ZKAC DAM: Zero-Knowledge Access Control for Dynamic and Anonymous Memberships V3

NUS FinTech Society Blockchain

Problem Statement

Unauthorized access to campus facilities is a growing concern. To address this, we propose a blockchain-based, zero-knowledge access control system for dynamic and anonymous memberships that allows students to prove their membership without revealing their identity, ensuring privacy and preventing the sharing of access keys. This system updates students' keys after each use, avoiding the need for traditional membership proofs that require group-wide changes. This approach is applicable to various access control scenarios and addresses an open research problem.

Pain Points with Current Solutions

Current access control solutions are plagued by several critical issues that make them unsuitable for environments requiring both security and privacy. These issues are particularly problematic in scenarios involving group membership, such as student access to campus facilities. Below are the key pain points:

- **Individual Identification Risks:** Traditional systems, such as student ID cards or RFID badges, require identifiable credentials for access. These methods compromise user privacy by exposing personal information during authentication. If the system is breached, attackers can obtain sensitive data, posing significant security risks.
- **Limitations of Group Membership Proofs:** Group membership proofs, like group signatures, enable users to prove their membership without revealing individual identities. However, these systems struggle with key management. If a key is compromised, the security of the entire group is at risk. Additionally, they lack flexible mechanisms for revoking or updating individual credentials without affecting the whole group.

Proposed Solution Overview

Our solution leverages interactive Zero-Knowledge Proofs (ZKPs) and ElGamal digital signature to create a Zero-Knowledge Access Control for Dynamic and Anonymous Memberships (ZKAC DAM). In this system, users (e.g., students) interact with a manager (e.g., school) and a smart gate (e.g., a library entrance system). The solution ensures that once a user's key is used for authentication, it is invalidated and replaced with a new key, without revealing any link between the old and new keys.

Key Features

- **Privacy-Preserving Access Control:** Users can authenticate using Zero-Knowledge Proofs (ZKPs) without revealing their secret keys or identities.
- **Group Membership Authentication:** The system allows users to prove membership in a group without disclosing their individual identity, maintaining anonymity within the group.
- **Dynamic Key Updates:** After each authentication, a user's key is invalidated and replaced with a new one. This prevents key reuse and reduces the risk of key compromise, while allowing individual key updates without affecting the entire group.
- **Unlinkability:** Even though the Smart Gate processes authentication requests and the blockchain records key updates, the system ensures that no party can link a user's past and future sessions, preserving privacy across multiple authentications.
- **Resilience to Attacks:** The system is designed to resist common attacks such as replay attacks, impersonation, and man-in-the-middle attacks, using cryptographic techniques and blockchain for tamper-proof key management.
- **No Impersonation:** The secret key is never shared to any other party, ensuring that no other party, even the manager, can impersonate the user.

Construction of Our Updated Scheme

Main Components

- **Manager (School):** A trusted entity that manages key verification and revocation but does not generate keys. The manager maintains a list of students' identities matched with their latest public keys.
- **User (Student):** An individual who generates their own secret and public keys and proves their membership without revealing their identity.
- **Smart Gate:** Verifies the user's membership, interacts with the blockchain to invalidate used keys, and generates an invalidation ID for key updates.

1. Registration

- **Key Generation by the User:** The student generates their own secret key x and computes the corresponding public key y as:

$$y = g^x \mod p$$

  where g is a generator of a large cyclic group and p is a large prime.
- **Public Key Submission:** The student submits their public key y to the school (manager), who keeps a record of the student's identity and their latest public key.
- **Blockchain Upload:** The manager uploads the public key y to the blockchain, making it publicly accessible. This serves as the student's commitment for future authentication.

2. Authentication Process

- **Zero-Knowledge Proof (Sigma Protocol):** When a student wishes to authenticate, they engage in a Zero-Knowledge Proof (ZKP) with the Smart Gate, proving their knowledge of the secret key x without revealing it.
- **Proof Procedure:**
  1) **Commitment:** The student selects a random r and computes the commitment:

$$a = g^r \mod p$$

     The student sends a to the Smart Gate.
  2) **Challenge:** The Smart Gate generates a random challenge e and sends it to the user.

3) **Response:** The student computes the response:

$$s = r - e \cdot x \mod (p-1)$$

The student sends s to the Smart Gate.

4) **Verification:** The Smart Gate verifies the proof by checking:

$$g^s \cdot y^e \stackrel{?}{=} a \mod p$$

If the equation holds, the authentication is successful.

- **Invalidation of Public Key & Invalidation ID:** Upon successful authentication, the Smart Gate uploads a transaction to the blockchain marking the student's public key y as invalidated. In addition:
  - The Smart Gate generates a unique **invalidation ID**, tied to the invalidated key.
  - The **invalidation ID** is encrypted using the manager's public key and sent to the manager. This ensures that only the manager can decrypt and view the invalidation ID.
  - The **invalidation ID** is also given to the authenticated user via NFC, to prevent remote replay attack.

3. Dynamic Key Update

- **Monitoring by the Manager:** The manager listens to on-chain events for invalidated public keys and maintains a list of invalidated keys.
- **Key Update by the User:** After a student's old key is invalidated, they generate a new secret key x_{new} and corresponding public key y_{new}, where:

$$y_{\text{new}} = g^{x_{\text{new}}} \mod p$$

To prove ownership of the old public key y, the student uses ElGamal digital signatures to sign the new public key y_{new} with their old secret key x_{old}. The signature process is denoted as:

$$\text{Sign}_{\text{ElGamal}}(x_{\text{old}}, y_{\text{new}})$$

The content sent by the student to the manager includes:

  - The old public key y_{old} (for reference).
  - The new public key y_{new} (in plaintext).
  - The digital signature $\text{Sign}_{\text{ElGamal}}(x_{\text{old}}, y_{\text{new}})$, proving ownership of the old key.
  - The **invalidation ID** obtained during the previous authentication.
- **Verification by the Manager:** The manager verifies the signature using the old public key y_{old} and checks if the old public key is in the invalidation list. Additionally, the manager checks that the provided **invalidation ID** matches the one it received from the Smart Gate. If the signature and invalidation ID are valid and the old public key is in the list, the manager removes the old public key from the list and uploads the new public key y_{new} to the blockchain.

4. Revocation Process

If a student's access is revoked (e.g., upon graduation or expulsion), the manager references the table of student identities and their latest public keys:

- **Public Key Invalidation:** The manager invalidates the student's public key on the blockchain.
- **Removal from Records:** The student's identity and public key are removed from the manager's records, and the public key is not added to the invalidation list, preventing re-registration.

SECURITY ANALYSIS

Threat Model

In this section, we describe the different types of attacks our system must defend against to ensure security and privacy for users.

- **Relay Attack (Man-in-the-Middle Collaboration):** An unauthorized user (Bob) could collaborate with a legitimate user (Alice) to relay authentication messages. Bob intercepts the challenge from the Smart Gate, forwards it to Alice, who computes the response and sends it back to Bob. Bob then forwards Alice's response to the Smart Gate, gaining access without knowing Alice's secret key.
- **Replay Attack:** An adversary captures a valid authentication message (such as a challenge-response pair) and tries to reuse it later to gain unauthorized access.
- **Man-in-the-Middle (MitM) Attack:** An attacker intercepts the communication between a legitimate user and the Smart Gate, attempting to manipulate or eavesdrop on the Zero-Knowledge Proof (ZKP) exchange to gain access or learn sensitive information.
- **Smart Gate's Attempt to Link Users' Identities (Linkability Attack):** The semi-trusted Smart Gate could attempt to link multiple authentication sessions to the same user by observing patterns in the authentication process or the blockchain, compromising user privacy.
- **Impersonation Attack:** An attacker attempts to authenticate as another user without knowing their secret key, using some form of manipulation or spoofing to impersonate a legitimate user.

Security Guarantees

Our system provides strong security guarantees against the threats listed above. Below, we explain how the core features of our protocol mitigate each attack.

- **Defense Against Relay Attack:** The introduction of an **invalidation ID** ensures that if a legitimate user (Alice) helps an attacker (Bob) relay authentication, Bob would receive the invalidation ID after successful authentication. Since the invalidation ID is required for future key updates, Bob could withhold it, preventing Alice from updating her key. This creates a strong disincentive for Alice to participate in such attacks, as she risks losing access to the system herself.
- **Defense Against Replay Attack:** Each authentication session involves a fresh challenge generated by the Smart Gate, making previously captured responses useless. Additionally, once a key is used for authentication, it is immediately invalidated and cannot be reused, preventing replay attacks.
- **Defense Against Man-in-the-Middle (MitM) Attack:** The use of Zero-Knowledge Proofs (ZKPs) ensures that even if an attacker intercepts the communication, they learn nothing about the user's secret key. The ZKP protocol only proves knowledge of the secret key without exposing it. Furthermore, the **invalidation ID** is encrypted using the manager's public key, ensuring that sensitive information remains secure even in the presence of an attacker.
- **Defense Against Smart Gate's Linkability Attack:** The system ensures unlinkability by decoupling the authentication process from the key update process. The Smart Gate handles user authentication and invalidates keys, but the issuance of new keys is managed by the trusted manager. Even though the Smart Gate can observe the blockchain, it cannot link previously invalidated keys to newly issued ones, preserving user anonymity and preventing identity linkage.

- **Defense Against Impersonation Attack:** The use of Zero-Knowledge Proofs ensures that only the user who knows the correct secret key can authenticate. Since the secret key is never revealed during the authentication process, it is impossible for an adversary to impersonate a legitimate user. Additionally, even the manager, who oversees public key updates, cannot impersonate the user since they do not have access to the user's secret key.

Value Proposition

The Zero-Knowledge Access Control for Dynamic and Anonymous Memberships (ZKAC DAM) prioritizes user privacy and security, addressing critical weaknesses in existing access control solutions. Traditional systems often necessitate a trade-off between privacy and security, either exposing user identities through identifiable credentials or relying on group proofs vulnerable to key compromises and inefficient group-wide updates. ZKAC DAM leverages Zero-Knowledge Proofs (ZKPs) to enable users to prove membership without revealing their identity, ensuring complete privacy. The system dynamically updates keys after each use, preventing key reuse and significantly reducing the risk of compromised credentials. With blockchain integration, ZKAC DAM offers immutable and auditable key management, enhancing both security and scalability. This approach provides a secure, scalable, and privacy-preserving solution suitable for diverse environments such as academic institutions, corporate settings, and any organization requiring secure, group-based, and privacy-preserving access control.

Future Efforts

The proposed solution addresses the primary concerns of privacy, unlinkability, and secure key updates. However, there are several areas for future exploration and improvement:

- **Handling Blockchain Delays:** The delay between key invalidation and the issuance of new keys could be a concern, especially in time-sensitive applications. Exploring techniques to minimize this delay or to allow for immediate key updates in a decentralized manner could be an area of further research.
- **Further Enhancing Privacy:** Although the current solution provides strong privacy guarantees (discrete logarithm), additional techniques could be explored to further enhance the privacy of users in various access control scenarios.
- **Scalability and Efficiency:** As the number of users grows, the scalability and computational efficiency of the system will become increasingly important. Further optimization in key update process and encryption techniques could help in scaling the solution to larger populations.
- **Balancing Privacy and Key Revocation:** While the current protocol ensures that the manager (e.g., the school) is unable to impersonate users, the manager is still capable of tracking each student's public key and associating it with their identity and access histories. This tracking is necessary for key revocation in cases such as graduation or expulsion. However, this introduces a potential privacy concern. Finding a balance between functionality (e.g., key revocation) and user anonymity is an ongoing area of research. Future work could focus on developing mechanisms that allow for key revocation without sacrificing user privacy, potentially through the use of decentralized identity management or zero-knowledge-based revocation mechanisms.

Tech Stack

- **Frontend:** React
- **Backend:** Node.js, Python
- **Blockchain:** Ethereum (Solidity)
- **Cryptography:** elgamal

Milestone Timeline

September - October: Project Definition and Initial Development

- Formalize the project scope, define security goals, and prove security properties (soundness, zero-knowledge, correctness).
- Finalize the overall system architecture and protocol management for each component.
- Design and develop smart contracts for key management on the blockchain.

November - December: Implementation of Core Components

- Implement backend cryptographic modules for each party (TTP, Authenticator, User).
- Conduct initial unit testing of backend components.

January: Frontend Development

- Develop the frontend user interface.
- Begin integration of the frontend with back-end APIs and services.
- Conduct initial UI/UX testing and make adjustments as needed.

February - March: Integration and System Testing

- Complete the integration of the frontend, backend, and blockchain components.
- Perform system-wide testing, including functional, security, and performance tests.
- Refine the system based on feedback from integration and testing phases.
- Deploy a Minimum Viable Product (MVP) for broader testing and feedback.

April: Finalization and Documentation

- Finalize the system, incorporating any remaining fixes or improvements.
- Complete comprehensive project documentation, including user guides, technical documentation, and project report.
- Prepare for deployment or road show presentation, if applicable.

Conclusion

This proposal outlines a privacy-preserving access control system that allows users to authenticate themselves as members of a group without revealing their identity. The system ensures that users have only one valid key at a time, preventing key sharing and reuse. By leveraging Zero-Knowledge Proofs and ElGamal encryption, the solution provides strong privacy guarantees, making it suitable for environments where secure and private access control is critical, such as on university campuses or other institutional settings.

References

[1] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991. [Online]. Available: https://doi.org/10.1007/BF00196725

[2] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[3] A. Sonnino, M. Al-Bassam, S. Bano, S. Meiklejohn, and G. Danezis, "Coconut: Threshold Issuance Selective Disclosure Credentials with Applications to Distributed Ledgers," in *Proc. Network and Distributed Systems Security (NDSS) Symposium 2019*, San Diego, CA, USA: The Internet Society, 2019, pp. 14–15.