GrAC: Graph-Based Anonymous Credentials From Identity Graphs on Blockchain

Wenyi Tang
Computer Science and Engineering
University of Notre Dame
Notre Dame, IN USA
wtang3@nd.edu

Shankha Shubhra Mukherjee
Computer Science and Engineering
University of Notre Dame
Notre Dame, IN USA
smukher4@nd.edu

Seongho Park
Information Systems
Hanyang University
Seoul, South Korea
seonghopark@hanyang.ac.kr

Changhao Chenli
Computer Science
Indiana Institute of Technology
Fort Wayne, IN USA
CChenli@indianatech.edu

Hyunok Oh
Information Systems
Hanyang University
Seoul, South Korea
hoh@hanyang.ac.kr

Jihye Kim
Electrical Engineering
Kookmin University
Seoul, South Korea
jihyek@kookmin.ac.kr

Taeho Jung
Computer Science and Engineering
University of Notre Dame
Notre Dame, IN USA
tjung@nd.edu

2024 IEEE International Conference on Blockchain (Blockchain) | 979-8-3503-5159-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/BLOCKCHAIN62396.2024.00024

***Abstract*—With the growing need for privacy and self-sovereign identity, traditional identity management relying on centralized data registries not only represents single points of failure but also lacks transparency and control over users' identity information. With the built-in tamper-proofness and transparency, blockchain has been widely studied to accommodate the challenges in traditional identity management. Still, it usually comes with privacy concerns due to its public accessibility. Anonymous credentials take advantage of the recent progress in zero-knowledge proof, allowing the unlinkable presentation of only the necessary attributes for a service to guarantee anonymity. However, the existing anonymous credentials require a secondary issuer to verify and manage the anonymized credentials, which compromises the overall transparency and causes indirect management of the user's identity. In this paper, we propose GrAC, a blockchain-based identity management system based on a novel identity graph, which allows users and identity providers to securely store and manage identity information on the blockchain without intermediate entities. GrAC also includes an anonymous authentication protocol suite based on zero-knowledge proof, allowing users to generate one-time anonymous credentials that selectively reveal minimal information to the service provider for authentication. The analysis and evaluations show that GrAC has a reasonable overhead and provides adequate anonymity protection while removing the need for intermediate issuers.**

***Index Terms*—Decentralized Identity Management, Verifiable Credentials, Privacy, Blockchain.**

1. Introduction

The identity management system (IDM) facilitates numerous identification, access control, and authorization applications across different scenarios. It is considered the fundamental component of the physical world and computer-based systems. Even the physical credentials (e.g., passport, driver's license) are being extended or transferred to the digital (e.g., e-passport, digital driver's license) world to perform critical authentication-related functions. Traditionally, the user will request a credential from the institution (known as an identity provider). The credential can be viewed as the combination of serveral attributes that describe the properties of a user, along with the identity provider's endorsement (e.g., digital signature or physical credential with anti-counterfeiting technologies). This process also includes updating the institution's data registry. The users can then present the credentials to the service provider to show that they satisfy certain criteria and get access to the service. The service provider may also retrieve information from the institution's data registry to check if the credential is valid (or has not yet been revoked).

With the increasing need for privacy protection and self-sovereignness in identity information (General Data Protection Regulation from Europe [1], California Consumer Privacy Act (CCPA) [2]), the traditional model presents several challenges that prevent its adoption in current or future internet-based applications. First, multiple identity providers maintain their own data registries for the identity information they manage. This not only leads to multiple single points of failure, but also allows the institutions to update user identity information without further notice, which does not meet the need for transparency and user self-sovereigns. The recent adoption of blockchain and Web3 concepts, which use decentralized ledger technology (DLT), can mitigate such challenges. Utilizing such a public decentralized infrastructure makes the identity data registry auditable to everyone, allowing self-sovereign for users. However, storing sensitive identity information on an open-access system like blockchain introduces serious privacy concerns, which must be addressed.

This work is partially sponsored by NASA ULI under Grant No. 80NSSC23M0058, NSF under Grant No. OAC-2312973, and Notre Dame International under Asia Research Collaboration Grant.

2834-9946/24/$31.00 ©2024 IEEE
DOI 10.1109/Blockchain62396.2024.00024

Second, the presentation of credentials leaks much information in many applications. For example, to access some age-restricted content, the users will be asked to present their credentials with date-of-birth information (e.g., driver's license or passport). Additional information not used in this authentication (e.g., name, appearance description, and address) is also leaked to the service provider. Such information could be linked together if multiple service providers collude, which could lead to privacy compromise for the user (e.g., online activity history). Because some credentials issued by different identity providers may contain redundant information (e.g., driver's license and passport), privacy leakage through linked credentials can be dramatically serious. This situation is also partially because identity providers must maintain their data registries. Recent research in anonymous credential [3]–[6] provides a potential solution by utilizing the zero-knowledge proof, which allows a statement about identity to be verified without leaking the identity information itself. These solutions focus on transforming the issued credentials into an anonymous credential stored on a public bulletin board or blockchain, and the users can then generate an anonymous presentation of the credential with any selected attribute statement about their identity and be verified by the service provider. However, such solutions introduce an extra layer of the intermediate issuer (who verifies the traditional credential, approves, signs, and uploads the anonymous ones to the public domain). The extra layer also makes managing identity information (which should involve and only involve the user and the identity provider) indirect. The Update of identity information will need to go through the intermediate issuer to be reflected on the approved credential list in the public domain. This poses an extra challenge to the freshness guarantees in the protocol design. Some research has been proposed to mitigate the first problem using distributed [6] or threshold cryptography [4]. However, the latter challenge persists commonly in these solutions.

A deeper look at these challenges shows that the intermediate centralized issuer is the major cause of most problems in anonymous credential solutions. The intermediate issuer needs to verify the existing legacy credentials and provide an anonymous/confidential storage form to the blockchain, which eventually serves as a data registry that can be audited to provide transparency and a reference for credential verification. We first noticed that if the public bulletin board or blockchain can serve as a shared data registry for all identity providers, the redundant information issuance can be avoided, and the complexity of identity updates can also be eased. Then, the next question is, can we manage the identity information securely and privately on blockchain to get the advantages of the distributed data registry while supporting a similar level of anonymous credentials?

In this paper, we propose GrAC, a framework for identity management based on blockchain with anonymous credentials. Instead of the credential-based paradigm in anonymous credential research, GrAC uses a novel secure graph structure storage on blockchain to manage the identity data directly. From the identity provider's side, instead of issuing new credentials that may contain redundant information, a transaction indicating the exact new attribute is submitted to the blockchain to represent the new identity attribute issued to the user. From the user's side, the identity information received from different identity providers can be managed in a graph structure on the blockchain. With the combination of symmetric encryption, the graph structure can be securely and anonymously stored on the blockchain while providing different views to users based on the keys they hold. As a result, the user will be able to generate credentials with an arbitrary combination of their identity information with our anonymous authentication protocol suite, which provides complete unlinkability, verifiability, and freshness guarantees. The service providers are only asked to express their attribute requirements and access the blockchain occasionally to get the latest image to verify the credential's freshness.

Our contributions can be summarized as follows:

- We propose GrAC, a decentralized identity information management framework based on blockchain and secure graph data structure, which is the first work in the anonymous credential that achieves direct management and freshness in credential verification.
- We design the anonymous authentication protocol suite based on the graph structure on the blockchain that allows users to generate one-time verifiable credentials with arbitrary attribute combinations, which can also be extended to express complex attribute predicates.
- We provide proof of concept implementation evaluations for the major components of the system. Experiments with our implementation indicate the overall performance

2. Related Work

Blockchain-based IDM With transparency and tamper-proofness, blockchain can address the multifaceted challenges in identity management, focusing on enhancing security measures [7]. One of its most significant contributions is the ability to obviate the necessity for a trusted third party, thereby decentralizing the need for central organizations to manage the identity data registry [8]. The decentralized nature of the blockchain ledger offers another layer of validation, assuring that users, their transactions, and communications are legitimate, making it feasible to build more reliable IDMs. For example, Sovrin [9] utilizes digital credentials through a self-sovereign identity system that operates independently of any centralized authority, leveraging Hyperledger to ensure privacy with pseudonyms and zero-knowledge proof. uPort [10] anchors on the Ethereum [11] to use a smart contract to manage identities and enable secure offline data sharing. ShoCard [12] allows users to control their digital identities on a blockchain platform, eliminating the need for third-party databases by using personal keys for identity verification and storing authentication on the blockchain, which supports the legitimacy of

identities and facilitates external verification. These works typically focus on replacing the infrastructures of traditional public key infrastructure with blockchain backend instead of serving as a universal identity management solution that provides verifiable credentials.

Certain blockchain applications focus on the privacy of user data by using cryptographically protected transactions or smart contracts [13]–[16], but they either can not or have not yet been applied to the realm of identity management adequately.

Anonymous Credential. The attempts to enable anonymous authentication have been studied in traditional scenarios and decentralized scenarios after the emergence of blockchain. Before the age of blockchain, anonymous credential schemes were introduced in the early 2000s, which usually were constructed by combining zero-knowledge proof, group signature schemes, commitment schemes, and protocols for generating and proving knowledge of signatures on committed values and proving commitment equality [17], [18], [20]. They generally rely on the identity provider to issue credentials that can be used in an anonymized way with selective disclosure of attributes. However, they do not consider the transparency issue of identity management.

With the emergence of distributed ledger technology [11], [21], works have presented solutions focusing on anonymity, employing techniques like blind signatures and smart contracts [19]. With the recent development of efficient non-interactive zero-knowledge proof, researchers have started to make use of the NIZK together with a public bulletin board, which can be easily instantiated with blockchain, to build anonymous credentials [3], [4], [6]. Existing credentials are verified and transformed (through an intermediate issuer) into anonymous data structures kept in public, and then the user can use NIZK to generate/re-randomize the credentials for authentication with the service providers.

Comparion with existing work. Table 1 summarizes the difference between existing anonymous credential schemes and GrAC. Existing anonymous credential research (like [3]) usually applies a paradigm of anonymizing existing credentials and generating new ones. They transform the existing credentials (like driver's licenses, passports, etc.) into anonymous credentials and summarize all verified credentials to an approved list, then make it publicly available through a bulletin board or blockchain. This provides compatibility with the legacy credentials but inevitably introduces a somewhat centralized party that must verify and manage the credentials in the approved list. The credentials update also needs to be performed based on the issuance of new credentials.

Instead of issuing verifiable credentials directly to the user for new identities, in GrAC, identity providers generate graph records on the blockchain as proof of identity issuance, which serves as a data registry for the service providers to reference during verification. As a result, the graph contains all identity information for every user, allowing the credential to contain an arbitrary combination of attribute information. It also allows a more efficient identity update process, which only requires the consent of the user and identity provider. The service provider can also apply freshness checks based on the blockchain data.

3. Definitions and Models

We will use $\mathcal{IDO}, \mathcal{IDP}, \mathcal{SP}$ to represent the identity owner (user), identity provider, and service provider in the rest of the paper to describe our design.

3.1. Cryptographic Gadgets

Non-interactive Zero-Knowledge Proof. The general purpose non-interactive zero-knowledge proof (NIZK) allows the prover to convince the verifier that a certain statement is true without revealing any information beyond the statement's validity. A NIZK scheme contains the following algorithms:

- $\mathsf{NIZK.Setup}(pp, desp) \rightarrow \mathsf{crs}$ generates a common reference string (crs) for an arithmetic circuit description and public parameters.
- $\mathsf{NIZK.Prove}(\mathsf{crs}, x, w) \rightarrow \pi$ proves the circuit described by crs is satisfied under public input x and witness w.
- $\mathsf{NIZK.Verify}(\mathsf{crs}, \pi, x) \rightarrow \{0, 1\}$ verifies the proof with the given public input.

A zero-knowledge proof protocol should be completeness, soundness, and zero-knowledge.

Cryptographic Accumulators. The cryptographic accumulators allow a set of elements to be compressed into a short value (the accumulator) and to generate membership proofs that are short and fast to verify. In this paper, we specifically use accumulators that are *dynamic*, meaning that it is possible to publicly compute from an accumulator acc_1 of a set S_1 to the accumulator acc_2 of a set $S_2 = S_1 \uplus S'$ without revealing S_1. In general, the dynamic accumulator contains the following algorithms:

- $\mathsf{ACC.Accum}(pp, S) \rightarrow \mathsf{acc}$ generates accumulator acc for a set S;
- $\mathsf{ACC.PrvMem}(pp, S, x) \rightarrow W_x$ outputs a membership proof showing $x \in S$;
- $\mathsf{ACC.VfyMem}(pp, \mathsf{acc}, x, W_x) \rightarrow 0/1$ accepts or rejects a membership proof;
- $\mathsf{ACC.Inc}(pp, \mathsf{acc}, S') \rightarrow \mathsf{acc}'$ generates accumulator acc' to $S \uplus S'$;

A secure cryptographic accumulator should be correct and collision-free.

With *commit-and-prove* as in [22], a zero-knowledge proof combined with the cryptographic accumulator can express a general structure of the relation $R(S, u) = u \in S \wedge P(u)$, in which the prover needs to convince the verifier that the private information u belongs to a set S and u also satisfies some relation $P(u)$. Such a statement can be proven without leaking any information about u using NIZK.

3.2. System Model

We use a blockchain as a shared infrastructure for all $\mathcal{IDP}$s and $\mathcal{IDO}$s to address the single point of trust issue

TABLE 1. COMPARSION OF EXISTING ANONYMOUS CREDENTIAL WORKS TO OUR WORK.

Protocols	Backend*	Decentralization	Transparency	Direct Update*	Intermediate Issuer-free*
[17], [18]	Unlinkable Signature	×	×	✓	✓
[19]	Unlinkable Signature & Blockchain	✓	✓	×	×
[4]	ZKP & Public Bulletin Board	✓	×	×	×
[3], [6]	ZKP & Public Bulletin Board	✓	✓	×	×
GrAC	ZKP & Blockchain	✓	✓	✓	✓

* The Backend refers to the cryptographic primitives used in the listed work. Direct Update refers to whether the identity update information from the identity provider and the user can be reflected in the validity of verifiable credentials. Intermediate Issuer-free refers to whether an additional entity is needed in the life cycle of anonymous credentials.

in the existing solutions. The blockchain serves as a decentralized ledger with an identity registry for all, providing transparency and tamper-proofness to the identity information. Specifically, GrAC provides the following processes to constitute a complete identity management life cycle.

Identity Update. We abstract the detailed interactions between the $\mathcal{IDP}$ and $\mathcal{IDO}$ and model the identity update as the process for $\mathcal{IDP}$s issuing/revoking new attributes to $\mathcal{IDO}$s. Since GrAC uses the blockchain as an identity registry, the attribute issuance and revocation will result in a new transaction submission. An off-chain interaction may be required between the $\mathcal{IDP}$ and $\mathcal{IDO}$ for certain parameter exchanges to generate the update transaction together and submit it to the blockchain network. In practice, this can be implemented through protocols like multisign transactions in Bitcoin [21], which requires consent from multiple parties to verify the validity of transactions. The process can also be applied anonymously in certain blockchain applications like [23], [24], where the identities of both parties are hidden from the public and the blockchain maintainers.

Credential Generation. The $\mathcal{IDO}$s will generate the anonymous credentials matching a pre-defined attribute requirement from certain $\mathcal{SP}$ using zero-knowledge proof with their secret input and the information stored on the blockchain. The credential generation can be performed by the $\mathcal{IDO}$ alone without any other party.

Credential Verification. In GrAC, we assume the $\mathcal{SP}$ will express an attribute requirement publicly (either on-chain or off-chain) to grant service to whoever presents a valid credential that meets the requirement. When the credential is presented by $\mathcal{IDO}$, $\mathcal{SP}$ will use the auxiliary information queried from the blockchain to verify the credential and provide the following service based on the verification.

GrAC can be deployed on any type of blockchain (permissioned or permissionless) because it is designed at the transactional data level in the blockchain system. Any participant of GrAC can play the role of blockchain peers. Even a third-party service provider like AWS (who is not a user of GrAC) can serve as the blockchain maintainer. Existing service providers like companies or identity providers (e.g. government agencies) should have an incentive to use or maintain the system as they can enjoy the decentralized infrastructure instead of maintaining their own. The consensus protocol of the blockchain can guarantee the integrity of the system if the majority of the maintainers are honest.

3.3. Security Model

In general, the blockchain maintainers as a whole are assumed to be *honest-but-curious* (or *semi-honest*) The integrity, including the ledger integrity and the smart contract execution integrity, can be guaranteed by the blockchain consensus protocol. However, certain/all blockchain maintainers are assumed to be curious about the identity information stored on the blockchain and the individual identity of the $\mathcal{IDO}$s. They will also try to retrieve private information from their store data.

For the identity management application users (include $\mathcal{IDP}, \mathcal{IDO}$ and $\mathcal{SP}$), we make almost the same assumption on the behaviors in their interactions as what is usually assumed in legacy identity management. Specifically, we assume the identity issuance/update information should be trusted when the correlated transactions are submitted and committed to the blockchain. The actual verification and granting process between $\mathcal{IDP}$ and $\mathcal{IDO}$ should be scenario-specific and performed out of the blockchain. With the blockchain-based design, we consider that any operations from $\mathcal{IDP}$ that require writing to the blockchain are transparent and will leave traces on-chain for future audit. Therefore, the $\mathcal{IDP}$s are considered *semi-honest*. We also assume the $\mathcal{SP}$s are *semi-honest*, which means they will honor the verification result of user credentials but try to retrieve sensitive information from them. The $\mathcal{IDO}$s are assumed to be *malicious*, which means they will try to make invalid credentials to $\mathcal{SP}$ to access services they are not supposed to get. Additionally, we assume any two parties in $\mathcal{IDP}, \mathcal{IDO}$ and $\mathcal{SP}$ would not collude with each other to compromise the IDM system. This is a common assumption in most IDM systems because the collusion of any two parties in $\mathcal{IDO}, \mathcal{IDP}$, and $\mathcal{SP}$ can easily compromise the other party regardless of the specific protocol design.

3.4. Design Goals

As described before, GrAC aims to build a decentralized IDM infrastructure with anonymous authentication, which includes several important design goals.

Decentralization and Transparency. The identity information committed to the blockchain should be managed and decentralized, meaning there should be no single point of failure in the system life cycle. The transparency requires any update regarding the identity information should be recorded on-chain, and the process should include the consent from both $\mathcal{IDO}$ and $\mathcal{IDP}$.

Flexible Credential Combination. The concept of Self-Sovereign Identity (SSI) was proposed to allow users to take full control over their identity information, including letting users choose what information to disclose. GrAC allows users to generate credentials with arbitrary attribute combinations from all $\mathcal{IDP}$s in the platform.

Anonymity based on Unlinkability. In each authentication asked by $\mathcal{SP}$, the $\mathcal{IDO}$ should be able to generate a one-time credential based on the identity data registry on the blockchain, which should correspond to the exact attribute needs required by $\mathcal{SP}$. The $\mathcal{IDO}$ anonymity requires 1) the credentials used in different authentications should not be linked (meaning they are computationally indistinguishable) by $\mathcal{SP}$ or any parties except for $\mathcal{IDO}$, even they correspond to the same attribute requirements; 2) the credential should not be linked to any specific piece of information on the blockchain by $\mathcal{SP}$ or any parties except for $\mathcal{IDO}$. The two unlinkabilities prevent the adversary from extracting any private information except for the binary information of credential validity, thus achieving high anonymity for $\mathcal{IDO}$.

Epoch-based Freshness. In IDM, getting verification freshness is a common challenge. In legacy IDM, it is either achieved by assigning the expiration date associated with the credential or checking the data registry for the latest status when verifying the credential. The former is easy to implement but cannot guarantee freshness when revocation is involved. The latter provides a more real-time freshness guarantee but requires the verifier to access the data registry constantly. To balance the freshness and overhead with revocation supported, we use an epoch-based freshness design in GrAC, which requires the service provider to access the blockchain at most once in an epoch for the latest data registry images.

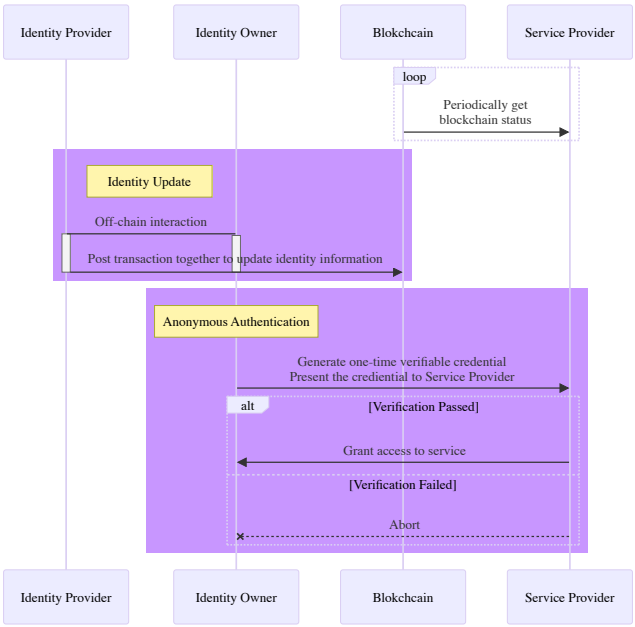

Figure 1. General workflow of GrAC.

4. Our Proposal: GrAC

4.1. System Overview

Our proposal of GrAC contains two major components: the blockchain-based decentralized identity information storage and the anonymous authentication protocol suite. Figure 1 shows the general workflow of GrAC. Specifically, we use a novel blockchain-based decentralized identity information storage structure to store and manage users' identity information. Then, we use zero-knowledge proof to let users generate one-time verifiable credentials that are presented to the service provider for anonymous authentication.

We first represent each user as an identity transaction with a unique identifier containing no identity information. The attribute information (specified by the identity provider who issues the attribute) is kept in plaintext in the attribute information. An extra transaction (called control transaction) will be committed to the blockchain to provide an encrypted computational path from the identity transaction to the attribute transaction, should the attribute be issued to the user. The identity update process (Identity Update in Figure 1) between the user and identity provider is then represented by updating the blockchain ledger data under a two-party protocol, resulting in a new transaction committed to the blockchain and eventually updating the identity. The encrypted graph storage can be built from the linear transaction list on the blockchain to guarantee the user's anonymity and confidentiality of user identity information.

Based on the secure graph storage for identity information, the anonymous authentication protocol suite of GrAC allows users to generate one-time credentials to present the existence of certain attributes required by the service provider (Anonymous Authentication in Figure 1). The credential is the zero-know proof of certain identity information stored in the latest graph status on the blockchain. The service provider can then periodically access the network to get a snapshot of the latest blockchain status as auxiliary information to verify the user credentials, which can be performed on-chain/off-chain (loop in Figure 1).

Notations. In the following section, we use $\mathcal{IDP}_{Attr}$ to represent the identity provider who manages the attribute $Attr$. The blackboard bold characters $\mathbb{BC}, \mathbb{T}, \mathbb{CT}$ represent the blockchain, transaction (as well as vertice node in the graph), and control transaction (as well as the edge node in the graph). For the zero-knowledge proof, we use the notation of $R = (*|Constraints)$ to present the statement that the public input $(*)$ satisfies $Constraints$. The rest of the inputs from the constraints that are not listed as public inputs are considered secret witnesses by default.

4.2. Blockchain-based Decentralized Identity Information Storage

In our setting, we use a blockchain system as a shared/decentralized infrastructure to replace the self-

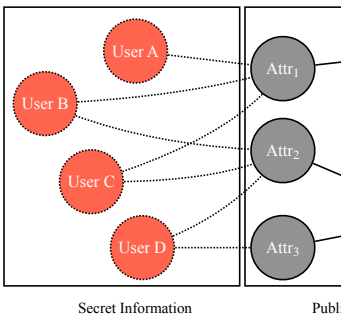

Figure 2. Example of graph representation for u
The vertices for users and the edges between u
remain secret from the public.

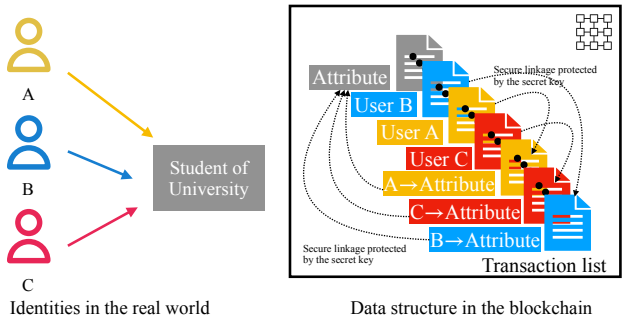

Figure 3. Example of securely storing identity graph data on the blockchain. The dotted arrows indicate the secure linkages from the user transaction to the attribute transaction.

maintained data registries from differ
dates of the blockchain data will be co
only when the blockchain nodes reach
transparency and tamper-proofness to the data stored on-chain.

Graph-like Identity Data Structure. As mentioned, one of our key observations is that some generalized attributes issued by $\mathcal{IDP}$ can be shared by multiple $\mathcal{IDO}$s. For legacy identity management systems, such attribute information is usually recorded on the credentials issued by the $\mathcal{IDP}$, which are conceptually bonded with the $\mathcal{IDO}$. Such information is usually redundant among credentials issued by different $\mathcal{IDP}$s and often leads to unnecessary privacy leakage and management overhead.

We use a different data structure to manage the attribute information. If there is a shared data registry framework for all $\mathcal{IDP}$s, instead of maintaining redundant attribute information for every new credential, an $\mathcal{IDP}$ can append the new attribute to $\mathcal{IDO}$ and let the $\mathcal{IDO}$ to combine the existing attributes he has for specific authentication. This will ease the burden of managing redundant information at the $\mathcal{IDP}$ side and also compress the storage need of identity information among different $\mathcal{IDP}$s. The core idea is to treat the $\mathcal{IDO}$s and attributes from different $\mathcal{IDP}$s as vertices in a graph, and the edges between the $\mathcal{IDO}$ vertex and attribute to represent if an $\mathcal{IDO}$ has the attribute. Attributes of an $\mathcal{IDP}$ can be shared by (connected) by multiple $\mathcal{IDO}$ vertices, which provides a management view for the $\mathcal{IDP}$. An $\mathcal{IDO}$ vertex can also connect to multiple attribute vertices, which represent all attributes the $\mathcal{IDO}$ has been assigned. As shown in Figure 2, the attributes and $\mathcal{IDP}$s can be viewed as public information since the information of the $\mathcal{IDP}$s and what attributes can be issued by them are considered public information. The vertice for the users and the linkages between the users and attributes should be kept secret from everyone except for the connected user and $\mathcal{IDP}$.

Secure Identity Graph Storage on Blockchain. The graph structure provides a flexible and compact management view for the identity information To make the graph-like data structure securely applicable on a blockchain, GrAC applies a similar idea of securely storing the provenance graph from [25] to store the identity graph, as shown in Figure 3.

Specifically, each $\mathcal{IDO}$ owns a transaction $\mathbb{T}_{\mathcal{IDO}}$ on-chain and a secret key $sk_{\mathcal{IDO}}$. The $\mathbb{T}_{\mathcal{IDO}}$ contains the hash of the $sk_{\mathcal{IDO}}$, which serves as a commitment of the secret key on-chain. Each $\mathcal{IDP}$ also owns a transaction $\mathbb{T}^{\mathcal{IDP}}_{Attr}$ on-chain and a secret key $sk^{\mathcal{IDP}}_{Attr}$ for each attribute.

Conceptually, the basic form of a credential showing an $\mathcal{IDO}$ has an attribute $Attr$ can be represented as a graph path: $\mathbb{T}_{\mathcal{IDO}} \rightarrow \mathbb{T}^{\mathcal{IDP}}_{Attr}$. Therefore, the process for $\mathcal{IDP}$ to issue a new attribute to the $\mathcal{IDO}$ can be done by collaboratively submitting a new transaction $\mathbb{CT}$ (control transaction), which represents the corresponding edge. In this model, such process involves a two-party interactive protocol between the $\mathcal{IDO}$ (the owner of $\mathbb{T}_{\mathcal{IDO}}$) and the $\mathcal{IDP}$ (the owner of the $\mathbb{T}^{\mathcal{IDP}}_{Attr}$), which echos to the identity issuance process between $\mathcal{IDO}$ and $\mathcal{IDP}$. The control transaction can be simplified as a structure of $\mathbb{CT} = (P, Next)$, in which P is a pseudo-random string that can be used to locate/identify a $\mathbb{CT}$ and $Next$ is a ciphertext encrypted with $sk_{\mathcal{IDO}}$, which includes the information related to $\mathbb{T}^{\mathcal{IDP}}_{Attr}$. Algorithm 1 shows the simplified process of $\mathbb{CT}$, which makes $\mathbb{T}^{\mathcal{IDP}}_{Attr}$ computationally reachable from the given sk_{IDO}. In practice, the process includes multiple interactions between $\mathcal{IDO}$ and $\mathcal{IDP}$, and $sk_{\mathcal{IDO}}$ would be kept secret from $\mathcal{IDP}$ during the collaboration. Refer [25] for a detailed description of the protocol.

After the corresponding $\mathbb{CT}$ is committed, the IDO can use an algorithm to go through the transaction path from the secret $sk_{\mathcal{IDO}}$ to the target $\mathbb{T}^{\mathcal{IDP}}_{Attr}$. Algorithm 2 shows the algorithm $\mathsf{Reach}()$ for the IDO to reach $\mathbb{T}^{\mathcal{IDP}}_{Attr}$ through the committed control transaction. Specifically, $\mathsf{Reach}()$ contains 2 computations H and Enc, and a database Query on blockchain. The idx used in both algorithms can be randomly chosen by $\mathcal{IDO}$ or referenced from public randomness (usually from blockchain). It provides a simple reference (or a tag-like feature) for $\mathcal{IDO}$ to reach multiple $Attr$s through $\mathsf{Reach}()$.

4.3. Anonymous Authentication Protocol Suite

We can now build an anonymous authentication protocol suite with the aforementioned identity data structure. In a

Algorithm 1 Generate $\mathbb{CT}$ (2-party algorithm*)

$\mathbb{T}_{\mathcal{IDO}} = \mathsf{H}(sk_{\mathcal{IDO}})$
$\mathbb{T}^{\mathcal{IDP}}_{Attr} = (AttID, \mathsf{Info})$
function GEN$\mathbb{CT}$($sk_{\mathcal{IDO}}, \mathbb{T}_{\mathcal{IDO}}, \mathbb{T}^{\mathcal{IDP}}_{Attr}$)
    $\mathcal{IDO}$ chooses an idx
    $P = \mathsf{H}(sk_{\mathcal{IDO}}, idx)$
    $Next = \mathsf{Enc}(sk_{\mathcal{IDO}}, AttID)$
    if $\mathcal{IDO}$ and $\mathcal{IDP}$ agree on $(P, Next)$ **then**
        Collaboratively generate $\mathbb{CT} = (P, Next)$
        Submit $\mathbb{CT}$ to Blockchain
    else
        Abort
    end if
end function

* We assume the interaction in the 2-party protocol will guarantee the correctness of its output control transaction.

Algorithm 2 Reach the attribute transaction $\mathbb{T}^{\mathcal{IDP}}_{Attr}$

function REACH($sk_{\mathcal{IDO}}, idx, \mathbb{BC}, \mathbb{T}^{\mathcal{IDP}}_{Attr}$)
    $P' = \mathsf{H}(sk_{\mathcal{IDO}}, idx)$
    $\mathbb{CT}' = \mathsf{Query}(P', \mathbb{BC})$
    if $\mathbb{CT}' = \perp$ **then**
        return False
    end if
    $AttID' = \mathsf{Dec}(sk_{\mathcal{IDO}}, \mathbb{CT}'.Next)$
    if $AttID' = \mathbb{T}^{\mathcal{IDP}}_{Attr}.AttID$ **then**
        return True
    end if
    return False
end function

Figure 4. The algorithms for generating $\mathbb{CT}$ and reach $\mathbb{T}^{\mathcal{IDP}}_{Attr}$. H refers to a secure hash function, $\mathsf{Enc}/\mathsf{Dec}$ refers to a symmetric encryption/decryption. $\mathsf{Query}(P', \mathbb{BC})$ means query transaction from the blockchain with key P'.

nutshell, we transform the execution of $\mathsf{Reach}()$ into a zero-know proof that anyone with access to the blockchain can publicly verify.

Credential Generation/Verification We start from a simple example where the user $\mathcal{IDO}$ needs to $\mathcal{SP}$ that he holds an attribute $Attr$. In the graph-like identity information storage, such connection can be expressed as a path in the identity graph in the form of $\mathbb{T}_{\mathsf{IDO}} \rightarrow \mathbb{CT} \rightarrow \mathbb{T}^{\mathcal{IDP}}_{Attr}$, which can be implied by proving $\mathsf{Reach}() = 1$. To generate verifiable anonymous credentials from the process, we use a combination of accumulator-based membership proof and general-purpose zero-knowledge proof. Specifically, the algorithm $\mathsf{Reach}()$ can be expressed as the correct computation of sub-processes H and Enc, along with the existence query of $\mathbb{CT} \in BC$. H and Enc can be expressed in ZKP-friendly arithmetic circuits. For $\mathbb{CT} \in BC$, we convert the query into a membership proof from a cryptographic accumulator $\mathsf{acc}_{\mathbb{CT}}$. The $\mathsf{acc}_{\mathbb{CT}}$ summarizes each $\mathbb{CT}$ into a digest (defined as $\mathsf{H}(\mathbb{CT}.P + \mathbb{CT}.Next)$, in which $+$ denotes concatenation and H denotes hash), and accumulates the digest of all $\mathbb{CT}$s in the blockchain. The $\mathsf{acc}_{\mathbb{CT}}$ can be calculated and updated by simply accumulating every new $\mathbb{CT}$, which can then be maintained by a smart contract $\mathcal{SC}_{\mathsf{acc}}$. We assume $\mathcal{SC}_{\mathsf{acc}}$ is called periodically to accumulate the control transactions in a given epoch.

In summary, the following relation is proven through ZKP as the credential for an attribute:

$$\left((\mathsf{acc}_{\mathbb{CT}}, \mathbb{T}^{\mathcal{IDP}}_{Attr}) \;\middle|\; \begin{array}{l} \mathsf{H}(sk, idx) = \mathbb{CT}.P; \\ \mathsf{Enc}(sk, \mathbb{T}^{\mathcal{IDP}}_{Attr}) = \mathbb{CT}.Next; \\ \mathsf{H}(\mathbb{CT}) \in \mathsf{acc}_{\mathbb{CT}} \end{array} \right)$$

, with $sk, idx, \mathbb{CT}$ and witness of membership for $\mathsf{acc}_{\mathbb{CT}}$ as secret input (witnesses).

To verify the presented credential, the $\mathcal{SP}$ is expected to periodically access the blockchain to get the latest image of $\mathsf{acc}_{\mathbb{CT}}$ from $\mathcal{SC}_{\mathsf{acc}}$. Then, a credential can be simply verified by using $\mathsf{NIZK.Verify}()$.

Revocation The revocation of attributes has been one of the most common challenges in any identity management system. In GrAC, achieving revocation is also not trivial because of the tamper-proofness of the blockchain. Deleting or modifying existing data on the blockchain is either impossible or requires a tremendous structural change to make it redactable [26].

In GrAC, we use the graph structure storage and non-membership proof of the cryptographic accumulator to achieve efficient attribute revocation. Specifically, if an attribute (represented by $\mathbb{T}^{\mathcal{IDP}}_{Attr}$) is revocable, the $\mathcal{IDP}$ and $\mathcal{IDO}$ can pre-define a revocation control transaction $\overline{\mathbb{CT}}_{\mathcal{IDO}-Attr}$, which should be submitted to the blockchain when attribute $Attr$ is revoked from $\mathcal{IDO}$. The $\overline{\mathbb{CT}}_{\mathcal{IDO}-Attr}$ should be computationally indistinguishable from $\mathbb{CT}_{\mathcal{IDO}-Atttr}$ or other control transactions in order to get unlinkability. we extend the structure of control transaction $\mathbb{CT} = (P, Next, r)$ To enable the revocation, we extend the structure of the control transaction by attaching a random seed r to it. The revocation control transaction can now be defined as the identical structure $\overline{\mathbb{CT}} = (P, Next)$, in which $\overline{\mathbb{CT}}.P = \mathsf{H}(sk_{\mathcal{IDO}}, (idx + r))$, $\overline{\mathbb{CT}}.Next = \mathsf{H}(sk_{\mathcal{IDO}}, (AttID + r))$

When the $\mathcal{IDO}$ wants to generate a credential to prove an effective attribute $Attr$ (which is not yet revoked) to a $\mathcal{SP}$, the credential should include the statement of $\mathbb{CT}_{\mathcal{IDO}-Attr} \in \mathsf{BC} \wedge \overline{\mathbb{CT}}_{\mathcal{IDO}-Attr} \notin \mathsf{BC}$. With such construction, $\mathcal{SP}$ can add the latter non-membership statement in the requirement to ensure the $\mathcal{IDO}$ holds a valid attribute (that has not been revoked). The relation of an effective attribute credential is now defined as:

$$\left((\mathsf{acc}_{\mathbb{CT}}, \mathbb{T}^{\mathcal{IDP}}_{Attr}) \;\middle|\; \begin{array}{l} \mathsf{H}(sk, idx) = \mathbb{CT}.P; \\ \mathsf{Enc}(sk, \mathbb{T}^{\mathcal{IDP}}_{Attr}) = \mathbb{CT}.Next; \\ \mathsf{H}(sk, (idx + r)) = \overline{\mathbb{CT}}.P; \\ \mathsf{Enc}(sk, \mathbb{T}^{\mathcal{IDP}}_{Attr}) - r = \overline{\mathbb{CT}}.Next; \\ \mathsf{H}(\mathbb{CT}) \in \mathsf{acc}_{\mathbb{CT}} \\ \mathsf{H}(\overline{\mathbb{CT}}) \notin \mathsf{acc}_{\mathbb{CT}} \end{array} \right)$$

, with $sk, idx, \mathbb{CT}, r, \overline{\mathbb{CT}}$ and membership/non-membership witness of $\mathsf{acc}_{\mathbb{CT}}$ as secret inputs (witnesses).

Rich Attribute Predicates Representation. The graph path-based representation of credential in GrAC provides the ability to demonstrate if the $\mathcal{IDO}$ holds the $Attr$ (or not). Multiple simple statements can be combined to express complex binary identity requirements from the $\mathcal{SP}$. Because every encrypted path and the corresponding anonymous credential is unforgeable (presented in Section 5), multiple credentials presented simultaneously can naturally be interpreted as the logic of and.

In addition to combining multiple binary representations of attributes, it is also possible to build a credential that represents or relationships among multiple attributes with the graph representation and the accumulator. Since the or relation is another form of anonymity in a given set (enumeration of all attributes that satisfy the requirement), we can use another accumulator that includes all $\mathbb{T}_{Attr}^{\mathcal{IDP}}$s that satisfy the given requirement. Then, we can include another accumulator into the credential relationship to provide another level of anonymity among the specific attributes to represent the or relation without revealing the specific attribute.

5. Security Analysis

We analyze the security of GrAC from the 2 aspects of the paper: the blockchain-based identity information storage and the anonymous authentication protocol. Due to the page limit, we provide a descriptive security analysis.

We claim the identity information storage mechanism in GrAC satisfies *confidentiality* and *correctness*. The anonymous authentication protocol in GrAC satisfies *unforgeability*, *$\mathcal{IDO}$ unlinkability*, *multi-show unlinkability*, and eventually *anonymity*. These claims should stand if the underlying cryptographic primitives (symmetric encryption, collision-resistant hash, and zero-knowledge proof) hold their corresponding properties.

Theorem 1. ***Confidentiality.*** *The identity information stored on the blockchain will be kept secret against any entity without the proper secret key corresponding to the $\mathbb{T}_{\mathcal{IDO}}$. Specifically, the identity information for an $\mathcal{IDO}$ should be indistinguishable from other $\mathcal{IDO}$s in GrAC.*

The confidentiality can be implied by the *linkage privacy* stated in [25]. Specifically, the linkage privacy of the graph storage guarantees that entities without corresponding secret keys can not learn any information about the graph or relationship between individual vertices, except for the information from the graph records and temporal relation (an edge can only connect two existing vertices). In GrAC, the vertices ($\mathbb{T}_{\mathcal{IDO}}$s) contain only hashes of secret key, and the edges ($\mathbb{CT}$s) only contain the salted hash and symmetric ciphertext, which are pseudo-random elements that contain no information about the user identity. Therefore, the identity information storage in GrAC guarantees confidentiality.

Theorem 2. ***Correctness*** *The anonymous credential generation/verification in GrAC will succeed if the input is valid graph paths stored on the blockchain and matches the attribute needs.*

The correctness of the anonymous authentication protocol in GrAC can be guaranteed if a valid witness is provided and the zero-knowledge proof satisfies completeness.

Theorem 3. ***Unforgeability.*** *An $\mathcal{IDO}$ should have a negligible probability of constructing valid anonymous credentials without holding the secret key of a legit $\mathcal{IDO}$ with the target attribute. Namely, an $\mathcal{IDO}$ cannot feasibly generate a credential and pass the verification without valid records in the graph on the blockchain.*

The unforgeability is guaranteed through the security of both symmetric encryption and the hash function used in the graph storage protocol and the property of the cryptographic accumulator. Specifically, for an outside attacker who is not an $\mathcal{IDO}$ (holds no key of any $\mathcal{IDO}$) in the system, it is hard to generate a valid witness of a credential, which implies the computation of a preimage from its hash and symmetric decryption without the corresponding secret key.

Theorem 4. ***$\mathcal{IDO}$ Unlinkability.*** *A credential generated/presented by an $\mathcal{IDO}$ should not reveal any information about the $\mathcal{IDO}$, except for attributes information disclosed by the credential (matches the need from $\mathcal{SP}$).*

As long as the ZKP protocol used to generate the credential holds the zero-knowledge property, the credential should not reveal any information about the witness of the statement within, which includes the computation for the graph path and the records in the graph. The public input of the proof only includes the attribute required by $\mathcal{SP}$ and a cryptographic accumulator $\mathsf{acc}_{\mathbb{CT}}$. The accumulator is a public summary of the blockchain data ($\mathbb{CT}$s to be specific), while the witness of the membership (or non-membership) is hidden with the ZKP. Therefore, the credential can not be associated with specific values/graph records linked to a single $\mathcal{IDO}$.

Theorem 5. ***Multi-show Unlinkability.*** *Multiple credentials generated/presented by an $\mathcal{IDO}$ should not reveal any information about the $\mathcal{IDO}$, except for attributes information disclosed by the credentials (matches the need from $\mathcal{SP}$).*

This theorem can be implied by Theorem 4, as every credential presented by $\mathcal{IDO}$ is zero-knowledge, which does not link to any specific record of the graph on the blockchain (except for the $\mathbb{T}_{Attr}^{\mathcal{IDP}}$s). Therefore, the $\mathcal{SP}$ (or other parties other than $\mathcal{IDO}$) cannot link different credentials to one $\mathcal{IDO}$ or any specific graph record other than the attribute information.

Theorem 6. ***$\mathcal{IDO}$ Anonymity.*** *If the $\mathcal{IDO}$ and $\mathcal{IDP}$s are honest, $\mathcal{SP}$s can not learn anything about $\mathcal{IDO}$ during the lifecycle of GrAC.*

The anonymity is composed of the anonymity from the identity information storage and the anonymity of the authentication. For the identity graph storage in GrAC, even though the records for attributes are publicly identifiable and the records for each $\mathcal{IDO}$ are unique among each other, the control transactions that connect $\mathcal{IDO}$ and attribute are pseudo-random parameters generated from symmetric en-

cryption and secure hash, which should be indistinguishable from randomness. The $\mathcal{SP}$s or any other entities without the secret key cannot learn the connection between the $\mathcal{IDO}$ records and attribute records from the information stored on the blockchain. As stated before, GrAC also guarantees that the multiple credentials shown by the $\mathcal{IDO}$ could not be linked and pointed to any $\mathcal{IDO}$ or specific record in the graph.

6. Evaluation

We instantiate GrAC with a concrete setting to evaluate the performance of our design. Specifically, we split the evaluation into two parts: the on-chain part and the off-chain part. For the on-chain part, we use Hyperledger fabric [27] to implement a smart contract, which includes creating and updating an RSA accumulator. For the off-chain part, we use Groth16 [28] with Mimc ciphertext [29] and RSA accumulator [22] to implement the zero-knowledge proof-based anonymous authentication. We also provide our source code for reproductivity at https://github.com/IyNew/Grac.

6.1. On-Chain Evaluation

The on-chain evaluation mainly contains the smart contract execution to maintain the accumulator of control transactions. Specifically, a smart contract $\mathcal{SC}_{\mathsf{acc}}$ is deployed and called periodically to accumulate the control transactions into an RSA accumulator in a given epoch. In the evaluation, we assume each epoch will contain a fixed number of control transactions and that the setup parameters for the RSA accumulator are in place and ready to use. Figure 5 shows the time consumption of the smart contract to accumulate different numbers of control transactions in each epoch from the blockchain. The time consumption for the computation of the accumulator has linear complexity (in terms of the number of exponentiation computations) to the number of control transactions to accumulate, which could be a big overhead when more control transactions are being accumulated. One possible solution to mitigate this is to use an incremental accumulator, which allows the use of an old accumulator to compute new accumulators with new elements incrementally. According to our evaluation in Figure 5, the incremental accumulator saves around 50% to 66% of computation time when computing the same accumulator size. We claim that GrAC utilizes a black-box design, so the system's overall efficiency can be improved by applying more efficient cryptographic primitives that support membership/non-membership proof.

6.2. Off-chain Evaluation

The off-chain evaluation contains the life cycle for the anonymous credentials. From the $\mathcal{IDO}$'s side, it mainly includes the credential generation, which is the NIZK proof generation for the encrypted path to an attribute transaction as the $\mathcal{SP}$ requested. The $\mathcal{SP}$ needs to verify the credential,

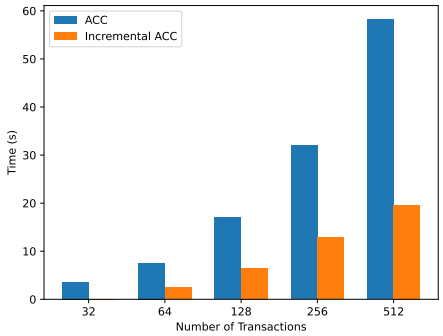

Figure 5. Time consumption for maintaining RSA accumulator on the blockchain. Blue bars show the time consumption for accumulating the corresponding number of control transactions on the x-axis. Orange bars show the time consumption for incrementally computing the accumulator from previous accumulator values.

TABLE 2. CREDENTIAL-RELATED EVALUATION FOR GRAC

		Prove Mem(s)	Prove Path(s)	Verify Mem(s)	Verify Path(s)
Single Attribute		0.436	1.36	0.035	0.096
Revocation	Existence	0.417	1.40	0.032	0.103
	Non-exist	0.385	1.42	0.030	0.114

namely the verification of the NIZK proof (the anonymous credential).

Witness Management. To generate an anonymous credential, the user needs to provide the corresponding witness to the circuits of the zero-knowledge proof. Specifically, the witness includes the secret key sk and the index idx for the control transaction that links to the required attribute, which is easy to maintain. It also needs the witness of the RSA accumulator, which is $\mathsf{acc}_{\mathbb{CT}}/\mathbb{CT}$ (the same computation of the accumulator without the target $\mathbb{CT}$). This computation is similar to the computation of the accumulator computation as shown in Figure 5.

Credential Generation/Verification. Table 2 shows the performance of the credential metric for a single attribute requirement and a single attribute with non-revoked status. We use a Pedersen commitment to connect the Groth16 proofs of the Mimc encryption and the membership proof from the accumulator as in [22]. For the accumulator part in the credential with/without non-revoked status, it is very efficient for the user to generate the membership proof as it only includes one power operation with the membership witness in the RSA accumulator. The verification phase only contains the retrieval of the accumulator from the blockchain and the verification of the zero-knowledge proof. Since we are using ZK-SNARK, which has constant time complexity in verification, the credential verification for the service provider can be as efficient as in the ms level.

7. Conclusion

In this paper, we propose GrAC, a blockchain-based framework that empowers anonymous credentials with se-

cure identity graph storage. GrAC utilizes a secure graph structure to store the identity information, which can be stored securely on the blockchain and provide a decentralized and transparent identity registration for the users, identity providers, and service providers. The authentication protocol suite in GrAC utilizes general zero-knowledge proof and blockchain-maintained cryptographic accumulator to realize anonymous credentials. With the blockchain-based identity graph as a data registry, GrAC achieves secure, transparent, and direct identity management for users while supporting anonymous credentials with freshness and effectiveness. Our evaluations show that GrAC has a reasonable overhead, and such a framework has the potential to be utilized in more application scenarios.

References

[1] "General Data Protection Regulation (GDPR) – Official Legal Text — gdpr-info.eu." https://gdpr-info.eu/. [Accessed 08-04-2024].

[2] "California Consumer Privacy Act (CCPA)." https://oag.ca.gov/privacy/ccpa, Oct. 2018.

[3] M. Rosenberg, J. White, C. Garman, and I. Miers, "zk-creds: Flexible anonymous credentials from zksnarks and existing identity infrastructure," in *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 790–808, IEEE, 2023.

[4] C. Garman, M. Green, and I. Miers, "Decentralized anonymous credentials," *Cryptology ePrint Archive*, 2013.

[5] P. Pauwels, "zkkyc: A solution concept for kyc without knowing your customer, leveraging self-sovereign identity and zero-knowledge proofs," *Cryptology ePrint Archive*, 2021.

[6] A. Sonnino, M. Al-Bassam, S. Bano, S. Meiklejohn, and G. Danezis, "Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers," *arXiv preprint arXiv:1802.07344*, 2018.

[7] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology?—a systematic review," *PloS one*, vol. 11, no. 10, p. e0163477, 2016.

[8] S. Raju, S. Boddepalli, S. Gampa, Q. Yan, and J. S. Deogun, "Identity management using blockchain for cognitive cellular networks," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2017.

[9] A. Tobin and D. Reed, "The inevitable rise of self-sovereign identity," *The Sovrin Foundation*, vol. 29, no. 2016, p. 18, 2016.

[10] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena, "Uport: A platform for self-sovereign identity," *URL: https://whitepaper. uport. me/uPort_ whitepaper_DRAFT20170221. pdf*, vol. 128, p. 214, 2017.

[11] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[12] P. Dunphy and F. A. Petitcolas, "A first look at identity management schemes on the blockchain," *IEEE security & privacy*, vol. 16, no. 4, pp. 20–29, 2018.

[13] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*, pp. 459–474, IEEE, 2014.

[14] T. H. Yuen, S.-f. Sun, J. K. Liu, M. H. Au, M. F. Esgin, Q. Zhang, and D. Gu, "Ringct 3.0 for blockchain confidential transaction: Shorter size and stronger security," in *International Conference on Financial Cryptography and Data Security*, pp. 464–483, Springer, 2020.

[15] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE symposium on security and privacy (SP)*, pp. 839–858, IEEE, 2016.

[16] B. Bünz, S. Agrawal, M. Zamani, and D. Boneh, "Zether: Towards privacy in a smart contract world," in *Financial Cryptography and Data Security: 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10–14, 2020 Revised Selected Papers*, pp. 423–443, Springer, 2020.

[17] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Annual international cryptology conference*, pp. 56–72, Springer, 2004.

[18] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham, "Randomizable proofs and delegatable anonymous credentials," in *Advances in Cryptology-CRYPTO 2009: 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pp. 108–125, Springer, 2009.

[19] R. Mukta, J. Martens, H.-y. Paik, Q. Lu, and S. S. Kanhere, "Blockchain-based verifiable credential sharing with selective disclosure," in *TrustCom*, pp. 959–966, IEEE, 2020.

[20] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *International conference on the theory and applications of cryptographic techniques*, pp. 93–118, Springer, 2001.

[21] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[22] D. Benarroch, M. Campanelli, D. Fiore, K. Gurkan, and D. Kolonelos, "Zero-knowledge proofs for set membership: Efficient, succinct, modular," in *International Conference on Financial Cryptography and Data Security*, pp. 393–414, Springer, 2021.

[23] D. Hopwood, S. Bowe, T. Hornby, and N. Wilcox, "Zcash protocol specification," *GitHub: San Francisco, CA, USA*, vol. 4, p. 220, 2016.

[24] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *2013 IEEE Symposium on Security and Privacy*, pp. 397–411, IEEE, 2013.

[25] W. Tang, C. Chenli, C. Ju, and T. Jung, "Trac2chain: trackability and traceability of graph data in blockchain with linkage privacy," in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pp. 272–281, 2022.

[26] G. Ateniese, B. Magri, D. Venturi, and E. Andrade, "Redactable blockchain–or–rewriting history in bitcoin and friends," in *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 111–126, IEEE, 2017.

[27] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, pp. 1–15, 2018.

[28] J. Groth, "On the size of pairing-based non-interactive arguments," in *Annual international conference on the theory and applications of cryptographic techniques*, pp. 305–326, Springer, 2016.

[29] M. Albrecht, L. Grassi, C. Rechberger, A. Roy, and T. Tiessen, "Mimc: Efficient encryption and cryptographic hashing with minimal multiplicative complexity," in *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, pp. 191–219, Springer, 2016.