Bringing Identity from Web2 to Web3: Zero-Knowledge Address Abstraction

Sanghyeon Park
Seoul National University
Seoul, Republic of Korea
lukepark@snu.ac.kr

Jeong Hyuk Lee
Hanyang University
Seoul, Republic of Korea
ahoo791@hanyang.ac.kr

Seunghwa Lee
Kookmin University
Seoul, Republic of Korea
ttyhgo@kookmin.ac.kr

Jung Hyun Chun
Hanyang University
Seoul, Republic of Korea
jungdev@hanyang.ac.kr

Hyeonmyeong Cho
UNIST
Ulsan, Republic of Korea
heyitsme@unist.ac.kr

MinGi Kim
Sungkyunkwan University
Seoul, Republic of Korea
lupin1001@skku.edu

Hyun Ki Cho
Sogang University
Seoul, Republic of Korea
blockchain@sogang.ac.kr

Soo-Mook Moon
Seoul National University
Seoul, Republic of Korea
smoon.snu.ac.kr

***Abstract*—The use of external identities in decentralized systems is a significant area of interest within the research community. Despite progress being made in this field, current solutions are limited by several challenges such as privacy leaks, blockchain-specific fragmented identifiers, and the requirement for mapping between web2 and web3 identities.**

To address these limitations, this paper introduces zero-knowledge address abstraction (zkAA), a new approach to incorporating external identities into decentralized systems in a privacy-preserving manner. Our solution leverages the zk-SNARKs to secure JSON Web Tokens (JWTs), which are used for user identification. The primary contribution of this paper is the introduction of the address abstraction identity scheme, which replaces traditional blockchain-specific identities and signatures with a unified solution that allows users to maintain a single identity that transcends different blockchain address systems and can be used seamlessly across both web2 and web3 platforms. The zkAA solution leverages existing web2 identity solutions and is implemented as a smart contract on top of existing blockchain infrastructure. This enables easy integration with existing systems without the need for a disruptive and complex hardfork process.

We implement the solution on aptos blockchain, and the implementation shows efficiency to be applied to web2-web3 hybrid dApps.

***Index Terms*—identity management; blockchain; privacy-preserving; zero-knowledge proofs;**

I. Introduction

The rapid growth of blockchain technology has led to the massive emergence of decentralized applications (dApps). Despite their decentralized nature, dApps often require centralized authentication methods, such as social login or know-your-customer (KYC) processes, to enhance efficiency and mitigate malicious activities, such as Sybil attacks. This results in the need for dApps to map external identities to blockchain addresses, which can lead to privacy leaks and difficulties in managing multiple blockchain-specific addresses, especially for multi-chain dApps.

To address these challenges, we present zkAA, a zero-knowledge address abstraction solution that utilizes JSON Web Tokens (JWTs) for identification on the blockchain. JWTs, issued and verified by trusted institutions, offer a user-friendly and efficient solution for identity verification without mapping to blockchain addresses. The use of zero-knowledge proofs and zkSNARKs ensures the privacy and security of JWTs. zkAA enables inter-compatibility among multi-chain and web2-web3 hybrid dApps, providing an effective way to identify users for identity management. In a nutshell, zkAA offers a novel approach to incorporating external identities into blockchain systems by using JWTs instead of mapping to blockchain addresses. This eliminates the need to store private keys and provides an effective way to identify users for identity management, enabling the development of multi-chain and web2-web3 hybrid dApps. Table I provides a comparison between the traditional identifier —blockchain address— and the new identifier $H(\text{JWT})$ —commitment of JWT, proposed by zkAA—.

zkAA is implemented as a smart contract on top of existing blockchain infrastructure, eliminating the need for a hardfork process and making it easy to integrate with existing systems and applications.

The focus of this paper is to present a unique approach to address abstraction in blockchain systems, utilizing Zero-knowledge Proofs. The key contributions of this paper are as follows:

- The new identity scheme, address abstraction, is introduced to break away from the traditional blockchain-specific identity and signature system.
- The paper presents a new solution for address abstraction, named zkAA, which utilizes Zero-knowledge Proofs to ensure strong privacy guarantees and practical usage.
- The feasibility of integrating zkAA with multi-modal decentralized applications, such as multi-chain and web2-web3 hybrid applications, is examined to provide a seamless user experience in blockchain systems.
- The implementation and evaluation of zkAA on the Ethereum blockchain are conducted, along with a thorough analysis of its performance and efficiency.

TABLE I: Comparison of Address and $H(\text{JWT})$ as Identifiers on the Blockchain

	Definition	Purpose	Issuing Authority
Address	A public identifier used for sending and receiving on-chain transactions.	To provide a way of identifying and managing digital data (including assets) in a blockchain.	Generated by users and not issued by any central authority.
$H(\text{JWT})$ in zkAA*	A solution that uses JSON Web Tokens (JWTs) as a form of identification on the blockchain.	To address the challenges present in using existing external identities, such as difficulty in mapping them into blockchain-specific identities and lack of privacy.	JWTs can be issued and verified by trusted Web2 certificate authorities.

	Underlying Cryptography	Secret	Chain-Specific	Cost
Address	Public Key Cryptography (& Hash)	Private Key	O	Typically low, with only minimal fees required for transaction processing.
$H(\text{JWT})$ in zkAA*	Zero-Knowledge Proofs & Hash	JSON Web Token	X (Identifiers are same across all chains)	Depends on the specific implementation, but in general, the use of ZKPs increases the cost.

* Zero-Knowledge Address Abstraction.

The organization of this paper is as follows: In Section II, we present a comprehensive background on JWT, zero-knowledge proofs, and related concepts to provide a clear understanding of the material presented in the paper. In Section III, we describe the design and implementation of a naïve solution for address abstraction and analyze its limitations. In Section IV, we introduce our proposed solution, Zero-knowledge Address Abstraction (zkAA), which leverages the power of zero-knowledge proofs to overcome the limitations of the naïve solution. We also present three methods for registering JWT-driven identities, each with its trade-off between cost-efficiency and the degree of abstraction. In Section V, we provide a sequence diagram that outlines the potential applications of zkAA. In Section VI, we discuss the potential applications of zkAA in various use cases. In Section VII, we review the related work in the field of address abstraction on the blockchain. Finally, in Section VIII, we summarize our proposed scheme, highlight its strengths and limitations, and provide directions for future work.

II. Background

A. JSON Web Tokens

JSON Web Tokens (JWTs) are widely used open standards (RFC 7519) for securely transmitting claims between parties. They are commonly used for authenticating users and passing user information between systems, especially as access tokens in API authentication. The JWT is comprised of three distinct parts: a header, a payload, and a signature. The header contains metadata about the token, such as the type and signing algorithm. The payload contains the claims, which are statements about the user entity and additional data. The signature serves to verify the integrity of the claims. JWTs are signed using public/private key cryptography algorithms to ensure their authenticity and integrity. Elliptic curve-based algorithms, such as EdDSA, are particularly secure and efficient for this purpose. The signature created through this process confirms that only the entity holding the private key could have signed the token, thereby certifying the authenticity of the source of the JWT.

It is worth noting that while JWTs are used as a representative of web2 identity in this paper, the proposed zero-knowledge address abstraction scheme can also be applied to other web2 identities that can be verified by an institution's distinct public information, such as a public key.

B. Blockchain and Smart Contracts

In the realm of decentralized systems, the concept of a blockchain and smart contracts have gained significant traction as innovative solutions for secure and efficient data storage and contract execution. A blockchain is a distributed ledger that securely records transactions across a network of nodes using cryptographic techniques. The integrity of the data is maintained through the creation of blocks that are linked together to form a chain, making it resistant to modification and fraud.

Smart contracts, which are stored on the blockchain network, provide a means for automatically executing predetermined actions when specific conditions are met, thereby reducing the need for intermediaries and increasing efficiency. Ethereum [12] is a prominent example of a blockchain platform that enables the creation of smart contracts and decentralized applications (dapps). The Ethereum Virtual Machine (EVM) provides a decentralized execution environment for smart contract scripts, utilizing a global network of public nodes. The network operates using the cryptocurrency Ether (ETH) for internal transactions, which can also be used to pay for transaction fees and computational services.

C. Decentralized Identifier

Decentralized Identifiers (DIDs) are unique identifiers for entities that are created, stored, and managed in a decentralized manner, without relying on centralized authorities or databases. Blockchain technology is often used to implement DIDs, providing a secure and tamper-proof mechanism for storing and managing data. Two examples of decentralized identity management systems that run on the blockchain are Proof of Attendance Protocol (POAP) and SoulBound Tokens (SBTs).

POAP is a platform that uses blockchain technology to create collectible tokens representing personal experiences, while SBTs are Non-Fungible Tokens that serve as a fundamental building block in the Decentralized Society trend of Web3. However, these techniques are based on the Web3 identifier, the blockchain address. Mapping between Web2 and Web3 identities is required if a Web2 identity is to be used on a

Web3 platform. In the case of our study, zkAA, the Web2 identity is directly used on the blockchain without mapping to a Web3 identity.

D. Account Abstraction

In the Ethereum blockchain, Account Abstraction is a technique that aims to unify External Owned Accounts (EOAs) and Contract Accounts (CAs). EOAs are traditional blockchain accounts controlled by a private key, enabling users to sign transactions and interact with the network. On the other hand, CAs do not have a direct private key but possess programmable logic and storage. Account Abstraction enables seamless integration of the features of both EOAs and CAs, allowing for programmability features in EOAs and signing capabilities in CAs. This integration can be achieved either through a hardfork on the Ethereum network or a contract-based wallet solution without the need for a hard fork.

In contrast, our proposed Address Abstraction is a new concept in blockchain technology that abstracts the identifier of a user from the conventional blockchain address to a JWT-based encrypted identifier. This simplifies identity management and enhances inter-compatibility across various blockchains, as users can engage with the blockchain using a single credential represented as a hash of a JWT, without being limited by the underlying blockchain address system that may vary from one blockchain to another.

E. Zero-Knowledge Proofs

The field of cryptography and blockchain technology has seen a surge in interest in Zero-Knowledge Proofs (ZKPs) as a solution to privacy and security challenges. Zero-knowledge proofs were first introduced in 1985 [2] as a mechanism for verifying the authenticity of information without requiring the revelation of the underlying data. Since the initial introduction of zero-knowledge proofs, the field has undergone significant advancements, including the development of zero-knowledge Succinct Non-Interactive Arguments of Knowledge (zkSNARKs). This technology allows for the efficient and succinct verification of information, making it an ideal solution for blockchain-based systems. A popular tool used in the development of zero-knowledge proofs is ZoKrates, a toolkit for writing, compiling, and executing ZKP-based smart contracts on the Ethereum blockchain.

III. Naïve Approach to Address Abstraction

The design of Address Abstraction is a pioneering approach to representing identifiers in blockchain that aims to simplify identity management by abstracting away the underlying cryptography. The concept of address abstraction is based on the utilization of JWT (JSON Web Token)-based encrypted identities. By applying a hash function, the JWT-based identity can be represented as a unique, secure and concise identifier that can be used as the user's identifier on the blockchain. This innovative approach offers a simple and user-friendly solution to the challenge of identity management on the blockchain, thereby making it an important contribution to the field.

A. Hash-based Approach

The hash-based approach to address abstraction is a technique that utilizes the properties of hash functions to generate a unique and secure identifier. This method involves the use of a JSON Web Token (JWT) and a hash function to generate a one-time identifier, represented as $H(\text{JWT})$. The generated hash value is used as the identifier on the blockchain, however, it is consumed through the revelation of the JWT.

This approach is based on the commit-and-reveal scheme, where the user initially commits to a value by providing its hash and later reveals the original value. It is important to note that this method is suitable for cases where the identity represented by the JWT does not need to be kept private and can be used for a single transaction. It does not offer any privacy-preserving properties and has limited continuity as the JWT contents are revealed during the reveal phase.

B. Using Merkle Tree

The Merkle tree structure provides another alternative for address abstraction. In this method, the root of the Merkle tree and the corresponding Merkle proofs are utilized to reveal the JSON Web Token (JWT) at a specified time n. This approach enables a limited number of usages for identities, unlike the hash-based approach. However, it is important to note that, like the hash-based approach, this method does not offer privacy-preserving properties for the user's identities.

IV. Design of Zero-Knowledge Address Abstraction

Identity management is a crucial aspect of decentralized systems, particularly blockchain networks. One of the challenges in managing identities is balancing privacy and security with efficiency and usability. The use of hash-only address abstraction methods, such as commit-and-reveal schemes, cannot provide a degree of privacy, as the contents of the JSON Web Token (JWT) are revealed during the *reveal* phase. Also, it is limited in that they can only be used for single-use or a publically registered identifier, as the JWT is revealed during the reveal phase, rendering it unusable for future transactions. Although Merkle tree methods provides limited, constant use of identifiers stored in the blockchain, this approach is still not suitable for unlimited use of identities without revealing the JWT.

If the requirement is for unlimited usage of identities and the preservation of privacy for the JWT contents, zero-knowledge proofs (ZKP) are necessary. ZKP offers a secure and privacy-preserving approach to verify the validity of a JWT, while keeping its contents hidden from view.

A. Setup, Certificate, Register, and Publish Phases

The zkAA protocol is a specific implementation of address abstraction that leverages the power of zero-knowledge proofs (ZKP) to provide privacy and security for registering identities on a blockchain network. In the zkAA system, a user's identity is represented by a JSON Web Token (JWT) that is hashed to produce the user's unique identifier, represented as $H(\text{JWT})$.

The ZKP is designed to verify the authenticity of the JWT, proving that the user holds the preimage corresponding to the $H(\text{JWT})$, without revealing the contents of the JWT. This results in a high degree of privacy, as the contents of the JWT remain confidential during all phases. The ZKP proof is verified on the Ethereum blockchain, providing a decentralized, tamper-proof way to manage and verify user identities. The use of zkSNARKs also ensures that the proof process is highly efficient and succinct, making the zkAA protocol well-suited for use on blockchain networks.

The zkAA protocol comprises four phases: Setup, Certificate, Register, and Publish, which are depicted in Figure 1.

0) Setup phase: Two trusted setup in zksnarks scheme. One is for JWT registration circuit, and another is for circuit which is using registered hash of JWT without revealing JWT.
1) Certificate phase: User requests JWT from institute such as Google, Facebook, Discord, etc.
2) Register phase: User calculates Hash of JWT and creates a proof to demonstrate the Hash H^k is calculated from JWT and that JWT is signed from institute, without revealing JWT. Then, user publishes H^k and proof on-chain and the zkAA contract verifies them. If valid, the contract submits H^k into the contract, which can be used as an identity.
3) Publish phase: User can now publish abstracted transactions freely without regard to the sender, as long as they have a valid JWT and can create the corresponding H(JWT) (which should match H^k). The abstract transaction includes a proof field which is verified in the zkAA contract to ensure that the sender is the legitimate owner of H^k identity, and then the original transaction is executed with the target address, function signature, calldata, and value.

Setup Phase. The Setup Phase is an important step in providing trustworthy identities in the Zero-knowledge Address Abstraction (zkAA) system. This phase involves the generation of two Common Reference Strings (CRSs), each of which is a unique pair of (proving key, verification key). In order to ensure the trustworthiness of the CRSs, it is recommended to use a Multi-Party Computation (MPC) protocol to remove any potential toxic waste that could result in fake proofs. This decentralizes the trusted setup process and ensures that, as long as at least one participant is honest, the toxic waste can be removed. To further reinforce the trustworthiness of the setup phase, all participants are encouraged to participate.

The CRSs are dependent on the circuit and thus, two CRSs are required in zkAA - one for the registration circuit (crs_R) and another for the circuit that uses the registered hash value to publish abstracted transactions (crs_P). These CRSs can be generated more trustworthy, based on decentralized parameters from the ongoing Perpetual Powers of Tau[] in Ethereum, with some additional circuit-specific procedures required.

Certificate Phase. The certificate phase involves the acquisition of a JSON Web Token (JWT) from an institute, such as Google, Twitter, or others. In this phase, the user initiates a request for a JWT from the institute. Upon successful authentication, the institute issues the requested JWT to the user. This phase has proceeded on the off-chain between the user and the institute.

Register Phase. The Register phase of the system is a crucial step in ensuring the validity of the identities that are being recorded in the blockchain. In this phase, the following verifications are carried out: Verification of Uniqueness, Verification of Hash, and Verification of JWT Signatures. To verify the uniqueness of H^k, the system checks that the same value is not already registered in the blockchain. This ensures that each identity has a unique representation in the system. Then the system verifies that the corresponding H^k value is indeed a hash made from the original JWT and that the JWT has been signed by the institute through zero-knowledge proof, therefore these verifications are done without revealing the JWT itself. The system uses the public key of the institute to verify the zero-knowledge proof. This verification is done on-chain and ensures that the proof has been verified as true. In this paper, it is assumed that the correct public key is registered through reliable Oracle techniques such as governance or majority voting.

If the above steps are successfully verified, the H^k value is then registered in one of three ways: (1) registering H^k as a standalone value, (2) mapping the smart contract wallet with H^k, or (3) NFT-izing H^k and assigning its ownership to an address. In the case of options 1 and 2, a nonce, a unique monotonic increment counter, is initialized to zero for each identity managed by the smart contract to maintain the sequence of abstracted transactions.

In the Zero-knowledge Attribute-based Authentication (zkAA) scheme, the validity of the JSON Web Token (JWT) is verified during the Register Phase, as opposed to the naive Attribute-based Authentication (AA) implementation where JWT validation occurs during the Publish Phase. This forward-awareness of JWT authenticity in the zkAA scheme provides the added advantage for dApps to offer user-specific services that rely on zkAA-driven identities in advance. Moreover, this approach also protects the privacy of JWT, unlike solely hash-based AA implementations.

Publish Phase. The message msg is composed that specifies the intended actions in the transaction. The inclusion of msg within a proof protects the intended actions from being altered, even if the proof is exposed to potential attacks or front-runner scenarios. However, censorship cannot be completely prevented and alternative methods for avoiding censorship should be considered.

To create a valid proof, the nonce must start from 1 and increment by 1 for each transaction. The nonce is initialized to 0 during the registration phase. The proof must be generated to demonstrate that the user has knowledge of the original preimage, which corresponds to the registered hash value in the contract. If the proof is successfully verified as true, the actions associated with msg will be executed. A non-zero amount of native asset, $value$, may be required to be sent together through the abstracted transaction.

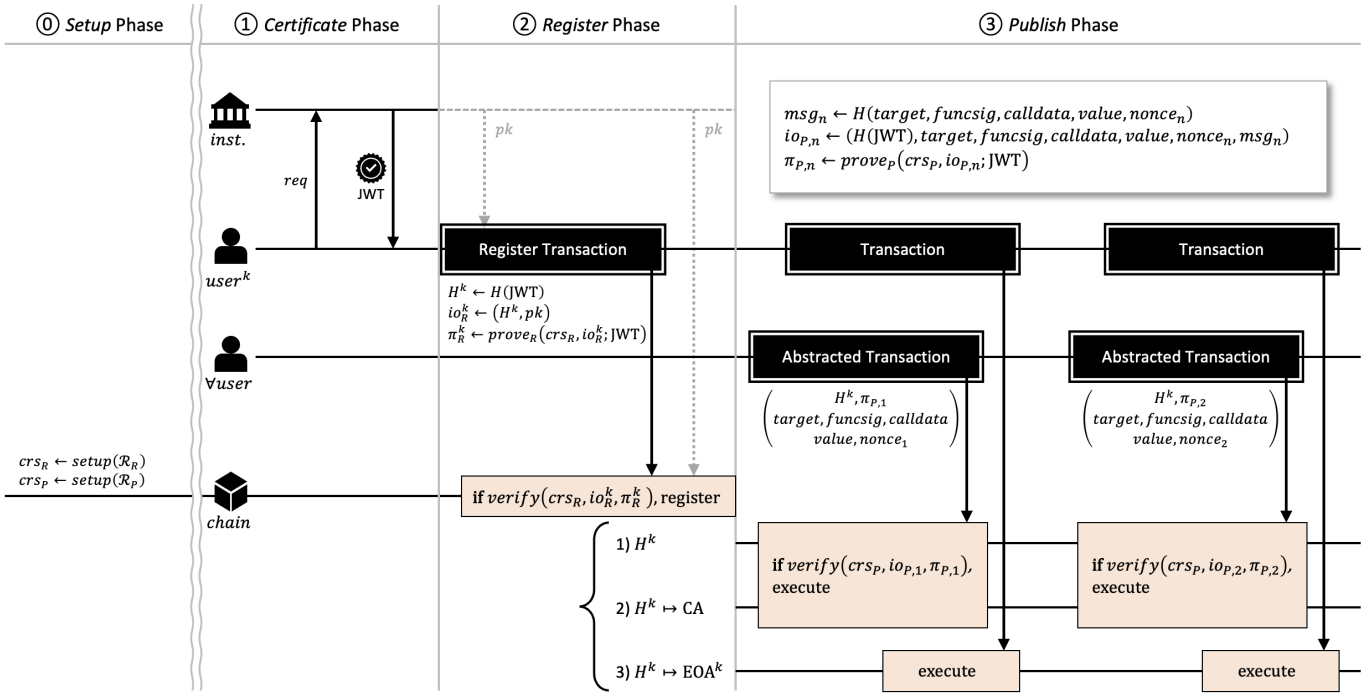

Fig. 1: Zero-Knowledge Address Abstraction

One of the main disadvantages of using zkp for address abstraction is the high computational cost of proof verification on Ethereum. This can lead to high gas costs for transactions, which can make it less practical for use in certain situations or for certain types of users. Additionally, while zkp can provide a high level of privacy and security, they can also add complexity and overhead to the system, making it more difficult to implement and use. Despite these drawbacks, the ability to use external identities on the blockchain without revealing private information is considered a valuable feature, and the use of zkp in zkAA is considered a strong solution for addressing this need.

B. Identifiers

In zkAA, there are three distinct methods for creating user identities, each with its own advantages and trade-offs. These methods include submitting H(JWT), deploying a contract wallet, and creating an NFT.

1) Hash: This method involves submitting the hash of a JSON Web Token (JWT) to the zkAA contract on the blockchain. The hash, referred to as H(JWT), serves as a unique identifier for the user and can be used to perform transactions while preserving the user's anonymity.

2) Contract Wallet: In this approach, the user creates a new contract wallet on the blockchain and maps the address of the contract to the H(JWT) identity. This allows the user to transact on the blockchain using the contract wallet, thereby maintaining anonymity.

3) NFT: The user can also create a non-fungible token (NFT) that represents the H(JWT) identity. This NFT can then be associated with a specific address and used as an identifier for the user, such as an SBT or POAP. This method provides a unique and verifiable identity on the blockchain while preserving the user's anonymity.

V. Applications

A. Using Hash Value as Identity

In this paper, we propose zkAA, a zero-knowledge proof-based authentication and authorization scheme for decentralized applications (dApps) on the blockchain. zkAA allows users to authenticate and authorize themselves on a dApp without revealing their identity, while also ensuring that the dApp can trust the authenticity of the user's identity.

One potential application of zkAA is in the context of social login for web3 games. A user could navigate to a game's website and click on the "login" button, which would redirect them to a social login provider (such as Google or Discord) to authenticate. Upon successful authentication, the user would be redirected back to the game's website with a JSON Web Token (JWT) that contains information about their identity. The user's browser would then calculate the hash of the JWT (H(JWT)) and create a zkp proof to demonstrate that the JWT was signed by the social login provider, without revealing the JWT. This proof, along with the H(JWT), would then be submitted to the zkAA contract on the blockchain. The contract would verify the proof and, if it is valid, associate the H(JWT) with the user's identity. The game's website could then use the H(JWT) as an anonymous identity for the user, allowing them to play the game and interact with other players while maintaining their privacy. The user could also use this H(JWT) identity to interact with other dApps and make transactions on the blockchain network.

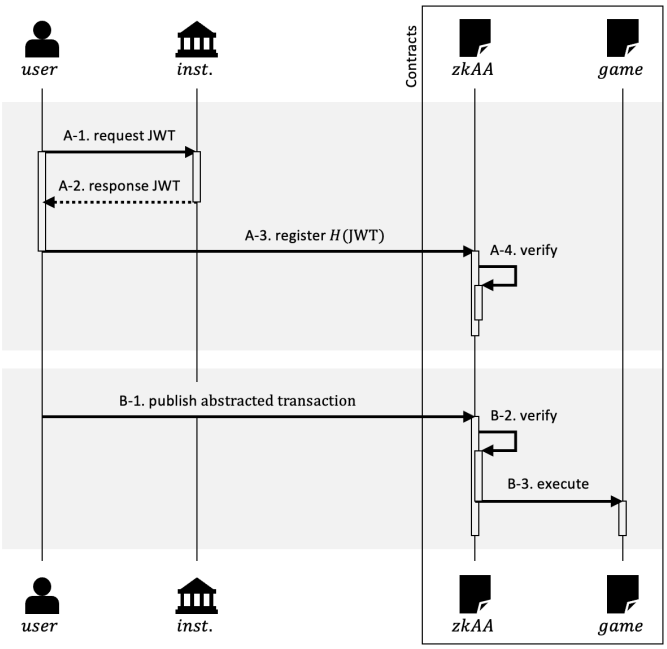

Fig. 2: Using $H(\text{JWT})$ as identity.

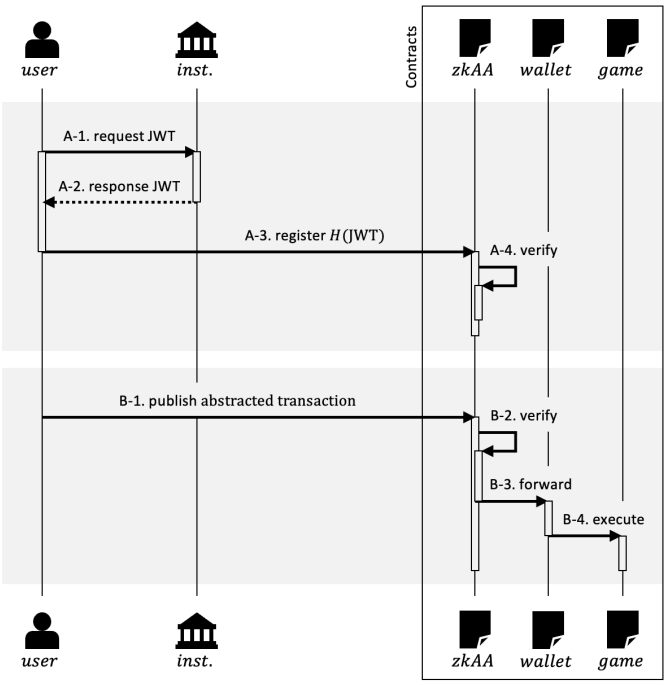

Fig. 3: Mapping $H(\text{JWT})$ to contract wallet.

B. Mapping Hash Value to Contract Wallet

Another potential application of zkAA is in the context of mapping an H(JWT) identity to a contract wallet on the blockchain. The user would first log into a social media platform, such as Google, Twitter, or Discord, using their existing credentials. The social media platform would issue a JWT to the user, which would include their unique identity and other relevant information. The user would then submit

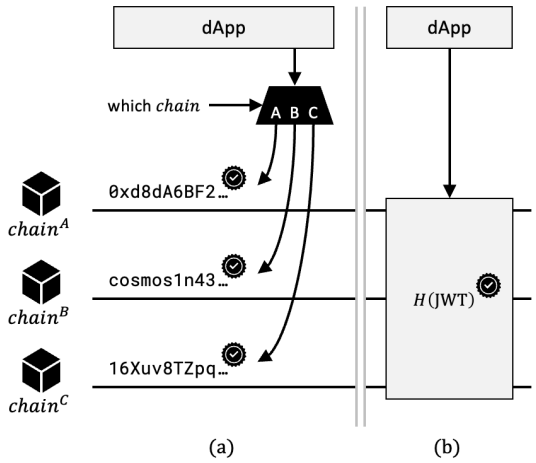

Fig. 4: (a) Traditional multi-chain dApp. (b) Using zkAA identity.

the JWT to the zkAA contract on the blockchain. To ensure that the JWT is valid and belongs to the user, the user would also generate a zk proof using the JWT registration circuit. This proof would demonstrate that the JWT is signed by the social media platform and that the hash of the JWT (H(JWT)) is calculated from the JWT, without revealing the JWT itself. The zkAA contract would verify the proof and, if it is valid, map the H(JWT) identity to the user's contract wallet on the blockchain. This would allow the user to play the web3 game while maintaining their anonymity. The contract wallet could interact with the game contract on the blockchain to perform transactions and execute smart contract functions. When the user wants to make a transaction, they would also include a zk proof generated by the publish circuit that demonstrates that they are the legitimate owner of the H(JWT) identity, without revealing the JWT. The zkAA contract would verify the proof and, if it is valid, execute the original transaction, allowing the user to play the game using their anonymous identity.

In conclusion, zkAA is a promising approach for enabling privacy-preserving authentication and authorization in dApps on the blockchain. The use of zk proofs allows users to prove their identity without revealing their JWT, ensuring privacy and security. The mapping of an H(JWT) identity to a contract wallet on the blockchain also allows users to perform transactions in dApps while maintaining their anonymity. We believe that zkAA has the potential to enable a wide range of new and exciting use cases for dApps on the blockchain.

C. Multi-chain Decentralized Applications

The traditional approach to user identity management in decentralized applications (dApps) across multiple blockchain systems has faced a significant challenge. This challenge arises from the incompatibility of address systems used by different blockchain networks, such as Ethereum, Cosmos, and Libra, among others. These address systems result in difficulties in identifying the same user across multiple dApps. To address this challenge, the Zero-knowledge Address Ab-

straction (zkAA) method offers a promising solution. zkAA employs a non-revealing JSON Web Token (JWT) and its hash to establish a uniform identity that is blockchain-agnostic, thereby overcoming the challenge posed by the disparate address systems. The use of zkAA not only enables the creation of multi-chain applications but also facilitates interoperability between different blockchain networks, enabling the development of decentralized cross-chain applications while maintaining a consistent and secure user identity.

In addition to enabling the seamless creation of multi-chain applications, the implementation of zkAA can also contribute to the optimization of gas fees in the decentralized ecosystem. One approach to reducing gas fee overhead is the adoption of lower-fee blockchains, such as Polygon, as an alternative to Ethereum.

In conclusion, zkAA presents a promising solution for the challenge of cross-chain identity management and gas fee optimization. Its blockchain-agnostic identity and compatibility with multi-chain applications, along with its potential to reduce gas fees, make it a versatile and efficient solution for a range of use cases in the decentralized ecosystem.

VI. Discussion

A. Trusted Setup

Trusted setup is a crucial component in the security and privacy of zkSNARKs-based systems, such as zkAA. It refers to the process of generating and distributing the necessary parameters for the correct operation of the zero-knowledge proof system. In zkSNARKs, the trusted setup is used to ensure that the parameters are generated correctly and cannot be manipulated by a single party for their advantage. To address this issue, a multi-party computation (MPC) protocol can be employed. This protocol allows for the generation of parameters by multiple participants, reducing the risk of manipulation. The MPC protocol works by having hundreds of participants work together to generate the data, and only one of them needs to keep their secret undisclosed for the final output to be secure. The powers-of-tau[] setup is a widely used MPC protocol in modern zkSNARKs.

Recent advancements in zero-knowledge proofs aim to introduce zkSNARKs without a trusted setup, making the system more decentralized and trustworthy. An example of such a system is the Spartan protocol [10].

B. Commit-and-Prove

In this work, we have leveraged the use of zkSNARKs to construct our zero-knowledge argument system, zkAA. However, the field of zero-knowledge proof systems is rapidly evolving and there exists room for improvement in terms of efficiency. One promising direction for future work is the integration of commit-and-prove (CP) schemes into zkSNARKs. CP is a type of SNARK that allows for knowledge to be verified through the use of pre-uploaded commitments. In CP-based SNARKs, the process of verifying hashes, which is typically the most computationally expensive part of generating a proof, is simplified through the use of Pederson commitments. This leads to a decomposition of computations into smaller, specialized proofs, thus potentially increasing the efficiency of proof generation. While the use of CP is expected to significantly increase the efficiency of proof generation, it may result in a slight increase in the cost of verification. This is due to the added complexity of verifying both the Pederson commitment and the original proof. However, it is important to note that the big-O complexity of zkSNARK verification remains constant and is still expected to be O(1).

In conclusion, our future work includes the exploration of CP-based SNARKs and a comprehensive investigation into its potential benefits and trade-offs. We believe that the integration of CP into zkSNARKs holds the potential to greatly improve the efficiency and widespread adoption of zkAA.

C. Privacy and Transaction Fees

In this work, we have considered privacy as a key aspect of the zkAA system. The zkAA approach of detaching identity from the original blockchain address system offers the potential for improved privacy by allowing users to generate fresh ephemeral addresses for each transaction. This approach allows users to maintain their identity while keeping their history of transactions private. However, the creation of a fresh ephemeral address also results in the absence of native assets, making it difficult for users to pay transaction fees. The public link created when sending assets to an ephemeral address may also compromise the privacy of the transaction.

To mitigate these issues, various solutions have been explored, including the study of Stealth addresses[]. One potential solution is the use of account abstraction's paymaster, which is applicable only in the case of ERC20s. Another option to enhance privacy fundamentally is by utilizing zero-knowledge proofs to hide fee-sending transactions. However, this approach increases implementation complexity and overhead. Based on that, we plan to further investigate solutions for preserving privacy in the zkAA system in future work.

D. Oracle Problem in Institute's Public Key

The verification of the validity of a JSON Web Token (JWT) issued by an institute requires the saving of institute's public key in a smart contract. This, however, poses a challenge in terms of trust as it involves an oracle problem in the blockchain. To overcome this issue, the public key can be stored through a governance process involving stakeholders. The other promising solution to address this challenge is to leverage the latest research in oracle problems, such as the Town Crier protocol[].

VII. Related Work

A. Linking Web2 and Web3 Identities

Several approaches have been proposed to link web2 and web3 identities, with the aim of facilitating the transition from web2 to web3. The WebttCom framework [14] relies on a mapping between web2 identities and blockchain-specific identities to achieve this goal. Similarly, Holonym [3] utilizes a mapping approach, using zkSNARKs to prove knowledge

of the JWT preimage and signatures on-chain. Other works such as [8], [11], [13] also focus on linking web2 and web3 identities, but they still rely on blockchain-specific addresses. However, this mapping approach requires careful management of both web2 and web3 identities to ensure their secure use. This not only imposes a responsibility on users to manage their identities securely, but also places a burden on service providers to securely keep and utilize both types of identities.

In contrast, our proposed system, zkAA, eliminates the need for mapping between web2 and web3 identities. The web2 identity, represented as a JWT, is used directly to create a unique proof without the need for a blockchain-specific address. This innovative approach enables the development of multi-chain dApps or web2-web3 hybrid dApps, a capability not currently offered by existing frameworks. Additionally, zkAA enhances privacy by eliminating the risk of data breaches, as the web2 identifier H^k is not publicly linked to any address. Instead, it can be utilized by any address who knows the original JWT, which is not related to the addressing system.

B. Zero-Knowledge Identity Systems

Several studies have leveraged zero-knowledge proofs (ZKPs) to preserve privacy in identity and credential systems. For instance, [1], [8], [9], [13] have utilized ZKPs to maintain the confidentiality of identities or credentials. [5] leverages ZKPs to conceal users' activities and service history from even malicious identity providers. [11] enables face matching for mapping actual individuals to accounts, where ZKPs are used for this mapping.

Holonym [3] and Notebook [6] also utilize ZKPs to bridge the gap between web2 and web3 identities. Holonym verifies the signatures and hashes of JSON Web Tokens (JWTs) using zkSNARKs. Notebook, a zero-knowledge identity infrastructure, verifies credentials signed by third-party organizations as proof of humanity, storing such credentials on blockchain addresses. However, both Holonym and Notebook rely on the original blockchain address as the identifier, rather than using the registered commitment of a web2 identity. In contrast, our proposed system, zkAA, utilizes the registered commitment of a web2 identity as the direct identifier, independent of the original blockchain address.

C. Web3 Login Solutions

In the decentralized application (DApp) arena, Web3 Login Solutions, such as Web3Auth [4] and Ramper [7], have been proposed to enhance the user experience and simplify the onboarding process for both mainstream and crypto-savvy users. These solutions aim to provide seamless authentication through traditional web2 social login mechanisms. Web3Auth employs Shamir Secret Sharing (SSS) and Threshold Cryptography to divide the private key into multiple key shares. In the context of social login, the key generation takes place through a 5/9 consensus mechanism, which ensures that the private keys remain under the control of the user and are not custodial to any party holding a key share. On the other hand, Ramper employs a third-party Key Management System (KMS), a hardware security module (HSM), and encrypted cloud storage. Transactions can be signed within a Trusted Execution Environment (TEE), ensuring that Ramper has no access to the private key and is unable to view it at any time.

However, these approaches still rely on the blockchain-specific identity system of addresses and protect their secrets through SSS or KMS/HSM. In contrast, zkAA fundamentally eliminates the need for storing private keys and instead utilizes the original JSON Web Token (JWT) for social login. As a result, zkAA is inherently non-custodial and does not require the use of specialized hardware such as TEE, or trust in other trusted third parties.

VIII. Conclusion

This paper presents a novel solution to the challenge of privacy-preserving identity management on blockchain platforms. Our contribution, the zkAA protocol, provides a valuable solution for the privacy-conscious users of the blockchain, as it enables them to establish and use their web2 identities on web3 while ensuring full functionality and protection of sensitive information.

The zkAA protocol leverages zero-knowledge proofs and succinct non-interactive arguments of knowledge to provide a secure and efficient framework for identity management. Its four-phase design, which includes Setup, Certificate, Register, and Publish, supports a robust and streamlined process for managing user identities. Additionally, the ability to map H^k with contract wallets or Externally-Owned Accounts (EOAs) expands the potential for decentralized applications, facilitating interoperability between web2 and web3, as well as among different blockchains, with an adjustable balance between the degree of abstraction and cost-effectiveness.

This contribution is deemed significant as it presents a practical and secure solution for privacy-sensitive applications such as social login and web3 games. In contrast to conventional blockchain-based identity management approaches, zkAA does not require the use of private keys or any blockchain-specific features, offering both users and service providers a highly versatile and friendly solution.

Acknowledgment

This work was supported by Coinplug corporations.

References

[1] Jing Chen, Zeyi Zhan, Kun He, Ruiying Du, Donghui Wang, and Fei Liu. Xauth: Efficient privacy-preserving cross-domain authentication. *IEEE Transactions on Dependable and Secure Computing*, 19(5):3301–3311, 2021.

[2] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In Robert Sedgewick, editor, *STOC*, pages 291–304. ACM, 1985.

[3] Nanak Nihal Khalsa, Caleb Tuttle, Lily Hansen-Gillis, Kushal Kahar, and Shady El Damaty. Holonym: A decentralized zero-knowledge smart identity bridge. 2022.

[4] Torus Labs Pte Ltd. Web3auth.

[5] Duc Anh Luong and Jong Hwan Park. Privacy-preserving identity management system on blockchain using zk-snark. *IEEE Access*, 11:1840–1853, 2023.

[6] Nathaniel Masfen-Yan, Solal Afota, Dhruv Mangtani, and Sacha Arroues-Paykin. Notebook: A zero-knowledge identity infrastructure layer.
[7] Inc Ramper. ramper.
[8] Deevashwer Rathee, Guru Vamsi Policharla, Tiancheng Xie, Ryan Cottone, and Dawn Song. Zebra: Anonymous credentials with practical on-chain verification and applications to kyc in defi. *Cryptology ePrint Archive*, 2022.
[9] Michael Rosenberg, Jacob White, Christina Garman, and Ian Miers. zk-creds: Flexible anonymous credentials from zksnarks and existing identity infrastructure. *Cryptology ePrint Archive*, 2022.
[10] Srinath Setty. Spartan: Efficient and general-purpose zksnarks without trusted setup. In *Advances in Cryptology – CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part III*, page 704–737, Berlin, Heidelberg, 2020. Springer-Verlag.
[11] Taotao Wang, Shengli Zhang, and Soung Chang Liew. Linking souls to humans with zkbid: Accountable anonymous blockchain accounts for web 3.0 decentralized identity. *arXiv preprint arXiv:2301.02102*, 2023.
[12] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 2014.
[13] Xiaohui Yang and Wenjie Li. A zero-knowledge-proof-based digital identity management scheme in blockchain. *Computers & Security*, 99:102050, 2020.
[14] Guangsheng Yu, Xu Wang, Qin Wang, Tingting Bi, Yifei Dong, Ren Ping Liu, Nektarios Georgalas, and Andrew Reeves. Towards web3 applications: Easing the access and transition. *arXiv preprint arXiv:2210.05903*, 2022.