffing.

In brute-force attacks, the attacker tries to determine authentication data through repeated attempts; OWASP recommends throttling login attempts and, where necessary, account lockout (OWASP, n.d.-a). In systems with a central authentication server, the server represents a single point of failure: if it stops working, authentication across the entire network may be disrupted. Compromise of an administrator account also poses a great risk, since an attacker who seizes the central administrator's privileges may modify user accounts, create new accounts or illegally extend access rights. These weaknesses served as the baseline for the comparative analysis.

Blockchain is used in the model not as an ordinary database for storing passwords, but for managing identifiers, public keys, authentication policies, key status and audit data. The user keeps their own private key, while the system verifies the digital signature through the corresponding public key. Confidential personal data are kept off-chain, and only hashes or necessary metadata are placed on the ledger.

The model was assessed against five attack classes: password theft, replay attacks, man-in-the-middle attacks, compromise of the central server and abuse of administrator privileges. For each class, the corresponding protection mechanism of the proposed system was identified and compared with that of a traditional centralized system.

RESULTS

The proposed system consists of eight main components: (1) user or device; (2) digital identifier; (3) cryptographic key pair; (4) blockchain network; (5) smart contract; (6) authentication server or gateway; (7) network resources; and (8) audit mechanism. Their interaction is shown in Figure 1.

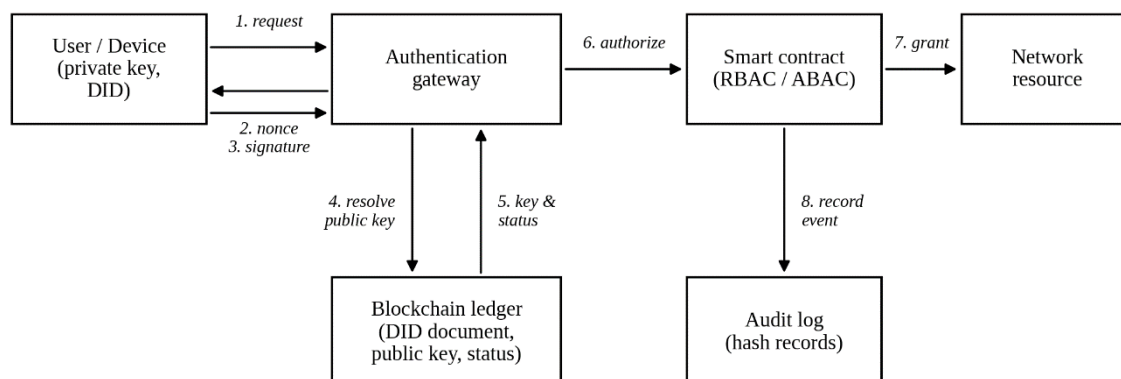


Figure 1. Architecture and message flow of the proposed blockchain-based authentication system

Each user is assigned a unique digital identifier based on the W3C Decentralized Identifiers concept,

which is intended for creating a digital identity and confirming cryptographic control without mandatory binding to a central identity provider [8]. A DID Document expresses the relevant cryptographic material and the mechanisms for managing the identifier.

The workflow of the system was formalized as eight consecutive stages:

1. **Identifier creation.** A DID or an internal decentralized identifier is created for the user.
2. **Key pair generation.** Public and private keys are generated; the private key is stored by the user in a secure environment.
3. **Ledger registration.** The user's public key, role and the necessary metadata related to the identifier are written to the blockchain network.
4. **Access request.** When accessing a network resource, the user sends an authentication request to the gateway.
5. **Challenge and signature.** The gateway sends a unique nonce; the user signs it with the private key.
6. **Signature verification.** The gateway obtains the public key from the blockchain or DID Document and verifies the signature.
7. **Authorization.** The smart contract or access-control policy checks the user's right of access to the requested resource.
8. **Session and audit.** If all conditions are met, a secure session or token is issued, and the authentication and authorization events are recorded for audit.

Through the smart contract, the user's identifier, public key and assigned rights are managed. The smart contract also records changes of rights, temporary blocking and revocation of identifiers [9]. The access-control model can be combined with RBAC or ABAC. The roles defined in the model are presented in Table 1.

Table 1. User roles and permissions defined in the smart contract

Role	Permission
Administrator	Managing system policy
Operator	Managing network resources
Regular user	Using designated resources
Guest	Using restricted resources
Device	Communicating with a designated service

The qualitative threat analysis yielded the protection mechanisms summarized in Table 2.

Table 2. Comparison of traditional and proposed authentication with respect to common attacks

Attack	Traditional	Proposed
--------	-------------	----------

	centralized system	blockchain-based system
Password theft	Password hashes stored in a central database; a breach exposes many accounts	Cryptographic key is the main authentication factor; no passwords stored on the ledger or server
Replay attack	Captured credentials or tokens may be reused	A fresh nonce for every session makes old signatures useless
Man-in-the-middle	Relies mainly on TLS	Signature-based mutual verification of parties, with TLS retained for the channel
Central server compromise	Single point of failure	Ledger replicated across several nodes; tamper-evident records
Administrator privilege abuse	One administrator account controls the system	Management powers split among several parties; multisignature approval

Password theft. Because the main authentication factor is a cryptographic key rather than a password, the need to store users' passwords on the blockchain or on a central server is reduced.

Replay attacks. In a replay attack, the attacker resends a previously transmitted authentication message. Since the server creates a new nonce for each session (Server → Nonce → User → Digital signature → Server), reusing an old signature becomes ineffective.

Man-in-the-middle attacks. Identification based on digital signatures and public keys allows the authenticity of the communicating parties to be verified; transport security mechanisms such as TLS should nevertheless be retained for the communication channel.

Central server compromise. Since the ledger is stored on several nodes, dependence on a single central database is reduced; NIST describes blockchain as a distributed, tamper-evident and tamper-resistant ledger.

Administrator privilege abuse. Instead of granting all management powers to a single administrator account, several authorized parties or multisignature mechanisms are used, which reduces the possibility of gaining full control of the system if one account is compromised.

The model can be extended with Verifiable Credentials, which express verifiable digital information about a subject through the relationship between issuer, holder and verifier [10]. For example, an enterprise may issue a digital credential to a new employee, who then uses it to access certain resources of the internal network; the system verifies the credential and determines the corresponding rights. This reduces the need to create a separate account in each service [11].

DISCUSSION

The results indicate that the proposed architecture offers five main advantages over centralized authentication. **Decentralized management** reduces dependence on a single central database. **Data integrity** is ensured because the cryptographic linking of blockchain records makes covert modification detectable. **Cryptographic authentication** with private/public keys verifies the user without relying on a password [12]. **Audit capability** leaves a verifiable trail of authentication and authorization events. **Extensible identification** through DID and Verifiable Credentials allows digital identity mechanisms to be integrated across different organizations.

These findings are consistent with earlier work showing that blockchain and smart contracts can serve as a basis for distributed access control in IoT environments and that removing a central trust point improves resilience and auditability. Unlike these studies, which focus primarily on IoT devices, the present model targets general-purpose computer networks and combines DID-based identity with challenge–response verification and role-based authorization in a single workflow [13].

At the same time, the architecture has a number of limitations. The first is **scalability**: recording a large number of authentication events on the blockchain may increase the network load. The second is **latency**: where a consensus mechanism is used, transaction confirmation time may affect near-real-time authentication. For this reason, only registration, key status changes and revocations should be written on-chain, while routine signature checks are performed by the gateway against the ledger state [14]. The third is **private key security**: if a user loses the private key or discloses it to an attacker, authentication security is compromised, so secure key storage and recovery procedures are required. The fourth is **smart contract security**: a software error in a smart contract may cause the entire access-control mechanism to behave incorrectly, and known classes of smart contract vulnerabilities must be taken into account. The fifth is **privacy**: because ledger records are immutable, personal data should be kept off-chain, with only hashes or necessary metadata stored on the blockchain.

It should also be emphasized that blockchain does not automatically protect against all cyberattacks. Network nodes, smart contracts, key storage tools and the application itself must be protected separately, and decentralized technologies such as blockchain and Web3 may themselves introduce new security problems. A further limitation of this study is its conceptual nature: the model has not yet been implemented as a prototype, and its performance has not been measured experimentally [15].

CONCLUSION

This study proposed a conceptual model of a blockchain-based decentralized authentication system for computer networks. The model replaces password-centric, server-dependent authentication with DID-based identification, private/public key signatures over session-specific nonces, smart-contract-based role management and blockchain-backed auditing.

The qualitative threat analysis showed that the proposed architecture mitigates password theft, replay attacks, man-in-the-middle attacks, central-server compromise and administrator-privilege abuse more effectively than a traditional centralized scheme. Its practical adoption, however, depends on resolving the issues of scalability, latency, private key management, smart contract security and privacy.

Future research should focus on implementing a prototype on a permissioned blockchain platform, measuring authentication latency and throughput under realistic network loads, and developing secure key recovery mechanisms for end users.

REFERENCES

- [1] OWASP Foundation, "Authentication cheat sheet," *OWASP Cheat Sheet Series*. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html (accessed Oct. 5, 2026).
- [2] P. A. Grassi *et al.*, "Digital identity guidelines: Authentication and lifecycle management," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, NIST Special Publication 800-63B, 2017, doi: 10.6028/NIST.SP.800-63b.
- [3] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, NISTIR 8202, 2018, doi: 10.6028/NIST.IR.8202.
- [4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>
- [5] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016, doi: 10.1109/ACCESS.2016.2566339.
- [6] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, 2017, pp. 618–623, doi: 10.1109/PERCOMW.2017.7917634.
- [7] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, "FairAccess: A new blockchain-based access control framework for the Internet of Things," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5943–5964, 2016, doi: 10.1002/sec.1748.
- [8] V. C. Hu, "Blockchain for access control systems," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, NIST IR 8403, 2022, doi: 10.6028/NIST.IR.8403.
- [9] L. Lesavre, P. Varin, and D. Yaga, "Blockchain networks: Token design and management overview," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, NISTIR 8301, 2021, doi: 10.6028/NIST.IR.8301.
- [10] World Wide Web Consortium (W3C), "Decentralized identifiers (DIDs) v1.0," W3C Recommendation, Jul. 2022. [Online]. Available: <https://www.w3.org/TR/did-core/>
- [11] World Wide Web Consortium (W3C), "Decentralized identifiers (DIDs) v1.1," W3C Candidate Recommendation Snapshot, Mar. 2026. [Online]. Available: <https://www.w3.org/TR/2026/CR-did-1.1-20260305/>
- [12] World Wide Web Consortium (W3C), "Verifiable credentials data model v2.1," W3C Working Draft, 2026. [Online]. Available: <https://www.w3.org/TR/vc-data-model-2.1/>
- [13] OWASP Foundation, "Multifactor authentication cheat sheet," *OWASP Cheat Sheet Series*.

[Online]. Available:
https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html
(accessed Oct. 5, 2026).

- [14] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts (SoK)," in *Principles of Security and Trust* (Lecture Notes in Computer Science, vol. 10204), M. Maffei and M. Ryan, Eds. Berlin, Germany: Springer, 2017, pp. 164–186, doi: 10.1007/978-3-662-54455-6_8.
- [15] D. Yaga and P. Mell, "A security perspective on the Web3 paradigm," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, NIST IR 8475, Feb. 2025, doi: 10.6028/NIST.IR.8475.