Decentralized and Privacy-Preserving Digital Identities with Verifiable Credentials and Revocable Anonymity

Joan Ferré-Queralt[1][0009−0002−0014−0065], Jordi Castellà-Roca[1][0000−0002−0037−9888], and Alexandre Viejo[1][0000−0003−2342−5100]

Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
UNESCO Chair in Data Privacy,
CYBERCAT-Center for Cybersecurity Research of Catalonia
Av. Països Catalans 26, E-43007 Tarragona, Catalonia, Spain
{joan.ferreq,jordi.castella,alexandre.viejo}@urv.cat

Abstract. Traditional centralized identity management systems present security and privacy vulnerabilities due to the centralization of sensitive personal information. Decentralized identity management systems (DIMS) leveraging blockchain technology offer the potential to address these challenges. This work introduces a novel approach within DIMS and attribute management, reducing trust in centralized actors while ensuring the validity of entities' attributes through blind signatures and Zero-Knowledge Proofs. It also implements revocable anonymity for misbehaving entities using a secret sharing scheme, thereby enhancing the security, privacy, and decentralization of identity solutions.

Keywords: Digital identities · Verifiable credentials · Privacy · Blockchain · Revocable anonymity

1 Introduction

The concept of identity permeates our daily existence, evolving continuously as we journey through life. During this process, we accumulate a rich tapestry of experiences, qualifications, and affiliations that shape our essence. These defining attributes seamlessly integrate with our sense of self, often requiring tangible validations such as a driver's license, passport, or professional accreditation. These credentials not only authorize our actions but also serve as affirmations of our unique traits.

The rise of the Internet[1] and its pervasive integration into almost every aspect of modern society have driven a transformation in the management of our identities. A growing proportion of our engagements – from financial transactions and professional engagements to social connections and civic participation – now

[1] Rohit Shewale: Internet User Statistics In 2024 — (Global Demographics), https://www.demandsage.com/internet-user-statistics

unfold within the internet, compelling users to validate diverse aspects of their identity electronically. For instance, individuals may be required to authenticate their age to access restricted content, validate their residency for governmental services, or confirm their qualifications for participation in professional networks.

The initial online identity management systems relied on public key infrastructures [7]. However, their dependence solely on certificates proved inadequate for widespread adoption due to challenges with interoperability and cost. Consequently, they primarily found utility in server authentication.

For the user authentication, most services compelled users to provide a plethora of personal information and generate unique usernames and passwords for each platform, all in pursuit of robust access security. Yet, this method often demands the disclosure of excessive personal information, even for routine authentication tasks like age verification. Moreover, this information is typically stored within centralized databases, safeguarded solely by passwords. These centralized repositories of sensitive data have proven themselves tempting targets for malicious actors, increasing the risk of data breaches and identity theft. The misuse of Facebook user data in the Cambridge Analytica scandal [8] vividly underscores this important concern.

As a consequence, there is a growing interest in delving into decentralized alternatives for *digital identities (DIDs)* and *verifiable credentials (VC)* supported by decentralized technologies such as the blockchain [9]. In this context, DIDs serve as unique identifiers for individuals, organizations, or even objects. VCs, tethered to DIDs, function as digitally signed claims containing attributes [11] issued by trusted authorities. Unlike conventional digital certificates, VCs empower users to selectively disclose only pertinent information in a given context, thereby enhancing both privacy and user autonomy.

Decentralized alternatives aim to address the deficiencies of classic centralized systems by crafting more secure, privacy-preserving solutions for managing digital identities and authentication attributes. However, the anticipated advantages are accompanied by relevant challenges that warrant consideration.

A primary concern revolves around ensuring the dependable linkage of diverse attributes and credentials to a singular user, thereby dealing with the delicate balance between addressing the absence of a universal identifier (e.g., email or social security number) and mitigating apprehensions regarding excessive traceability across various services.

A second significant challenge lies in establishing trust and governance. Within a decentralized model, determining who is authorized to issue trustworthy credentials and implementing measures to prevent an inundation of low-quality credentials is imperative for maintaining the system's integrity.

Finally, within a decentralized identity system, individuals wield greater authority over the dissemination and accessibility of their personal data. However, there may arise situations where it is necessary to revoke anonymity in order to address issues such as fraud, abuse, or illegal activity. In such decentralized environments, identifying and holding individuals accountable for their actions becomes more challenging without infringing on their privacy and autonomy.

Therefore, finding ways to balance the need for revocable anonymity with the principles of decentralization and privacy is an important consideration in the design and implementation of a decentralized identity system.

1.1 Related Work

In response to the limitations posed by centralized approaches, researchers shifted their focus toward decentralized models for Digital Identity Management Systems (DIMS). In doing so, they discovered that the inherently decentralized blockchain technology was the ideal solution to support their designs.

First schemes along this path included initiatives like Hyperledger Aries and Indy, both of which originated from the Sovrin Foundation [12], or the MIT project ChainAnchor [6]. These schemes operate on *permissioned blockchains*, a specific type of blockchain network where access to participate in the network and perform certain actions is restricted to a predefined set of entities or participants. In particular, these proposals require specialized and trusted nodes (referred to as "Stewards" in [12] and "Validators" in [6]) to execute the consensus algorithm. While utilizing restricted blockchains for global identity management may appear impractical, the inclusion of these specialized consensus nodes poses the risk of ledger manipulation in instances of corruption beyond a certain threshold.

In contrast to permissioned blockchains, *public blockchains* allow for unrestricted participation, enabling anyone to join the network and engage in the consensus process and, in turn, enhancing trust decentralization and user control. The scientific community has leveraged the advantages of public blockchains to introduce innovative decentralized identity (DID) systems that effectively tackle the challenges posed by previous solutions.

In this manner, [2] introduced blind registration protocols where entities published hashed attributes, digitally signed by an *Identity Provider*, on a public ledger. However, the authors of this work did not address attribute revocation or revocable anonymity. Furthermore, this proposal is susceptible to record linkage attacks; an observer could link multiple attributes to a specific pseudonym and potentially re-identify a certain individual.

The authors in [1] presented a relevant solution wherein users publish attributes to a smart contract, followed by signing and revocation procedures facilitated by Certificate Authorities (CAs). Despite its innovative approach, this proposal allows the public linkage of public keys to attributes, and it requires the aforementioned CAs to be fully trusted.

Finally, Yang et al. [14] addressed the privacy concerns stated above by leveraging Zero Knowledge Proofs (ZKP) in conjunction with blockchain technology. Their novel solution links attributes to "hashclaims" published by CAs. Through the use of ZKPs, individuals maintain anonymity while still being able to prove attribute ownership without directly revealing sensitive information to service providers. Consequently, this work effectively detaches any public association between an individual and her attributes, leaving the *Identity Manager* as the solely actor capable of disclosing these connections. Despite its undeniable merits, this

work still leaves unaddressed some significant challenges. First, the complete confidence placed in *Identity Managers* is a so strong assumption. Second, the fact that each person has one single key pair for managing all attributes implies that, if these keys are lost, so too would all associated attributes.

1.2 Contributions and plan of this paper

Our proposal addresses the drawbacks highlighted in Yang et al.'s work [14]. Primarily, our novel approach effectively limits the extent of information accessible to Identity Providers by introducing a new actor known as the *Identity Manager*, along with a suite of cryptographic protocols. Secondly, this essential actor assumes a key role in the re-identification process of anonymous misbehaving individuals, while safeguarding privacy during legitimate transactions. Lastly, the proposed system tackles the impact of private key loss, thereby ensuring the system's resilience against against such events.

The rest of the paper is organized as follows: Section 2 introduces the new proposal; Section 3 formalizes the various protocols involved; Section 4 evaluates the proposed scheme; and Section 5 reports the conclusions.

2 Our proposal

This section gives an overview of our system. It first starts with the list of actors and their role, followed by the overall sequence of the protocol and it ends with the requirements.

2.1 Actors

The proposed system involves the following four actors:

- *Entity*: This actor is the one holding an identity backed up by an DID. *Entities* are interested in possessing attributes to gain authorization by service providers. *Entities* can be individuals or organizations.
- *Identity Manager (IdM)*: This actor possesses the capability to identify an *Entity* in the event of misconduct and is managed by a consortium of legal authorities. Consequently, the trust demanded by this actor is distributed among the members of said consortium. All actions undertaken by the *IdM* necessitate agreement from the consortium members. To facilitate this, they utilize a shared *Private Database* to store information on registered and authorized *Entities*, and communicate via a secure channel protocol to determine when an *Entity* must be identified.
- *Identity Provider(IdP)*: This actor is responsible for issuing and signing attributes that validate specific properties of an *Entity*.
- *Attribute Smart Contract (ASC)*: This component handles the issuance of attributes, associating them with *Entities* that have been pseudonymized to protect their privacy.
- *Service Provider*: These actors provide services to *Entities* with verified attributes.

2.2 General overview

Our envisaged system, as illustrated in Figure 1, integrates new actors aimed at reducing reliance on Identity Providers, a challenge encountered by previous proposals, while simultaneously ensuring the preservation of user privacy during attribute acquisition processes.

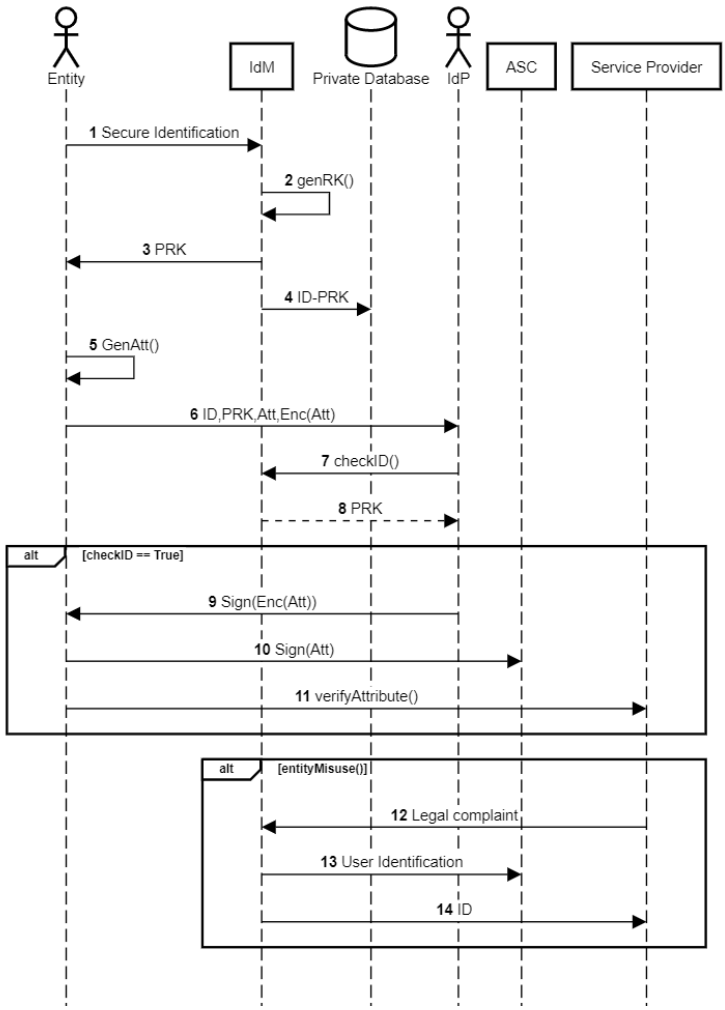


Fig. 1. Overall sequence diagram

The workflow of new scheme comprises the following five distinct phases:

(i) *Entity Registration*: To initiate the process, the *Entity* authenticates itself to the *IdM* through a secure channel (step 1 in Figure 1). Upon successful

authentication, the *IdM* verifies the entity's personal data and assigns a public revocation key (steps 2-3), linked to the ID to identify the user in case of misbehavior (step 4).

(ii) *Attribute Claiming*: In this phase, the *Entity* creates a claim for an attribute (steps 5-6). The *IdP* is then responsible for verifying the claim's validity (steps 7-8) through a cut & choose protocol. Upon successful verification, the *IdP* employs a blind digital signature algorithm to securely sign the attribute without knowledge of its contents. Next, the resulting blind signed attribute is sent back to the *Entity* (step 9).

(iii) *Attribute Registration*: Following this, the *Entity* removes the blinding factor of the blinded signature and registers the signed attribute with the *ASC* (step 10). This registration step ensures that, in the ultimate phase of the process, *IdPs* will possess a reliable means to verify the attribute's authenticity by referencing an immutable public record.

(iv) *Attribute Verification*: During this phase, the *Entity* employs a Zero Knowledge Proof (ZKP) to demonstrate to an *IdP* its ownership of a specific attribute, all without disclosing any sensitive information (step 11).

(v) *Anonymity Revocation*: This phase is activated in the case of an *Entity* that misbehaves. Firstly, the *Service Provider* sends a legal complaint to the *IdM* (step 12). Next, the *IdM* verifies the attribute within the *ASC* (step 13), unveiling the true identity of the *Entity* to the *Service Provider* (step 14), who then undertakes appropriate measures.

2.3 Requirements

To achieve the goals of the proposed system, it is designed to meet the following requirements:

- *Unique Identifiers*: Each *Entity* in the system must possess a unique identifier which goes further than a simple blockchain address and has some legal validity.
- *Attribute Validity and Selective Verification*: Attributes should have validity and must be signed by trusted *IdPs*. These *IdPs* verify user identities prior to signing attributes, ensuring the authenticity of each attribute associated with each *Entity*. The *Entity* should then demonstrate possession of only selected attributes to *Service Providers*.
- *Unlinkability*: The system should prohibit the association of an *Entity*'s identity with their attributes and prevent the linkage of a set of attributes to the same blockchain address.
- *Public Verification*: All attributes and their associated signatures must be publicly verifiable on a public manner.
- *Non-Repudiation*: *Entities* must be unable to deny possessing an attribute they have previously claimed.
- *Preventing False Attribute Issuance*: The system should prevent both individual *IdPs* and collusive groups of deceitful *Entities* and *IdPs* from creating and allocating fraudulent attributes to other *Entities* without any means of identifying the responsible party.

- *Limit IdP's Knowledge*: The system should restrict the extent of information accessible to *IdPs* regarding the *Entities*.
- *Revocable Anonymity*: The system should possess the capability to identify misbehaving *Entities* while preserving privacy for legitimate transactions.
- *Private Key Loss Mitigation*: The system should mitigate the impact of private key loss by an *Entity*, thereby ensuring resilience and data integrity.

3 Protocols

In this section, we describe in detail each phase of the proposed system, providing a comprehensive understanding of it.

3.1 Attribute Smart Contract Deployment

The *Attribute Smart Contract (ASC)* is the first component to be deployed. Its main task is to keep track of all attributes issued by *Entities* and signed by *IdPs*. This component consists of a class which holds the following fields:

- *IdMAddress*: Blockchain address of the *IdM*. It must be provided by the *IdM* when the *ASC* is deployed.
- *entitiesAddresses*: Blockchain addresses of authorized *Entities* which are able to issue attributes.
- *IdPPublicKeys*: Public keys of *IdPs* which can digitally sign attributes.
- *attributes*: List of attributes issued by *Entities* and signed by *IdPs*. Each attribute has the following fields:
  - *name*: Name of the attribute.
  - *value*: Value of the attribute.
  - *issuer*: Public key of the *IdP* which signs the attribute.
  - *expire*: Expiration date of the attribute.
  - *revoked*: Revocation status of the attribute.
  - *hash*: $\mathcal{H}(name, value, issuer, expire)$.

After deployment of the *ASC*, the other actors can interact with it through the following methods:

- *newEntity()*: This method can only be called by the IdM. It adds the *entityAddress* of a new *Entity* into the *entitiesAddresses* field, granting the corresponding *Entity* the ability to issue attributes.
- *newIdP()*: This method can only be called by the IdM. It is used to register a new IdP that will be able to sign attributes. The public key of the new IdP is added to the *IdPPublicKeys* field.
- *addAttribute()*: Authorized *Entities* invoke this method. It accepts a tuple $[Attribute, c, HC, s]$ (further explained in the next section) as an argument, verifies the validity of the signature, and then adds the attribute to the *attributes* field.
- *revokeAttribute()*: The IdM invokes this method. It accepts an attribute's identification as an argument and sets the *revoked* field of that attribute to *True*.

3.2 Initialization

The initial phase entails generating revocation keys, which are held by the *IdM* and employed to identify an *Entity* in the event of misbehavior. To generate these key pairs, *IdM* members utilize a secret sharing scheme without the involvement of any trusted third party (TTP), as described in works like [13].

The sequence of steps in this phase proceeds as follows:

1. The *Entity* authenticates to the *IdM* via a secure channel (i.e., TLS/SSL), providing necessary personal information to verify her ID and a list of authorized Blockchain addresses α^A.
2. The *IdM* uses the ElGamal cryptosystem [5] to compute a public revocation key PKr (note that, $PKr = (p, g, y)$) and its corresponding secret key SKr. This pair of keys can be used for multiple *Entities*.
3. The secret key SKr is partitioned into m shares, which are then disseminated among the n members of the consortium supporting the *IdM*. Consequently, SKr can only be disclosed if m out of n consortium members reach consensus.
4. The *IdM* stores the tuple $[ID, PKr, \alpha^A]$ in the *Private Database* shared by the n members of the consortium.
5. The *IdM* utilizes the *newEntity()* method within the *Attribute Smart Contract (ASC)* to grant authorization to α^A (i.e., the *Entity* associated with this blockchain address) for issuing attributes.
6. The *IdM* sends PKr to the *Entity*.

3.3 Attribute Claiming

The second phase involves an *Entity* claiming an attribute, which must be signed by an *IdP*.

1. The *IdP* generates an RSA [10] key pair (PK_{IdP}, SK_{IdP}).
2. The *Entity* generates a *sha-256 Zero-Knowledge (ZK)* key pair (PKz, SKz), and creates the attribute fields:

$$attribute = (name, value, issuer, expire, revoked, hash)$$

3. The *Entity* encrypts its ID using PKr.
4. For $i = 1$ to b (b is an adjustable parameter) the *Entity* does:
   (a) Selects a random value $r_i \in \mathbb{Z}_q$.
   (b) Computes the ciphertext: $c^i = (c_1^i, c_2^i) = (g^{r_i}, y^{r_i} \cdot \text{ID})$.
   (c) Computes the hash claim: $HC_i = \mathcal{H}(hash, c_i, PKz)$.
5. The *Entity* generates the table T of transactions:

$$T = \{t_1, \ldots, t_b\} \text{ where } t_i = [\text{attribute}, c^i, HC_i] \ \forall i \in [1, b]$$

6. The *Entity*, in possession of the *IdP*'s RSA public key PK_{IdP} (note that, $PK_{IdP} = (N, e)$), encrypts the set of transactions as follows:
   (a) Generates w_i such that $\gcd(w_i, N) = 1 \ \ \forall i \in [1, b]$.

(b) Calculates $w_i^e \mod N \ \ \forall i \in [1, b]$.
(c) Calculates:

$$H = \{h_1, \ldots, h_b\} \text{ where } h_i = \mathcal{H}(t_i) \cdot w_i^e \mod N \ \ \forall i \in [1, b]$$

7. The *Entity* sends the tuple $(H, PKr, ID, attribute)$ to the *IdP*.
8. The *IdP* picks a value $j \in [1, n]$ and sends it to the *Entity*.
9. The *Entity* sends back to the *IdP* the data associated with all transactions except for the chosen j:

$$\{(c_1, r_1, w_1, HC_1), \ldots, (c_i, r_i, w_i, HC_i)\} \ \forall i \in [1, b] \setminus j$$

10. The *IdP* verifies the ID with the *IdM* and confirms the correspondence between PKr and that ID.
11. The *IdP* verifies that the attribute aligns with its agreement with the *Entity*.
12. The *IdP* verifies the next two statements for $\forall i \in [1, n] \setminus j$:

$$c_i = c_i' = (c_1', c_2') = (g^{r_i}, y^{r_i} \cdot \text{ID})$$

$$h_i = \mathcal{H}(t_i') \cdot w_i \mod N \text{ where } t_i' = [\text{Attribute}, c^i, HC_i]$$

13. If the aforementioned verifications succeed, the *IdP* proceeds to compute a blind signature [4] as follows:

$$z = h_j^{SK_{IdP}} \mod N$$

14. The *IdP* sends z to the *Entity*.
15. Finally, the *Entity* retrieves the final *IdP*'s signature on the attribute by computing:

$$s = z \cdot w_j^{-1} \mod N = \mathcal{H}(t_j)^d$$

3.4 Attribute Registration

In this phase, the *Entity* adds the signed attribute to the *ASC* smart contract. This is done as follows:

1. The *Entity* creates a tuple $[Attribute, c_j, HC_j, PKr, s]$ and adds it to the *ASC* by invoking the *addAttribute()* method from one of the addresses α^A.
2. The *ASC* performs the three verifications:

$$\alpha^A \in entitiesAddresses$$

$$issuer \in IdPPublicKeys$$

$$s^{issuer} = \mathcal{H}(Attribute, c, HC)$$

3. If the aforementioned verifications succeed, the *ASC* adds the tuple to its *attributes* field.

3.5 Attribute Verification

In this phase, the *Entity* demonstrates ownership of a specific attribute through a Zero Knowledge Proof (ZKP). The ZKP protocol utilized is zk-SNARK [3]. The procedural steps in this phase mirror those outlined in [14]. Here, we highlight the most pertinent steps. For a more detailed understanding of this process, readers are directed to [14].

1. The *Entity* compiles a set of zk-SNARK constraints $Cst_{response}$.
2. The *Entity* generates a responding waste $N_\delta = sha256(SKz|\delta)$ with salt δ, which avoids reusing proofs and feeds public and private inputs:

$$pubInp_{response} = (attribute, HC, N_\delta) \text{ and } priInp_{response} = (SKz, \delta)$$

3. The *Entity* calculates the witness:

$$\psi_{response} = witnessGen(Cst_{response}, pubInp_{response}, priInp_{response})$$

Using the previously computed proving key p, the proof σ can be generated to demonstrate to the *Service Provider* the possession of the attribute, utilizing the verification key:

$$\sigma = proofGen(p, \psi_{response})$$

3.6 Anonymity revocation

This phase is run if a *Service Provider* detects fraudulent use of a specific attribute by an *Entity*. In that situation, the *IdM* reveals the misbehaving *Entity*'s ID as follows:

1. The *Service Provider* notifies the *IdM* requesting re-identification of the *Entity*.
2. The *IdM*, acting as legal authorities, assess the situation and determine whether re-identification of the *Entity* is warranted.
3. If the verification above is affirmative, the *IdM* retrieves the attribute from the *ASC* smart contract.
4. The *IdM* merges the required number of key shares and obtains the secret key SKr which corresponds to the *Entity*'s PKr.
5. The *IdM* utilizes SKr to decrypt the *Entity*'s ID, which was previously encrypted using PKr.
6. The *IdM* verifies if the address used for the attribute transaction corresponds to one of the addresses associated with the ID stored in the *IdM*'s *Private Database*.
7. If the aforementioned verification succeeds, the *IdM* takes the necessary actions.

4 Discussion

This section discusses the achievement of the requirements outlined in Section 2.3.

- *Unique Identifiers*: Each *Entity* possess a unique ID issued by legal authorities and it is linked to the real world.
- *Attribute Validity and Selective Verification*: *Entities* use ZKPs to verify their ownership of specifically chosen attributes, digitally signed by *IdPs*, thus guaranteeing their authenticity. The selection of which attributes to verify is determined by the *Entity*.
- *Unlinkability*: Attributes are linked solely through decrypting the ID embedded within each attribute. Consequently, an adversary lacking the corresponding decryption key cannot access this information. As a result, only the *IdM* possesses the capability to associate a specific attribute with a particular *Entity*.
- *Public Verification*: The proposed system discloses all attributes on a public blockchain, facilitating public verification.
- *Non-Repudiation and Preventing False Attribute Issuance*: If a malicious *IdP* collaborates with a deceitful *Entity B* to assign a counterfeit attribute to the ID of *Entity A*, the fake attribute will be appended to the *ASC* smart contract, bearing the signature of the *IdP* and linked to an authorized blockchain address of *Entity B*. In the event of anonymity revocation, the *IdMs* can detect the impersonation attack on *Entity A* and identify both the malicious *IdP* and *Entity B*.
- *Limit IdP Knowledge*: *IdPs* sign attributes by means of blind signatures. Consequently, when the signed attributes are added to the *ASC*, *IdPs* cannot link them with the corresponding *Entities*.
- *Revocable Anonymity*: The ID contained in each attribute can be decrypted by the *IdM* if a certain number of members of the consortium agree to decrypt it using a sharing secret scheme.
- *Private Key Loss Mitigation*: In the event of a private key loss, only the attributes specifically associated with the *HC* linked to that key SKz will need to be renewed.

5 Concluding remarks

This work proposes a novel decentralized identity management system with the following key advantages: i) it reduces the trust on the *Identity Providers* while maintaining their crucial role in attribute issuance; ii) it uses blind signatures and ZKPs to prevent *Identity Providers* from linking specific attributes to certain *Entities*, enhancing privacy; iii) it incorporates revocable anonymity allowing the identification of malicious *Entities* when necessary, balancing privacy with the need for accountability; and iv) it reduces the impact of private key loss by allowing *Entities* to use different keys for each attribute.

As future research, we plan to develop a prototype of the proposed system, enabling us to conduct a thorough assessment of its feasibility.

Acknowledgments and disclaimer

This research is supported by Project PID2021-125962OB-C32 "SECURING/DATA' funded by MCIN/AEI/ 10.13039/501100011033 and by "ERDF A way of making Europe"; by Project HERMES funded by INCIBE and the European Union NextGenerationEU/PRTR; and by Grant SGR2021-00115 funded by AGAUR, Generalitat de Catalunya.

References

1. Al-Bassam, M.: Scpki: A smart contract-based pki and identity system. In: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts. pp. 35–40 (2017)
2. Azouvi, S., Al-Bassam, M., Meiklejohn, S.: Who am i? secure identity registration on distributed ledgers. In: Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS 2017 International Workshops, DPM 2017 and CBT 2017, Oslo, Norway, September 14-15, 2017, Proceedings. pp. 373–389. Springer (2017)
3. Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M.: Succinct {Non-Interactive} zero knowledge for a von neumann architecture. In: 23rd USENIX Security Symposium (USENIX Security 14). pp. 781–796 (2014)
4. Chaum, D.: Blind signatures for untraceable payments. In: Advances in Cryptology: Proceedings of Crypto 82. pp. 199–203. Springer (1983)
5. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory **31**(4), 469–472 (1985)
6. Hardjono, T., Pentland, A.: Verifiable anonymous identities and access control in permissioned blockchains. arXiv preprint arXiv:1903.04584 (2019)
7. Lioy, A., Marian, M., Moltchanova, N., Pala, M.: Pki past, present and future. International Journal of Information Security **5**, 18–29 (2006)
8. Meyer, R.: The cambridge analytica scandal, in three paragraphs. The Atlantic **20** (2018)
9. Reed, D., Sporny, M., Longley, D., Allen, C., Grant, R., Sabadello, M., Holt, J.: Decentralized identifiers (dids) v1. 0. Draft Community Group Report (2020)
10. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM **21**(2), 120–126 (1978)
11. Schwalm, S., Alamillo-Domingo, I.: Self-sovereign-identity & eidas: a contradiction? challenges and chances of eidas 2.0. Wirtschaftsinformatik **58**, 247–270 (2021)
12. Tobin, A., Reed, D.: The inevitable rise of self-sovereign identity. The Sovrin Foundation **29**(2016), 18 (2016)
13. Wang, X., Lin, C., Li, Y.: Proactive secret sharing without a trusted party. In: 2013 5th International Conference on Intelligent Networking and Collaborative Systems. pp. 511–515. IEEE (2013)
14. Yang, X., Li, W.: A zero-knowledge-proof-based digital identity management scheme in blockchain. Computers & Security **99**, 102050 (2020)