International Journal of Research in Advanced Engineering and Technology
www.allengineeringjournal.in
ISSN: 2455-0876
Received: 24-07-2026, Accepted: 23-08-2026, Published: 07-09-2026
Volume 12, Issue 3, 2026, Page No. 44-48

Blockchain-based authentication for secure VoIP communication

Ritu Dahiya
Assistant Professor, Department of Computer Science, Chhotu Ram Arya College, Sonipat, Haryana, India
DOI: https://doi.org/10.66856/ijraet.2026.12.3.12030

Abstract
Voice over Internet Protocol (VoIP) systems provide flexible and cost-efficient communication, but their dependence on Internet infrastructure exposes them to identity theft, session hijacking, replay attacks, call tampering, and denial-of-service attacks. Conventional VoIP authentication mechanisms generally rely on centralized servers and password-based credentials, creating single points of failure and attractive targets for attackers. This paper proposes a blockchain-based authentication framework for secure VoIP communication. The framework uses a permissioned blockchain to manage decentralized identities, public-key credentials, device registration, and authentication events. Smart contracts enforce registration and credential-revocation policies, while cryptographic challenge-response protocols provide mutual authentication between users and VoIP endpoints. The proposed design separates sensitive voice media from blockchain transactions, thereby preserving the latency requirements of real-time communication. Security analysis indicates that the framework can improve resistance to credential forgery, replay attacks, unauthorized endpoint registration, and centralized database compromise. The paper also discusses performance, privacy, scalability, and deployment challenges associated with blockchain-enabled VoIP authentication.

Keywords: Blockchain, VoIP security, decentralized identity, mutual authentication, smart contracts, SIP, permissioned ledger, real-time communication

Introduction

Voice over Internet Protocol (VoIP) transmits voice communication through packet-switched networks using protocols such as the Session Initiation Protocol (SIP) and the Real-Time Transport Protocol (RTP). VoIP is widely deployed in enterprise networks, cloud communication platforms, call centers, healthcare systems, and Internet-based telephony services. Despite these advantages, VoIP systems face significant security threats because signaling, identity management, and media transport frequently depend on distributed Internet infrastructure.

Traditional VoIP authentication commonly uses usernames, passwords, centralized identity servers, or authentication databases. These mechanisms are vulnerable to credential theft, password reuse, phishing, database compromise, and service disruption. A compromised authentication server can allow an attacker to impersonate legitimate users or register unauthorized devices. Furthermore, the lack of a tamper-resistant and independently verifiable authentication history makes it difficult to detect fraudulent registration activity.

Blockchain technology provides a distributed ledger in which authenticated participants maintain a synchronized and tamper-evident record of transactions. When combined with public-key cryptography and smart contracts, blockchain can support decentralized identity management, transparent credential validation, and auditable access control. A permissioned blockchain is particularly suitable for enterprise VoIP environments because participating organizations can be identified and authorized before joining the network.

This paper presents a blockchain-based authentication architecture for secure VoIP communication. The primary contributions are as follows:

- A permissioned blockchain architecture for managing VoIP identities and endpoint credentials.
- A mutual authentication protocol based on digital signatures and challenge-response verification.
- Smart-contract mechanisms for endpoint registration, credential validation, and revocation.
- A security analysis addressing common VoIP and blockchain-related threats.
- A discussion of performance, privacy, interoperability, and deployment considerations.

Background and Related Work

a. VoIP Communication Architecture

A typical VoIP system contains user agents, proxy servers, registrar servers, location services, and media gateways. SIP is generally responsible for session establishment, modification, and termination, while RTP carries voice media after the session is established.

SIP authentication is often based on digest authentication, in which a client proves knowledge of a shared secret without directly transmitting the password. Although digest authentication provides protection against simple password disclosure, it remains vulnerable to weak passwords, credential theft, replay under inadequate nonce management, and compromise of centralized authentication services.

Media confidentiality is commonly provided by Secure RTP (SRTP), which encrypts and authenticates voice packets. However, SRTP does not independently solve the problem of authenticating the users or endpoints participating in a call. Consequently, a secure VoIP system requires both authenticated signaling and protected media transport.

b. Blockchain for Identity Management

A blockchain ledger provides a shared record of transactions maintained by multiple nodes. Each block references the cryptographic hash of the preceding block, making

unauthorized modification detectable. Consensus mechanisms ensure that participating nodes agree on the order and validity of transactions.

For VoIP authentication, the blockchain should not store passwords, private keys, voice data, or complete call content. Instead, it may store public identifiers, public-key hashes, endpoint metadata, credential status, and audit events. Sensitive information remains off-chain in protected identity services or endpoint devices.

c. Permissioned Blockchain

A public blockchain permits unrestricted participation and generally introduces unnecessary latency, transaction fees, and privacy concerns for organizational VoIP systems. A permissioned blockchain restricts participation to approved entities, such as enterprises, telecom operators, service providers, and identity authorities.

Consensus protocols designed for permissioned environments, including Practical Byzantine Fault Tolerance and Raft-based mechanisms, can provide lower latency than proof-of-work systems. The choice of consensus mechanism depends on the trust model, number of organizations, and required fault tolerance.

d. Research Gap

Existing blockchain-based identity systems demonstrate the suitability of distributed ledgers for credential management, but many designs do not address the strict latency, privacy, and interoperability requirements of VoIP. In particular, writing every signaling event to a blockchain would introduce unacceptable delay and transaction overhead. Therefore, a practical architecture must use blockchain for identity validation and auditability while allowing SIP and RTP traffic to operate through conventional low-latency paths.

System Model and Design Requirements

a. System Entities

The proposed system contains the following entities:

- **User Agent:** A software or hardware VoIP endpoint used to initiate or receive calls.
- **Identity Provider:** An authorized organization responsible for verifying users and issuing credentials.
- **VoIP Registrar:** A SIP service that records endpoint bindings and provides session-registration services.
- **Blockchain Nodes:** Authorized nodes that maintain the distributed ledger.
- **Smart Contracts:** Blockchain programs that enforce registration, authentication, and revocation rules.
- **Certificate Authority:** An optional authority that issues certificates binding public keys to organizational identities.
- **Media Security Service:** A service that supports SRTP key establishment and media-security policies.

b. Design Requirements

The system must satisfy the following requirements:

1. **Mutual authentication:** Both communicating endpoints should verify one another.
2. **Credential integrity:** Unauthorized modification of identity records must be detectable.
3. **Replay resistance:** Previously captured authentication messages must not be reusable.
4. **Revocation support:** Compromised devices and invalid credentials must be revoked promptly.
5. **Low latency:** Authentication must not require blockchain processing for every RTP packet.
6. **Privacy preservation:** Personal information and voice content must remain off-chain.
7. **Availability:** The compromise or failure of one blockchain node must not disable authentication.
8. **Interoperability:** The design should operate with SIP, TLS, and SRTP.
9. **Auditability:** Authorized administrators should be able to inspect authentication events.
10. **Scalability:** The system should support multiple organizations and large endpoint populations.

c. Threat Model

The attacker is assumed to be capable of intercepting, modifying, replaying, or injecting network packets. The attacker may attempt to steal credentials, register a fraudulent endpoint, impersonate a legitimate user, compromise a registrar, or disrupt authentication services. The attacker is not assumed to compromise a majority of authorized blockchain nodes simultaneously.

The following assumptions are made:

- Private keys are stored in secure endpoint storage or hardware-backed modules.
- Blockchain nodes use authenticated channels.
- At least a quorum of blockchain nodes remains honest.
- VoIP media is protected using SRTP or an equivalent secure media protocol.
- Initial identity verification is performed by an authorized identity provider.

Proposed Blockchain-Based Authentication Architecture

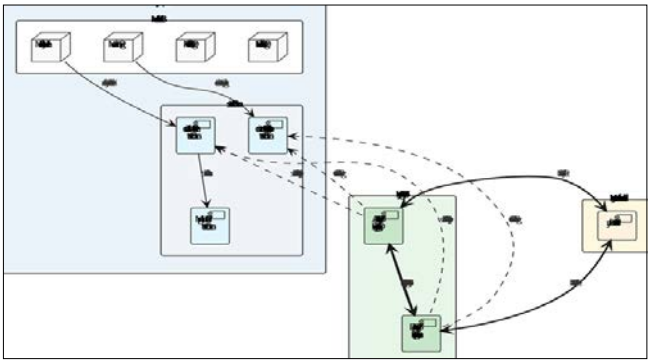
a. Architectural Overview

The proposed architecture uses a permissioned blockchain as a decentralized trust registry. Each registered user or endpoint is assigned a decentralized identifier and a public-key credential. The corresponding private key remains under the control of the endpoint or user.

The blockchain stores the following information:

- A decentralized identifier.
- A hash of the associated public key.
- Endpoint type and organization identifier.
- Credential issuance time.
- Credential expiration time.
- Credential status.
- Revocation information.
- Authentication-event references.

The blockchain does not store passwords, private keys, biometric information, call recordings, SIP message bodies, or RTP packets.



b. Decentralized Identity Representation

Let a user or endpoint be represented by the decentralized identifier

$$D_U = H(ID_U \parallel PK_U \parallel T_I) \quad (1)$$

Where ID_U is the logical identity, PK_U is the public key, T_I is the issuance timestamp, $\parallel$ denotes concatenation, and H is a cryptographic hash function.

The identity provider signs the credential:

$$\sigma_U = Sign_{SK_{IP}}(D_U \parallel PK_U \parallel T_I \parallel T_E) \quad (2)$$

Where SK_{IP} is the identity provider's private signing key, T_I is the credential-issuance time, and T_E is the expiration time.

A verifier can validate the credential using the identity provider's public key. The blockchain stores the credential status and a verifiable reference to the public-key record.

c. Smart-Contract Functions

The proposed smart-contract layer provides the following operations:

- **RegisterEndpoint:** Adds a new user or endpoint after identity-provider approval.
- **UpdateKey:** Replaces a public key after successful authorization.
- **RevokeCredential:** Marks a credential as revoked.
- **ValidateCredential:** Returns the status and validity interval of a credential.
- **RecordAuthentication:** Stores a compact authentication-event reference.
- **ResolveIdentity:** Maps an authorized decentralized identifier to permitted service metadata.

Smart contracts should enforce role-based permissions. For example, only an approved identity provider may issue a credential, while only an authorized security administrator may revoke a credential.

d. Authentication Protocol

The protocol consists of endpoint enrollment, SIP registration, and call authentication.

1. Endpoint Enrollment

During enrollment, the endpoint generates a key pair:

$$(PK_U, SK_U) \leftarrow KeyGen() \quad (3)$$

The private key SK_U is stored locally and must not be transmitted. The endpoint sends its public key and identity information to the identity provider through a protected enrollment channel. After verification, the identity provider creates and signs the credential.

The credential reference is submitted to the blockchain through the registration smart contract. Blockchain nodes validate the transaction according to the permissioned consensus policy.

2. SIP Registration

When registering with a VoIP service, the endpoint generates a fresh nonce N_U and a timestamp T_U. It signs the registration context:

$$\sigma_R = Sign_{SK_U}(D_U \parallel ID_R \parallel N_U \parallel T_U) \quad (4)$$

Where ID_R identifies the registrar.

The registrar performs the following checks:

1. Verify the identity-provider signature.
2. Query the blockchain for credential status.
3. Confirm that the credential has not expired or been revoked.
4. Verify σ_R using PK_U.
5. Confirm that N_U and T_U satisfy freshness requirements.
6. Create the SIP registration binding.

The registrar returns a signed response containing the original nonce and a registrar-generated nonce:

$$\sigma_S = Sign_{SK_R}(D_U \parallel N_U \parallel N_R \parallel T_R) \quad (5)$$

Where SK_R is the registrar's private key, N_R is a fresh registrar nonce, and T_R is the response timestamp.

3. Mutual Call Authentication

Before establishing a call, the originating endpoint and receiving endpoint exchange signed authentication challenges. Let C denote the call context, including the SIP Call-ID, dialog identifier, and negotiated security parameters. The originating endpoint signs

$$\sigma_A = Sign_{SK_A}(D_A \parallel D_B \parallel C \parallel N_A) \quad (6)$$

And the receiving endpoint signs

$$\sigma_B = Sign_{SK_B}(D_B \parallel D_A \parallel C \parallel N_B) \quad (8)$$

Each endpoint verifies the peer's credential through the blockchain or a trusted cached ledger state. The call proceeds only if both signatures are valid, the credentials are active, and the nonces are fresh.

e. Media-Key Establishment

Blockchain authentication should not be used to transport voice media keys directly. After endpoint authentication, the endpoints may use Datagram Transport Layer Security Secure Real-Time Transport Protocol or another

authenticated key-establishment mechanism to derive SRTP session keys.
The media path then uses encryption and authentication independently of the blockchain:

$$\text{SRTP packet} = \text{Encrypt}_{K_{SRTP}}(\text{RTP payload}, \text{sequence number}).$$

This separation prevents blockchain latency from affecting individual voice packets.

Security Analysis

a. Impersonation Resistance

An attacker attempting to impersonate a legitimate endpoint must produce a valid signature using the endpoint's private key. Knowledge of a decentralized identifier or public key is insufficient to generate such a signature. If private keys are protected by secure hardware or strong endpoint security, identity impersonation becomes substantially more difficult.

b. Replay-Attack Resistance

The protocol includes nonces, timestamps, and call-specific context. A captured authentication message contains a previously used nonce or expired timestamp and therefore fails freshness validation. Binding the signature to the SIP Call-ID and dialog context also prevents an authentication message from being reused in another call.

c. Credential Forgery

A fraudulent credential must contain a valid identity-provider signature and a blockchain record accepted by the smart contract. An attacker cannot create a valid credential without compromising the identity provider's signing key or the permissioned blockchain consensus process.

d. Credential Revocation

If an endpoint is lost, compromised, or decommissioned, an authorized administrator can invoke the revocation function. Registrars and endpoints reject authentication attempts associated with revoked credentials. Short credential lifetimes and periodic status synchronization can reduce the interval during which a compromised credential remains usable.

e. Registrar Compromise

A compromised registrar should not be able to create valid user credentials because credential issuance is controlled by the identity provider and recorded on the permissioned ledger. Mutual authentication also limits the registrar's ability to impersonate endpoints. However, a malicious registrar may still disrupt service by refusing registrations. Redundant registrars and multiple blockchain nodes are therefore necessary for availability.

f. Ledger Tampering

Each blockchain block is linked to the previous block by a cryptographic hash. An attacker modifying historical identity records must alter subsequent blocks and obtain agreement from the required consensus quorum. Under the assumed threat model, compromise of fewer than the required number of blockchain nodes cannot silently rewrite the ledger.

g. Denial-of-Service Attacks

Blockchain does not eliminate denial-of-service attacks. Attackers may flood SIP registrars, identity services, or blockchain gateways with requests. Rate limiting, endpoint admission control, caching of validated credentials, and network-level filtering should be deployed to reduce this risk.

h. Privacy Protection

Blockchain records are replicated among authorized nodes and may remain available for long periods. Therefore, storing personally identifiable information or detailed call histories on-chain creates privacy risks. The proposed architecture stores hashes, identifiers, status values, and minimal audit references. Detailed records remain in encrypted off-chain repositories subject to organizational access policies.

Performance and Deployment Considerations

a. Latency

Real-time voice traffic requires low and predictable delay. The design avoids placing blockchain transactions in the RTP path. Blockchain queries are used during endpoint registration, credential validation, and selected authentication events. Cached credential status can reduce repeated ledger queries during active communication.

b. Scalability

A large deployment may contain thousands or millions of endpoints. Storing every SIP message on-chain would create excessive ledger growth. The proposed system stores only identity changes, credential status transitions, and compact authentication references. High-volume signaling remains off-chain.
Blockchain gateways may improve scalability by maintaining synchronized read-only caches of credential status. Cache entries must have controlled lifetimes and must be invalidated when revocation updates are received.

c. Fault Tolerance

Multiple blockchain nodes should be distributed across organizations or network regions. A quorum-based consensus mechanism enables continued operation when a limited number of nodes fail. Registrar redundancy and replicated identity providers further reduce the risk of service interruption.

d. Key Management

The security of the system depends heavily on private-key protection. Recommended controls include:

- Hardware-backed key storage.
- Secure endpoint boot processes.
- Key rotation policies.
- Separate keys for signing and encryption.
- Multi-factor authorization for administrative operations.
- Immediate revocation after suspected compromise.
- Recovery procedures for lost devices.

e. Interoperability

The proposed architecture can be integrated with existing VoIP protocols by placing the blockchain authentication function at the identity and registration layer. SIP messages can carry decentralized identifiers, credential references, signatures, and nonce values through standardized or vendor-defined headers. TLS can protect signaling channels, while SRTP protects media.

Interoperability requires agreement on credential formats, signature algorithms, nonce encoding, timestamp validation, revocation status, and blockchain gateway interfaces.

Limitations

The proposed framework introduces additional infrastructure, including blockchain nodes, smart contracts, identity providers, and secure key-management systems. This increases deployment and administration complexity compared with conventional centralized authentication.

Blockchain consensus and synchronization may also introduce delay during credential updates or revocation. Although this delay does not affect established media packets, it may influence new registrations and call setup. In addition, blockchain replication can expose metadata such as credential changes and organizational relationships to authorized ledger participants.

The architecture cannot protect an endpoint whose private key has been extracted by malware or whose operating system has been fully compromised. Endpoint security, secure software updates, intrusion detection, and revocation procedures remain essential.

Finally, the security of the system depends on the governance model of the permissioned blockchain. Collusion among a sufficient number of trusted organizations could undermine ledger integrity or availability.

Future Work

Future research should investigate the following areas:

- Formal verification of the smart contracts and authentication protocol.
- Benchmarking under realistic VoIP call volumes and network conditions.
- Privacy-preserving credential status mechanisms.
- Zero-knowledge proofs for identity validation without unnecessary disclosure.
- Automated key rotation and decentralized recovery.
- Integration with 5G voice services and edge-computing platforms.
- Cross-domain authentication between independent telecom providers.
- Machine-learning methods for detecting anomalous registration behavior.
- Post-quantum signature schemes for long-term credential protection.

A comprehensive evaluation should measure registration latency, call-setup delay, transaction throughput, storage overhead, revocation propagation time, and resilience under node failures and network attacks.

Conclusion

This paper presented a blockchain-based authentication framework for secure VoIP communication. The proposed design uses a permissioned blockchain to provide decentralized identity registration, public-key verification, credential revocation, and tamper-evident auditability. Mutual authentication is achieved through digital signatures, nonces, timestamps, and call-specific context. Voice media remains outside the blockchain and is protected using SRTP-compatible mechanisms, preserving the latency requirements of real-time communication.

The architecture improves resistance to endpoint impersonation, credential forgery, replay attacks, and centralized authentication-server compromise. Nevertheless, practical deployment requires careful attention to key management, privacy, consensus performance, governance, and interoperability. Blockchain should therefore be used as a trusted identity and policy layer rather than as a transport mechanism for signaling or media packets.

References

1. Rosenberg J et al. SIP: Session Initiation Protocol. RFC 3261, Internet Engineering Task Force, 2002.
2. Bormann C, Casner M, Jacobson V. RTP: A Transport Protocol for Real-Time Applications. RFC 3550, Internet Engineering Task Force, 2003.
3. Baugher M et al. The Secure Real-time Transport Protocol. RFC 3711, Internet Engineering Task Force, 2006.
4. Rescorla E, Dierks T. The Transport Layer Security Protocol Version 1.2. RFC 5246, Internet Engineering Task Force, 2008.
5. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.
6. Castro M, Liskov B. Practical Byzantine fault tolerance. in Proc. 3rd Symp. Operating Systems Design and Implementation, New Orleans, LA, USA, 1999, 173-186.
7. Arner DW, Buckley RP, Zetzsche DA. Fintech, RegTech, and the role of compliance report. Journal of Financial Transformation,2018:47:1-13.
8. National Institute of Standards and Technology. Digital Identity Guidelines. Special Publication 800-63-3, Gaithersburg, MD, USA, 2017.
9. Stallings W. Cryptography and Network Security: Principles and Practice. 8th ed. Boston, MA, USA: Pearson, 2020.
10. Shirey A. Internet Security Glossary, Version 2. RFC 4949, Internet Engineering Task Force.