zk-Embedded Authentication: A Ledger-Free Decentralized Identifier Scheme

Yunsik Ham
Chungnam National University, Information Security Laboratory

***Abstract*—Current implementations of Decentralized Identifier (DID) face limitations in terms of interoperability and flexibility due to their reliance on distributed ledgers and associated governance mechanisms. Specifically, ledger-based DIDs are dependent on specific ledgers and trust anchors, making it challenging to establish an authentication framework resilient to Byzantine conditions. This paper proposes a DID authentication scheme for a ledger-free environment to overcome these limitations. Our approach integrates the structured Verifiable Credential (VC) model using Sparse Merkle Trees (SMT) with zk-SNARK-based authentication mechanisms and introduces wallet-based embedded authentication through hardware wallets. By decoupling the issuer's trust from the ledger and leveraging Zero-Knowledge Proofs (ZKP), our solution simultaneously ensures privacy, security, and interoperability. Additionally, it supports cross-chain authentication by adopting standard signature methods such as EIP712 and zk-friendly elliptic curves. The proposed scheme addresses ledger dependency and trust anchor issues through the auditable data structure of DID documents, implementing a flexible and scalable DID authentication framework adaptable to various scenarios, including Byzantine conditions.**

***Index Terms*—Decentralized Identifier, zk-SNARK, Sparse Merkle Tree, Verifiable Credential, Cross-Chain Authentication, Zero-Knowledge Proof.**

I. Introduction

DECENTRALIZED Identifiers (DIDs) are a core component of Self-Sovereign Identity (SSI), enabling users to directly manage and control their identity information [**?**]. With the increasing adoption of DID systems, various verification methods and credential architectures have been proposed [**?**], [**?**], [**?**]. However, current ledger-based DID approaches face interoperability challenges due to their isolated nature [**?**]. This limitation arises from the reliance of DID controllers on the trustworthiness of distributed ledgers and their governance mechanisms, compounded by the lack of robust trust models [**?**].

The consensus process in distributed ledgers ensures the reliability of ledger data, encompassing not only cryptographic integrity but also trust derived from consensus procedures and governance. To maintain this trust implication in multi-chain environments, auditable DID architectures have been defined [**?**], [**?**]. The W3C Verifiable Credentials Data Model v2.0 introduces *Embedded Proofs* as a securing mechanism [**?**]. Embedded Proofs incorporate cryptographic proofs, such as digital signatures, directly within the data model, enabling verification without external references. These are compatible with Zero-Knowledge Proof (ZKP) schemes like *Selective Disclosure* and *Unlinkable Proof*, allowing for privacy-preserving verification [**?**].

Recent research has explored approaches that simultaneously satisfy privacy and interoperability requirements using pairing-based ZKP technologies, such as the BBS Scheme [**?**]. For instance, DIDAPPER [**?**] proposes a DID architecture that preserves user privacy through group signatures and selective disclosure mechanisms while enabling conditional auditing. These studies and standardization efforts primarily target ledger-based DID methods, defining auditable architectures indirectly via ZKP. SSI systems ensure security, privacy, and interoperability of identity data by selecting cryptographic technologies, ledgers, and protocols based on the system's characteristics [**?**].

However, these approaches still fail to fully address the issues of ledger dependency and trust anchors. Existing SSI solutions often consider issuers, such as government institutions, as trusted authorities but lack robust trust management frameworks for issuers, including individuals [**?**], [**?**]. This limitation restricts interoperability among diverse SSI systems and conflicts with the goal of SSI to grant users complete control over their identity.

To address these challenges, this study aims to propose a DID authentication scheme for a ledger-free environment. Specifically, the following are proposed:

1) **zk-SNARK-based DID Authentication Scheme**: Introduces a DID VC/VP structure utilizing Sparse Merkle Trees (SMTs) and defines a scheme for verifying VC integrity and issuer signature validity without a ledger, leveraging Zero-Knowledge Proofs (ZKP).
2) **Wallet-based Embedded Authentication**: Employs hardware wallets to manage VCs and generate VP proofs, enabling client-side ZKP operations. This approach allows for verification without relying on external ledgers or governance mechanisms and supports quick and efficient VP generation in browser environments using tools like Circom and SnarkJS.

The contribution of this study lies in mitigating ledger dependency and trust anchor issues in DID authentication systems. By exploring the feasibility of ledger-free authentication under Byzantine conditions, this research proposes a new standard that enhances privacy, security, and interoperability. Subsequent sections will elaborate on these concepts, presenting practical implementation strategies and trust models.

This work was supported by Chungnam National University, Information Security Laboratory.

II. Background and Motivation

A. SSI System and DID

Self-Sovereign Identity (SSI) aims to enable individuals to independently manage and control their digital identities [**?**]. Decentralized Identifiers (DIDs) implement the concept of SSI in a decentralized manner, with standards established by the W3C (World Wide Web Consortium). DIDs overcome the limitations of traditional centralized identity management systems, granting users full control over their identities.

B. W3C Verifiable Credential V2.0

The W3C Verifiable Credential Data Model (V2.0) [**?**] provides a mechanism for representing credentials such as driver's licenses, diplomas, and passports on the web in a cryptographically secure, privacy-respecting, and machine-verifiable manner. An example of a W3C Verifiable Credential using an *Embedded Proof*, one of its securing mechanisms, is shown below:


```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://www.w3.org/ns/credentials/examples/v2"
  ],
  "id": "http://vc.example/credentials/4643",
  "type": [
    "VerifiableCredential"
  ],
  "issuer": "https://issuer.example/issuers/14",
  "validFrom": "2018-02-24T05:28:04Z",
  "credentialSubject": {
    "id": "did:example:abcdef1234567",
    "name": "Jane Doe"
  },
  "proof": {
    "type": "DataIntegrityProof",
    "verificationMethod": "did:key:zUC7GsSgQ4NbriEoGBNeyC7WnG7c5McQMJryqVAK
J2nVgJWvWd3Vv4Ab81iJdDd5WBqSPDxwNjUedgkY2GbsJ5LcsFA5FM6Ub2eoUAnN9NbfZP89UYE
gvfrdurZofw2qfPAFzq6",
    "cryptosuite": "bbs-2023",
    "proofPurpose": "assertionMethod",
    "proofValue": "u2V0ChVhQiR6NOwNzbZYgP59zmtijxU0yht5cA8aiS0c50JdPzar9lIX
SfD6gQ1TXhDOEop2HHOL-Rn4BtrsKWu5PpYecz5j_kXDW7gv4ufcHhca7jnZYQGoquS0xvf4Mnm
ZoNkbVv5w8HN4gxs_sTcqvh3oJhCb2vqENcCm8D2khyMGr7-FGFdx818_ufbFmo8hKn_2FgMpYY
K5F3ceZ2_Sp8-fh6XzU7h6IO_rR82lnfEmpsNzTVnuS2lfrkjyvNUrJik-kyoecigyt0PDfU43v
VYs_mzcYjx2PjkSg_6KB0UFEXB1G6LE7Wxo4xdOnYmErU2Rih3SB6VggbsU9KqsBqZbwlwELyu5
h26lJahE9SSOyOa8h9YtizMWBZy9pc3N1ZXI"
  }
}
```


Fig. 1. W3C Verifiable Credential using an Embedded Proof

This data structure provides the basis for independently verifying the authenticity and integrity of a Verifiable Credential, ensuring the reliability of DID documents across diverse environments. The model supports features like *Selective Disclosure* and *Unlinkable Proof*, enabling privacy protection while offering verification capabilities through ZKP-based VC/VPs built on the BBS+ Scheme [**?**].

C. EIP712 and EdDSA Signature

EIP-712 [**?**] is a standard for structured data signing within the Ethereum ecosystem, enhancing security and enabling safe, consistent data signing across applications and protocols. By introducing a structured data standard instead of simple byte strings, EIP-712 ensures both multi-chain interoperability and security under Byzantine conditions. This standard utilizes a *domain separator* to prevent signature collisions and incorporates fields like *chain ID* to guarantee isolated signatures across chains and DApps.

D. ZKML and Infrastructure

Recent advancements have introduced infrastructures leveraging Zero-Knowledge Proofs (ZKPs) to simultaneously satisfy privacy protection and cryptographic integrity verification. In Zero-Knowledge Machine Learning (ZKML) research, methods have been proposed to verify machine learning inference results using ZKPs, establishing trust between clients and servers. Clients perform local model inference while maintaining data confidentiality and provide proofs to servers, which validate these to ensure computational integrity. Combined with technologies like *Multi-Party Computation (MPC)* and *Trusted Execution Environments (TEE)*, such architectures protect sensitive data and model weights, demonstrating potential as robust censorship models that meet both regulatory and privacy requirements.

E. ZK-SNARK Ecosystem

Various tools support the zk-SNARK ecosystem, including *Circom*, a constraint DSL, and the *SnarkJS Backend*. These tools facilitate circuit definition and proof generation for prominent ZKP systems such as *Groth16* and *Plonk*, making them a popular choice for research in zero-knowledge proofs.

F. BFT Consensus-Based Distributed Systems

Distributed systems based on Byzantine Fault Tolerance (BFT) consensus mechanisms form the backbone of trustless and decentralized environments. These systems ensure fault-tolerant and reliable operations, even in the presence of malicious actors, making them integral to decentralized architectures.

III. Trust Management System

A. Overview of Trust in SSI Systems

In Self-Sovereign Identity (SSI) systems, trust is a core element that ensures the security, privacy, and interoperability of identity data. However, most existing SSI solutions rely heavily on distributed ledgers or official issuers as trust anchors, which introduces the *Trust Anchor Problem*. This section analyzes this problem and proposes a novel trust management approach to address it.

Currently, SSI solutions predominantly assume credential issuers (e.g., government institutions) as "official" sources of trust. However, systematic trust management for personal issuers is lacking, creating challenges in scalability and decentralization.

B. Trust Anchor Problem

- **Centralized Trust Anchor**: Ledger-based SSI systems often overly depend on specific trust anchors, such as government agencies or issuer entities. For example, in public chain-based SSI, the issuer's signature and its

associated public key serve as the ultimate source of trust. Any malicious actions or mistakes by the issuer can compromise the entire system's reliability [**?**].

- **Governance Dependency**: Distributed Ledger-based DIDs depend on the trustworthiness of specific ledgers and their governance mechanisms, including consensus algorithms and policies. Issues with ledger governance or user preferences for alternative DID methods reveal a lack of adaptability in existing systems [**?**]. Ledger-based DIDs are "locked" to specific ledgers, making portability challenging in cases of disputes or new requirements.

Consequently, ledger-based DIDs exhibit limited flexibility when ledger trustworthiness is questioned or new authentication methods are required. To address these challenges, this study proposes an authentication mechanism leveraging a ZKP-based trust model to alleviate the Trust Anchor Problem and enhance interoperability.

C. ZKP Trust Model

Recent research, such as ZKML [**?**] and W3C Data Integrity BBS Cryptosuites v1.0 [**?**], has explored various trust models using Zero-Knowledge Proofs (ZKP). These models include mechanisms for censorship-resistant verification and privacy-preserving integrity checks.

- **ZKML (Client-Side Computation)**: By transforming inference functions of machine learning models into ZKPs, clients can perform inference locally and provide proof of results to servers without exposing sensitive data. This approach ensures both data confidentiality and inference integrity.
- **MPC and Secure Neural Network Inference**: Multi-Party Computation (MPC) and Trusted Execution Environments (TEE) enable sensitive data to be processed collaboratively without direct exchange, ensuring simultaneous data confidentiality and computation integrity.
- **Verifiable Credentials (BBS+ Scheme)**: The W3C BBS Cryptosuite supports selective disclosure and unlinkability, allowing clients to disclose only the necessary data while protecting the rest. Through ZKP, the authenticity of disclosed data can be verified without relying on ledgers, ensuring data integrity and privacy protection.

These ZKP-based infrastructures enable auditable integrity checks for private inputs. W3C's specifications for ZKP-based Verifiable Credential (VC) and Verifiable Presentation (VP) using BBS+ highlight the potential for combining ledger-based DID systems with externally extensible trust models.

D. Defining a Wallet-Based Trust Management System

This section defines key elements for a trust model that ensures authentication interoperability among issuers, provers, and verifiers. The proposed trust model assumes Byzantine conditions in multi-ledger environments and aims to reduce reliance on ledger-based SSI systems.

- **Auditable Data Structure**: A data structure is required to ensure the integrity of Verifiable Credential data even outside ledgers. This structure integrates cryptographic signatures within DID documents, enabling standalone cryptographic integrity verification. ZKP and scalability considerations are also incorporated.
- **Byzantine Fault Tolerance (BFT) and Anchor Independence**: The model assumes certain ledgers or issuers may act maliciously, emphasizing the need for independent management of trust anchors. While specific solutions are beyond the scope of this study, approaches like Attribute Aggregation in ATIB [**?**] can be applied to zk-Embedded Authentication, enabling governance-independent trust for diverse issuers.
- **Client-Side ZKP**: By securely managing VC data within hardware wallets and performing ZKP computations on the client side, privacy and security are enhanced. Users can directly generate VP proofs, ensuring reliable authentication.
- **Cross-Chain Verifiability**: By utilizing structured signature standards like EIP712 and zk-friendly elliptic curves such as BabyJubjub, the model ensures interoperability across multiple blockchain networks. This approach enables sustainable Verifiable Presentation verification despite governance issues or ledger dependencies.

This section highlights the proposed trust model components and solutions, particularly focusing on Client-Side ZKP. By reducing ledger dependencies and enabling cross-chain authentication in diverse environments, this trust model provides a foundation for sustainable authentication systems. The next section will present the detailed design and implementation strategies for these solutions.

IV. Wallet-Based zk-Embedded Authentication

This section defines the Wallet-Based zk-Embedded Authentication scheme for supporting DID authentication in a ledger-free environment. The scheme manages Verifiable Credentials (VCs) in hardware wallets (or local wallets), structures attribute integrity using Sparse Merkle Trees (SMTs), and ensures trust by signing the Merkle root using BabyJubjub signatures from the issuer. Additionally, it supports privacy-preserving verification using Zero-Knowledge Proofs (ZKPs) with selective disclosure.

A. zk-Embedded Authentication

This section presents the concept of Wallet-Based Embedded Authentication and its implementation. This approach securely stores VCs in the user's hardware wallet and establishes trust using ZKP-based authentication. It supports cross-chain environments while ensuring interoperability and trust without relying on ledgers.

B. Overview

To address the ledger dependency and trust anchor issues in existing DID authentication, this study proposes an auditable and interoperable authentication approach that operates independently of external ledgers. The core idea involves defining a scheme between the Issuer, Prover, and Verifier as follows:

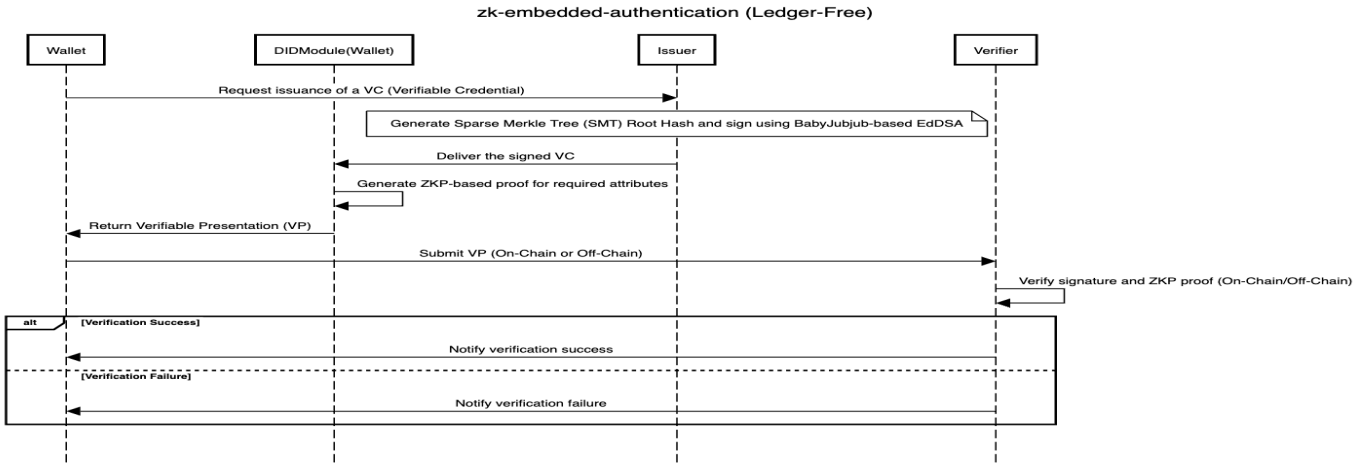

Fig. 2. zk-Embedded Authentication. This figure defines embedded verification among the issuer, prover (wallet user), and verifier under a trustless assumption. The scheme issues VCs to the prover, who manages them in a DID module and generates VP proofs through circuits. Proofs are verified via smart contracts or off-chain computation.

- **SMT-based VC Structuring**: Attributes within the Verifiable Credential are inserted into a Sparse Merkle Tree and aggregated into a single Merkle root.
- **BabyJubjub Signature**: The issuer authenticates the SMT root using BabyJubjub-based EdDSA signatures.
- **ZKP-based Selective Disclosure**: The prover (wallet user) uses Zero-Knowledge Proofs to demonstrate satisfaction of specific attribute conditions without exposing the entire VC, thus protecting privacy.

This scheme offers the following features:

- **Ledger Independence**: It minimizes reliance on external ledgers by verifying VCs through SMT root signatures and ZKP verification.
- **Selective Disclosure & Privacy**: Demonstrates specific attribute conditions (e.g., age $\geq$ 18) using ZK-SNARKs.
- **Cross-Chain & Interoperability**: Utilizes structured signature standards like EIP712 to support authentication across multi-chain environments.
- **Auditable & Extensible**: Employs auditable data structures based on SMT and embedded proofs.

C. Sparse Merkle Tree (SMT) & BabyJubjub-Based EdDSA Signature Scheme

This study leverages zk-friendly signatures based on the BabyJubjub curve and the Sparse Merkle Tree data structure as an auditable data structure to ensure the integrity and privacy of Verifiable Credentials. These are compatible with existing zk-SNARK cryptographic frameworks.

Each attribute $attr_i$ is processed using a ZK-friendly hash function H, such as Poseidon, and stored as a leaf node:

$$Leaf(attr_i) = H(attr_i)$$

All attribute leaves are combined upwards through hashing to obtain the final Merkle root. This Merkle root represents a cryptographic summary of all VC attributes, allowing easy generation of inclusion/exclusion proofs through the SMT structure. The issuer signs the SMT root using BabyJubjub-based EdDSA, leveraging the zk-SNARK-friendly elliptic curve and its compatibility with the Poseidon hash function.

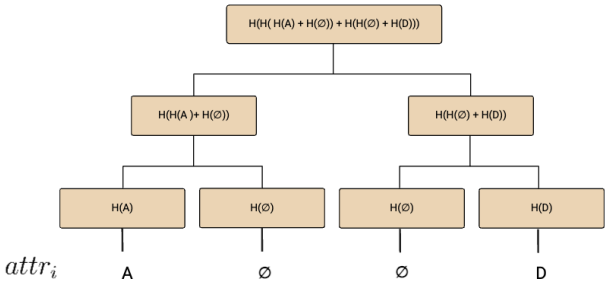

Fig. 3. SMT is a hash-based tree structure that efficiently provides inclusion and exclusion proofs over a large key space. As shown, all leaf nodes maintain a fixed height and accommodate sparse key spaces. Keys for specific attributes (e.g., `age:000`, `name:010`) are fixed, allowing for inclusion and exclusion proofs commonly used in membership verification.

Using the issuer's secret key sk_I and public key pk_I:

$$\sigma_I = \text{Sign}_{sk_I}(root)$$

The generated signature σ_I and Merkle root are included in the proof field of the VC. This enables the prover to verify that the VC was legitimately issued by the issuer using pk_I, the SMT root, and σ_I, independent of external ledgers and compatible with zk-SNARK cryptographic frameworks.

Fig. 4 illustrates an example VC issued under this scheme. The attributes are extracted from the DID document, inserted into the SMT, and the Merkle root is signed.

D. Selective Disclosure with ZKP

The prover generates Zero-Knowledge Proofs using Circom and SnarkJS within the wallet. All witnesses (VC attributes, SMT inclusion proofs, issuer signatures, etc.) are processed locally, and only the ZKP π is shared with the verifier.

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.example.org/examples/v1"
  ],
  "id": "http://chungnam.ac.kr/credentials/3732",
  "type": [
    "VerifiableCredential",
    "AlumniCredential"
  ],
  "issuer": {
    "id": "https://infosec.chungnam.ac.kr",
    "name": "Chungnam National University Information Security Lab",
    "publicKey": {
      "Ax": "13277427435165878497778222415993513565335242147425444199013288855685581939
      "Ay": "13622229784656158136036771217484571176836296686641868549125388198837476602
    }
  },
  "issuanceDate": "2024-02-11T09:30:24Z",
  "credentialSubject": {
    "id": "did:example:abcdef1234567890",
    "name": "ham3798",
    "age": 25,
    "studentNumber": "201902769",
    "alumniOf": {
      "id": "did:example:c34fb4561237890",
      "name": "Chungnam National University",
      "department": "Information Security Lab"
    }
  },
  "proof": {
    "type": "BabyJubJubSMTSignature2024",
    "created": "2024-10-08T05:59:43.348Z",
    "proofPurpose": "zk-Embedded-Authentication",
    "verificationMethod": "https://infosec.chungnam.ac.kr",
    "merkleRoot": "176,113,248,159,95,186,151,221,221,94,150,40,177,156,132,61,209,46,5
    "signature": {
      "R8x": "14843327923499602530760840822364653819560149483324056861844414827173045
      "R8y": "16624741311753369209800270717954129820251768033844592664257222687029635
      "S": "20033401010550130937315327336029792718861572862925232699317496954120079912
    }
  }
}
```


Fig. 4. This document extends the W3C Verifiable Credential's embedded proof security policy by defining a Snarkifiable VC scheme through BabyJubjub curve-based signing of Sparse Merkle Tree root hashes. The VP scheme transforms the VC for specific attribute proofs. It supports interoperability through standardized data signature definitions, such as EIP712.

```
pragma circom 2.0.0;

include "/circomlib/circuits/eddsaposeidon.circom";
include "/circomlib/circuits/comparators.circom";
include "/circomlib/circuits/smt/smtverifier.circom";

// Main circuit: includes all SMT proofs, signature verification, age check
template MainCircuit(nLevels) {
    ...

    component smtVerifierAge = SMTVerifier(nLevels);
    smtVerifierAge.enabled <== enabled_age;
    smtVerifierAge.root <== root;
    smtVerifierAge.siblings <== siblings_age;
    smtVerifierAge.oldKey <== oldKey_age;
    smtVerifierAge.oldValue <== oldValue_age;
    smtVerifierAge.isOld0 <== isOld0_age;
    smtVerifierAge.key <== key_age;
    smtVerifierAge.value <== value_age;
    smtVerifierAge.fnc <== fnc_age;

    ...

    // EdDSA Verification Component
    signal input enabled_eddsa;
    signal input Ax;
    signal input Ay;
    signal input R8x;
    signal input R8y;
    signal input S;
    signal input M; // Use root hash as message

    component eddsaVerifier = EdDSAPoseidonVerifier();
    eddsaVerifier.enabled <== enabled_eddsa;
    eddsaVerifier.Ax <== Ax;
    eddsaVerifier.Ay <== Ay;
    eddsaVerifier.R8x <== R8x;
    eddsaVerifier.R8y <== R8y;
    eddsaVerifier.S <== S;
    eddsaVerifier.M <== M;

    component ageCheck = GreaterEqThan(32);
    ageCheck.in[0] <== value_age; // Age value from SMT proof
    ageCheck.in[1] <== 25;
    ageCheck.out === 1;
}

component main = MainCircuit(64);
```

Fig. 5. VP Circuit. This circuit, implemented in Circom, includes SMT inclusion proofs, signature validity checks, and proofs of attributes such as age and affiliation. Using this circuit composition, various proofs for Verifiable Credentials can be generated.

Various arithmetic circuits can be combined to enable different types of proofs. This study focuses on a simple proof example: demonstrating that the prover's age is ≥ 18. The prover generates a ZKP satisfying the following circuit conditions:

$$\begin{aligned} SMT.\text{InclusionProof}(root, key_{\text{age}}, value_{\text{age}}) &= \text{True} \\ \text{VerifySign}(pk_I, root, \sigma_I) &= \text{True} \\ value_{\text{age}} &\geq 18 \end{aligned}$$

The prover inputs their actual age, SMT proof, signature, etc., as witnesses and generates π by invoking:

$$\text{Prove}(provingKey_{snark}, C, witness)$$

The verifier validates the conditions using π and public inputs (e.g., issuer public key, minimum age condition) without accessing private data.

E. zk-Embedded Authentication

This section presents the concept of Wallet-Based Embedded Authentication and its implementation. This approach securely stores VCs in the user's hardware wallet and establishes trust using ZKP-based authentication.

It supports cross-chain environments while ensuring interoperability and trust without relying on ledgers. This approach securely stores VCs in the user's hardware wallet and establishes trust using ZKP-based authentication. It supports cross-chain environments while ensuring interoperability and trust without relying on ledgers.

CONCLUSION

This study demonstrated the feasibility of a ledger-free DID (Decentralized Identifier) authentication scheme that leverages Zero-Knowledge Proof (ZKP) to achieve interoperability, privacy, and trust without relying on a specific ledger or trust anchor. By employing Sparse Merkle Trees (SMTs) for structured data integrity and BabyJubjub-based signatures for Verifiable Credentials (VCs), managed directly within a hardware wallet, the proposed approach enables selective disclosure of attributes through ZKP proofs. Verification can occur either on-chain or off-chain, offering flexibility and compatibility with various governance models and implementations. This approach overcomes limitations of existing ledger-dependent DID methods and introduces a more adaptable and scalable

trust model, paving the way for more robust and flexible SSI (Self-Sovereign Identity) ecosystems.

REFERENCES

1) A. Satybaldy, M. S. Ferdous, and M. Nowostawski, "A Taxonomy of Challenges for Self-Sovereign Identity Systems," *IEEE Access*, vol. 12, pp. 16151-16177, 2024, doi: https://doi.org/10.1109/ACCESS.2024.335794010.1109/ACCESS.2024.3357940.
2) T. Zhong, P. Shi, and J. Chang, "JointCloud Cross-chain Verification Model of Decentralized Identifiers," in *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, Austin, TX, USA, 2021, pp. 1-8, doi: https://doi.org/10.1109/IPCCC51483.2021.967936310.1109/IPCCC51483.2021.9679363.
3) L. Xia *et al.*, "DIDAPPER: A Practical and Auditable On-Chain Identity Service for Decentralized Applications," in *2023 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*, Athens, Greece, 2023, pp. 151-157, doi: https://doi.org/10.1109/DAPPS57946.2023.0002810.1109/DAPPS57946.2023.00028.
4) A. Grüner, A. Mühle, and C. Meinel, "ATIB: Design and Evaluation of an Architecture for Brokered Self-Sovereign Identity Integration and Trust-Enhancing Attribute Aggregation for Service Provider," *IEEE Access*, vol. 9, pp. 138553-138570, 2021, doi: https://doi.org/10.1109/ACCESS.2021.311609510.1109/ACCESS.2021.3116095.
5) Z. A. Lux, D. Thatmann, S. Zickau, and F. Beierle, "Distributed-ledger-based Authentication with Decentralized Identifiers and Verifiable Credentials," in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, IEEE, 2020, pp. 71–78, doi: https://doi.org/10.1109/BRAINS49436.2020.922325610.1109/BRAINS49436.2020.9223256.
6) World Wide Web Consortium (W3C), "Verifiable Credentials Data Model v2.0," 2023. [Online]. Available: https://www.w3.org/TR/vc-data-model-2.0/. [Accessed: Dec. 7, 2024].
7) World Wide Web Consortium (W3C), "Data Integrity BBS+ Signature Suite v2023," 2023. [Online]. Available: https://www.w3.org/TR/vc-di-bbs/. [Accessed: Dec. 7, 2024].
8) R. Mukta, H. Paik, Q. Lu, and S. Kanhere, "Credential-based Trust Management in Self-Sovereign Identity," in *ACM WomenEncourage 2021*. [Online]. Available: https://womencourage.acm.org/2021/wp-content/uploads/2021/07/87_extendedabstract.pdf. [Accessed: May 3, 2023].
9) M. Kubach and H. Roßnagel, "A Lightweight Trust Management Infrastructure for Self-Sovereign Identity," in *Open Identity Summit 2021*.
10) M. S. Ferdous, F. Chowdhury, and M. O. Alassafi, "In Search of Self-Sovereign Identity Leveraging Blockchain Technology," *IEEE Access*, vol. 7, pp. 103059–103079, 2019, doi: https://doi.org/10.1109/ACCESS.2019.293117310.1109/ACCESS.2019.2931173.
11) Ethereum Foundation, "EIP-712: Ethereum Typed Structured Data Hashing and Signing," 2017. [Online]. Available: https://eips.ethereum.org/EIPS/eip-712. [Accessed: Dec. 7, 2024].