Policy-based Credential Disclosure in SSI by Using ORCON-based Access Control

Stefano Bistarelli[1], Chiara Luchini[1,2] and Francesco Santini[1]

[1]Dipartimento di Matematica e Informatica, Università degli Studi di Perugia, Via Vanvitelli 1, 06123 Perugia (PG), Italy
[2]Dipartimento di Matematica e Informatica "Ulisse Dini", Università degli Studi di Firenze, Viale Giovanni Battista Morgagni, 67/a, 50134 Firenze (FI), Italy

Abstract

This paper explores some challenges that can arise in authentication and authorisation processes between holder and verifier in the paradigm of *Self-Sovereign Identity* (*SSI*). The authentication phase within the SSI framework is crucial in ensuring the integrity of secure and private data exchanges between the holder and verifier. In particular, we analyse the unauthorised use of credentials, which can be a source of privacy and protection concerns. For instance, sending data to unauthorised third parties could give them access to more information than necessary. We propose a prospective solution for monitoring access to users' personal information. The focus is on defining a *Disclosure Policy* (*DP*) within an *Attribute-Based Access Control* (*ABAC*) model based on the *Originator Control* (*ORCON*) paradigm.

Keywords

Self-Sovereign Identity, Blockchain, Access Policy, ORCON

1. Introduction

In an interconnected digital environment, the accidental exposure of sensitive information to unintended third parties poses a significant threat to individual privacy and security. The abundance of personal data shared online and stored in diverse databases raises concerns about potential mishandling or unauthorised access. The consequences of such disclosures extend from identity theft and financial fraud to reputational damage and physical harm, posing a substantial risk to users' digital identities.

In response to these challenges, *Identity and Access Management* (*IAM*) approaches have evolved from traditional centralised models to more contemporary user-centric ones. The primary objective is to empower users with greater control over their personal data. Various options, including the utilisation of *Personal Authentication Devices* (*PADs*) like smartphones or smartcards, have been considered to store authentication credentials, eliminating the need for a third-party entity such as an *Identity Provider* (*IdP*) [1]. These devices securely manage sensitive information, offering a secure and user-friendly alternative to traditional centralised approaches.

DLT 2024: 6th Distributed Ledger Technologies Workshop, May 14-15 2024, Turin, Italy
stefano.bistarelli@unipg.it (S. Bistarelli); chiara.luchini@unifi.it (C. Luchini); francesco.santini@unipg.it (F. Santini)
https://bista.sites.dmi.unipg.it/ (S. Bistarelli); https://francescosantini.sites.dmi.unipg.it/ (F. Santini)
0000-0001-7411-9678 (S. Bistarelli); 0009-0001-6846-0922 (C. Luchini); 0000-0002-3935-4696 (F. Santini)

However, as noted in previous studies [2], the user-centric paradigm has yet to gain momentum and is viewed as an extension of the IdP model with increased user control. Current understanding suggests that users must authorise or refuse their IdP to share specific personal attributes requested by a *Service Provider* (*SP*). In response to these challenges, the concept of *Self-Sovereign Identity* (*SSI*) emerged as a decentralised approach to identity management. SSI places individuals in control of their data, allowing them to create and manage digital identities across platforms without intermediaries. Grounded in privacy, security, and user control principles, SSI offers advantages over traditional identity systems, reducing the risk of identity theft, fostering trust, and enhancing privacy and autonomy.

Furthermore, a correlated problem is exposing sensitive information to an unauthorised actor [3]. Information exposures can arise from various errors in a product, with severity depending on the context, sensitivity of the information, and potential benefits to attackers. Sensitive information may include personal data, system status, business secrets, network configuration, code, metadata, and indirect information. Different parties, such as users, organisations, administrators, and developers, have different expectations for information protection. Information exposures can occur when sensitive information is explicitly or indirectly injected or when the code intentionally manages resources containing sensitive information but unintentionally makes them accessible. This can result in a loss of confidentiality, which is a technical impact arising from various weaknesses.

This paper addresses potential authentication and authorisation challenges associated with the IAM model, which may lead to misusing users' credentials and jeopardising privacy. The issue we will address is the transmission of user credentials to an unauthorised entity. Our proposed solution involves monitoring access to credentials using ABAC combined with ORCON. The latter enables the originator of the credentials to define access requirements rather than the possessor, as in a *Discretionary Access Control* (*DAC*) model. Furthermore, we selected an attribute-based access control model because it offers greater flexibility and improves access control accuracy. It allows for more precise rules and a greater range of variable combinations without specifying the individual relationships between each subject and each object [4].

The proposed solution involves leveraging the Ethereum blockchain to develop a *Disclosure Policy* (*DP*), which is an access control policy defined in a smart contract whose main objective is the protection of user credentials. The creator of the DP is also known as the originator of the credentials or the issuer in the SSI system. This model is suitable for use in contexts where the credentials require additional protection, such as organisation or company VCs. Furthermore, as previously stated, our model is based on an ABAC model, which considers the verifier's attributes, and an ORCON model, in which the policy is defined by the issuer, i.e., the creator of the verifiable credential.

The rest of this paper is divided as follows. Section 2 overviews the information regarding the types and terminology used in Self-Sovereign Identity and Access Models. SSI authentication and authorisation problems are reported in Section 3 along with proposed solutions based on AC models. Finally, Section 4 defines some related literary works, and Section 5 concludes with suggestions for further research.

2. Background

This section presents the background necessary to comprehend the following topics better. In particular, we explain the SSI concept and *Access Control* (*AC*) models.

2.1. Self Sovereign Identity

In the preceding section, we briefly introduced the concept and evolution of Identity and Access Management (IAM) systems. Before delving into a classification of these models, it is essential to highlight three fundamental concepts: *identification*, *authentication*, and *authorisation*. Identification involves recognising an individual through *unique attributes* or *identifiers*, such as a passport or email address. Authentication verifies the identity of a user, agent, or device, while authorisation grants the right or permission for system entities to access resources [5, 6].

The increasing demand for digital identities has spurred the development of IAM models, offering services related to identity creation, management, and removal, as well as authentication and authorisation for resource access. In traditional IAM models, SP and IdP play key roles. SPs offer specific services and products, while IdPs enable users to authenticate across different services using the same credentials [1].

The transition from centralised to SSI models is depicted in Figure 1. SPs and IdPs are indistinguishable in centralised systems, leading to usability concerns and password reuse. IdPs were introduced to simplify authentication, allowing users to register with a few IdPs and use the credentials across various SPs, reducing the burden on both users and SPs.

Protocols such as SAML, OAuth 2.0, and OpenID Connect were developed to facilitate secure interactions. While these models simplified identifier and password management, they also resulted in the creation of large silos of private information. The evolution from centralised to SSI models reflects a shift towards more secure, user-centric, and privacy-preserving IAM systems.

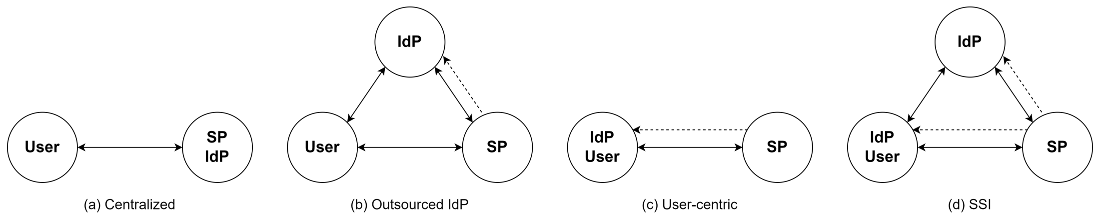

Figure 1: IAM models [1].

As we transition towards more decentralised systems, the distinction between service and identity providers becomes clearer. SSI emerges as a cutting-edge solution, ensuring high privacy for users' information. Recent studies aim to establish an IAM system without a central trusted third party, leveraging SSI. The fundamental concept involves empowering individuals to own and manage their digital identity, fostering a user-centric model [7]. In this framework, users (referred to as "holders") exclusively manage their credentials, typically stored in private storage. These credentials, known as *verifiable credentials* (*VCs*), are issued by entities such as individuals or corporations. Verifiable credentials are tamper-evident and cryptographically verifiable, containing claims representing statements about subjects. The attestation issuer's signature provides cryptographic verification as evidence of the claim's authenticity and the issuer's private key ownership [8]. Holders can generate *verifiable presentations* (*VP*), sharing multiple credentials with verifiers to access specific resources. Verifiable presentations ensure data integrity and authenticity by encoding data in a tamper-evident format. Cryptographic verification safeguards against alterations or tampering, even after data has been shared or transmitted [8].

As mentioned, credentials are typically stored in private storage, but specific information requires public storage. For instance, public keys associated with SSI entities are stored in a *Distributed Ledger Technology* (*DLT*), commonly referred to as a blockchain. Blockchain technology facilitates new methods of personal data management due to its decentralised consent protocol and distributed approach [9, 10]. It serves as a substitute for the registration authority in traditional IAM models. The technology can be categorised into two registry models: the *Identifier Registry Model* and the *Claim Registry Model* [7]. This model can be an extension of the Identifier Registry Model, as it stores identity identifiers and cryptographic data related to identity claims [7].

Verifiable, decentralised digital identification is also made possible by the novel identifier known as *Decentralised Identifiers* (*DID*) [11]. As specified by the *DID controller*, a DID may relate to any entity, including people, organisations, objects, data models, and abstract entities. Unlike traditional federated identities, DIDs are purposefully made to function independently of centralised registries, identity providers, and certificate authorities. In essence, DIDs are *Uniform Resource Identifier* (*URI*) that link a *DID subject* with a *DID document*, enabling reliable interactions about that subject. Every DID document contains cryptographic information, verification techniques, or services, giving a DID controller many options to effectively show control over the DID. By using distributed protocols like *DIDComm*[1] and standards like the W3C DID specification, DIDs create an open infrastructure that promotes interoperability and broad acceptance. They promise to eliminate data silos and improve the effectiveness of digital identity management, and their application spans a wide range of industries, including financial services, healthcare, and e-commerce.

2.2. Access Control models

Access Control (AC) systems are used in a variety of settings where it is necessary to link user characteristics to their roles or groups [4, 12]. Users' access to information is controlled by discretionary protection policies based on the user's identity and authorisations (or rules) that outline the access modes (such as read, write, or execute) that are permitted for each individual (user or group of users) and object in the system. A policy is often connected to a service or resource to increase security. It may be considered a collection of requirements that must be met to access a protected resource. When a user has to take action on an object, such as reading a file, they must be authorised by the policy. Based on the policy check result, they may or may not be able to perform the given action or the specified object. These rules frequently relate to the attributes or qualities of a particular user in a specific situation. These characteristics

[1]DIDComm: https://didcomm.org/

might include user roles in a business or organisation, in which case the model is referred to as *Role-Based Access Control* (*RBAC*) [13], whereas *Attribute-Based Access Control* (*ABAC*) is used in other situations to take into account user attributes [4].

The properties considered by the model and the policy's author alter the kind of AC system. Indeed, there are many different models (RBAC, ABAC, MAC, etc.); however, in this paper, we will discuss just a few of them. In particular, we will describe the ORCON after explaining briefly the MAC and DAC models.

The *Mandatory Access Control* (*MAC*) model is a security paradigm in which a central authority defines and enforces access rules for system objects and users. This implies that the end-user has no management or control over the service's security. This model is often used in high-security environments, such as the military or government, to ensure tight control over access. MAC governs access in a system based on classifying subjects and objects [14].

Conversely, the *Discretionary Access Control* (*DAC*) model assigns access control over objects to those who own them, allowing users to grant or revoke access [15]. This model offers greater flexibility and enables users to manage digital assets. Still, it can lead to potential vulnerabilities as control is based on user discretion and may not always align with the organisation's security objectives. In contrast to the MAC paradigm, end users have total control over their assets under the DAC system, allowing them to choose who can access them. Compared to the other models, particularly the MAC model, this one is seen to be the least restrictive. The choice between these models depends on the specific environment and the need to balance security and user autonomy.

The access control policy known as *Originator Control* (*ORCON*) is positioned between MAC and DAC, as noted [16]. It addresses the gap in access control that MAC, DAC, or a combination of the two cannot fully fill [17]. ORCON is similar to MAC in that access restrictions on original objects are propagated to derived objects. However, it differs from MAC in that policies can be modified on a subject/object basis. This differs from DAC because only the object's originator can modify control privileges. In contrast, in DAC, the owner of a derived object can often modify control privileges on the object or its copies. In summary, original data owners are still able to maintain control over their object even after it has been shared, copied, merged, and authored by other users because it tightly regulates access control and particular access modes at the individual user level [18]. The ORCON designation often identifies secret intelligence sources or procedures vulnerable to countermeasures. This allows the originator to maintain knowledge and oversight of subsequent intelligence usage beyond the initial distribution. The information carrying this mark may be disseminated inside recipient elements and included in other briefings or productions, but only with prior approval from the source [19]. Agencies must develop mechanisms to apply the most restrictive marking to sensitive intelligence and promptly evaluate further distribution requests. However, more control over how and which credentials are revealed could be necessary in some circumstances. As a result, we propose an ORCON AC model integrated into the SSI paradigm.

3. Identity issues in SSI

As was previously noted, protecting one's private information is quite important. Recognising that the material to be provided is sensitive is one of the biggest challenges for a user. A user may be unaware of the various issues, such as privacy violations, arising from the quantity and kind of information revealed. Privacy violations stem from information disclosed in one context seeping into another. Data reduction, or limiting the information sought and received to the bare minimum required, is the advised preventive approach. Global regulations like the *General Data Protection Regulation* (*GDPR*)[2] and the *Health Insurance Portability and Accountability Act* (*HIPAA*)[3] define some rules and practices to adopt when dealing with sensitive information.

In verifiable credentials, issuers should adhere to data minimisation by limiting content to what potential verifiers need. This includes selective disclosure through a signature scheme or the atomisation of information. An example would be a driver's license with more information than is necessary to determine age, such as ID number, height, weight, birth date, and residential address [8]. Recognising the gravity of these risks, individuals and organisations must prioritise safeguarding sensitive information. Strict access controls, robust encryption measures, and comprehensive cybersecurity protocols are pivotal in limiting unauthorised access to personal data. It is about protecting one's information and being responsible custodians of the data entrusted to us by users, customers, or clients. This includes avoiding subsequent disclosure of consumers' information to other third parties, the so-called collusion problem.

One notable privacy risk in SSI revolves around the aggregation of verifiable credentials. Even when information is sourced through distinct channels, possessing two pieces of knowledge about the same subject often unveils more comprehensive insights. Each source may contribute unique perspectives or details that others do not. Comparing these two pieces of knowledge allows a deeper understanding of the subject. In the context of SSI, verifiers may request multiple credentials from users through different channels or a single one, and users are compelled to share these credentials to gain access to specific resources. While this practice is commonly employed for security purposes to verify identity and grant access, it raises concerns about potential abuse by verifiers. The risk lies in verifiers acquiring more information than necessary, potentially compromising the user's sensitive data and enabling the construction of a detailed identity profile.

Managing the actions of third parties with access to personal information presents a complex challenge. Whether it be vendors, partners, or service providers, the potential for data mishandling increases when information is shared outside the immediate control of the data owner. It becomes crucial to establish and enforce stringent contractual agreements, conduct regular audits, and implement secure data-sharing practices to mitigate these risks. Nevertheless, the dynamic nature of digital ecosystems makes it inherently difficult to monitor and control every action performed by third parties.

Blockchain technology, known for its decentralised and tamper-resistant nature, can improve traceability and security in access control systems. It creates an immutable ledger of access permissions and data transactions using smart contracts, ensuring transparency and account-

[2]GDPR: https://gdpr.eu/
[3]HIPAA: https://www.hhs.gov/hipaa/index.html

ability. This reduces reliance on a single point of control, making it harder for malicious actors to compromise. Blockchain is a trusted custodian, ensuring credentials are used according to established arrangements and protecting the ecosystem from threats.

Integrating an access management system in a decentralised system like SSI proves advantageous for both issuers and holders when interacting with the verifiers. Specifically, implementing a smart contract containing an access policy enables any verifier to authenticate the access requirements for information held by the holder. This ensures robust control against unauthorised access and fosters transparency in the administration of access policies.

It should be emphasised that when a holder's information is shared with a third party, managing and monitoring how it will be used becomes complex. In this case, we move from management within an IT context to management within a legislative context. For instance, the *termsOfUse* property in verifiable credentials provides information about the conditions under which a verifiable credential or presentation was issued. The issuer incorporates their terms into the VC, while the holder includes theirs in a VP. It outlines required, prohibited, or permitted actions necessary for acceptance.

Listing 1: Example of termsOfUse property [8]

```
"termsOfUse": [{
    "type": "holderPolicy",
    "id": "http://example.com/policies/credential/6",
    "profile": "http://example.com/profiles/credential",
    "prohibition": [{
      "assigner": "did:example:ebfeb1f712ebc6f1c276e12ec21",
      "assignee": "https://wineonline.example.org/",
      "target": "http://example.edu/credentials/3732",
      "action": ["3rdPartyCorrelation"]
    }]
  }]
```

This feature is expected to be applied in government-issued credentials, guiding digital wallets to restrict usage to similar entities to protect citizens from unexpected data usage. Also, private industry-issued credentials may limit their use to specific departments or business hours. In Listing 1, the verifier ("https://wineonline.example.org"), who is also the *assignee* (row 7 of Listing 1), was prohibited from utilising the information supplied to correlate the holder or subject via a third-party service by the holder (the assigner), who is also the subject. The terms under which the holder generated the presentation would be broken if the verifier used a third-party service for correlation.

3.1. Our proposal

As anticipated, the idea is to create an SSI system with an ORCON-type Access Control model to track access to a given holder's VCs. In particular, we assume that the creator of the holder's VCs, i.e., the issuer, is also the creator of the access policy. Such a system could be applied in specific contexts, such as a corporate or military context [20]. In this case, the company (issuer) could use a decentralised system such as SSI to manage company-related VCs and trace unauthorised access by other verifiers. As mentioned earlier, the main objective is not to prevent unauthorised access to credentials but to track them through decentralised technologies.

According to our concept, it is feasible to confirm that the credentials are being used appropriately since the issuer generates and applies an *Smart Policy* (*SP*), also called a Disclosure Policy, to the holder's credentials. A *Disclosure Policy* (*DP*) is a smart contract that includes an access policy designed to regulate VCs' disclosure. Although the holder fully owns the credentials, implementing a policy ideally aims to limit and trace their use. This is due to the potential risk of the holder sharing their information with unauthorised verifiers, which would violate the issuer's policy. Moreover, in our proposal, the verifier's attributes are verified between the holder and the DP. The DP is responsible for checking attributes and maintaining an updated list of verifiers who can access the holder's credentials. The holder is responsible for requesting the credential from the verifier and verifying its validity, i.e., expired/revoked credentials or invalid signatures. This creates a separation of duties among system components, particularly useful in organisations for maintaining administrative control and preventing security compromises.

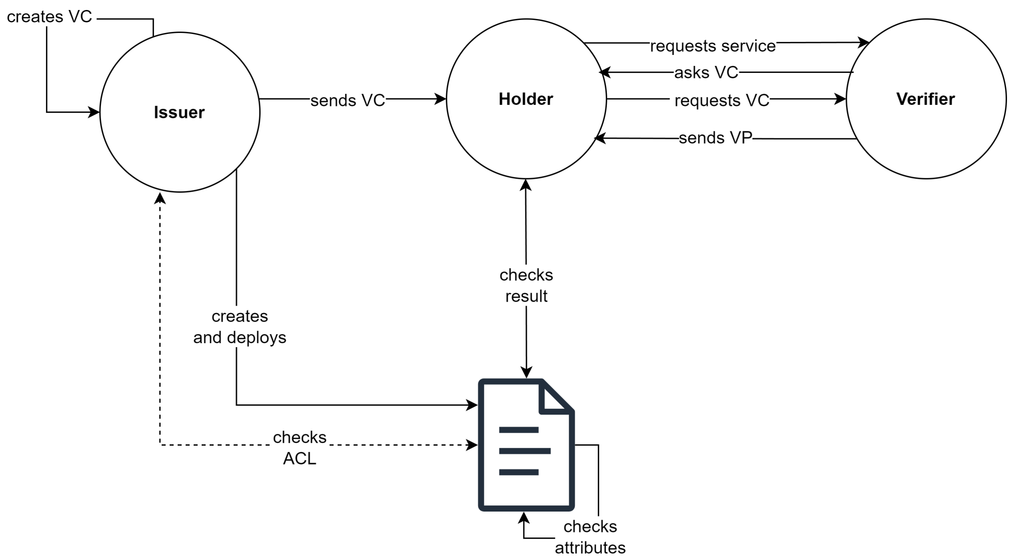

Figure 2: Originator Control Policy with SSI.

Figure 2 summarises the workflow of our proposal. The DP limits and monitors their access by saving information such as the name or date of access. Therefore, the DP can be defined as a credential data access log registry where an *Access Control List* (*ACL*) is used to save the verifier's public information. The ACL is a hash table that stores data as key-value pairs, where the key, in our case, is the verifier Ethereum address, and the value is a structure of different types of information. It is intended as a list showing the last access attempt made by a given verifier and is part of the DP smart contract. In this instance, each time a verifier performs attribute checks, their information and results are overwritten. In this instance, events can be employed to check the access history of a verifier. The issuance of events is contingent upon the verification of attributes. Events represent an abstraction of Ethereum's logging and event-watching protocol. This makes it straightforward to retrieve the history of the results of the attribute check. In this manner, all transactions related to a verifier's attributes can be easily located. However, the ACL structure can also be represented as an array containing the verifier information. In Paragraph 3.1, a comparison is presented in terms of execution costs of the DP methods based on different ACL structures. Table 1 provides an illustrative example of an ACL record. A record comprises a verifier's DID, the name of the holder's VC where the DP is applied, the timestamp related to the time of the access attempt and the verification result.

One potential development of the DP is the incorporation of a time control referencing the most recent positive verification by a specific verifier. The time control entails verifying whether the verifier's attribute check was successful and the time elapsed since the verification was done. This approach enables the establishment of a temporal limit within which a verifier may access the holder's information, after which it is deemed to be unauthorised access. This would be a useful method, for instance, in the case of medical prescriptions that have a short expiry date. In such a case, the verifier (i.e. doctor or pharmacist) would be denied access to a user's prescription that has expired.

We used a local blockchain via Ganache as a test environment. Ganache is a personal blockchain that offers a secure and predictable environment for developing, deploying, and testing distributed applications for Filecoin and Ethereum. We used the Ethereum accounts created by Ganache when the workspace was established. Specifically, we required three accounts for the three entities in our system: issuer, holder, and verifier. Each entity has been assigned a DID and its public and private keys. Resolving the DID means retrieving the on-chain information stored in a DID Registry. In order to interact with the smart contract, we opted for the web3.js[4] library, which is a collection of modules that contain functionality for the ethereum ecosystem.

The issuer creates and deploys the disclosure policy, which includes functions for controlling attributes and saving information. The disclosure policy also performs attribute checks. Public information, defined as the DID of the verifier and their Ethereum account, is assumed to be non-sensitive and, therefore, can be made public. In our case, the verifier's attributes considered for control are non-sensitive information, such as the company name or country of origin. This is because information becomes public when a call is made, and a transaction occurs with the DP. Therefore, privacy concerns may arise when dealing with private or sensitive information due to the transparency feature of blockchain [21].

Key	**Value**			
address	verifierDID	VCName	timestamp	AccessResult
0x6· · · 3491	did:ethr:1337:0x· · · 76147f1477ae	DepartmentInfo	1581314197	true
0x6· · · 7942	did:ethr:1337:0x· · · 854956265644	DepartmentInfo	1584356894	false

Table 1
Example of ACL record.

Following the workflow shown in Figure 2, suppose the holder requests a service from a verifier who asks to access his information to provide the service. The holder knows the DP applied to his work-related VCs, so he asks for the attributes required for the verification.

[4]Web3.js: https://web3js.readthedocs.io/en/v1.2.11/index.html

Then, the verifier prepares a VP with the requested information and sends it to the holder, who verifies the validity and calls the function to check the attributes with the required parameters shown in Table 1. The verifier information is stored in the ACL for successful and unsuccessful verifications. The issuer and the holder must ensure that company-related information is only submitted to an authorised verifier and that the holder's VC is delivered in compliance with the issuer's requirements. If the holder violates the rules and provides their information to an unauthorised verifier, that verifier will unlawfully own material to which they were not permitted access. This will result in no transactions being recorded in the DP history or any records being kept in the ACL.

The Disclosure Policy is written in Solidity[5] and is composed of the following functions:

- **evaluate_attributes()**: This function evaluates the attributes needed for the holder's VC access. It receives the attributes to be verified from the verifier's VP. Then, the function saves it in the ACL along with relevant information such as the verifier's DID, the name of the credential held by the holder and the verification result (either positive or negative). The verifier address is associated with all relevant information by the ACL, including the details above and a timestamp indicating the time of the transaction.
- **check_ACL()**: this function allows to check whether a particular verifier has already performed an attribute verification. It takes an account as input, checks its presence in the ACL, and returns the associated values.
- **add_ACL()**: This function is a private function called by the evaluate function and adds the record to the ACL, as shown in Table 1. It takes as input parameters the attribute to check and all the necessary information required for storage.
- **isAdressListed()**: This function allows anyone to check if a user is in the access list by providing their address. It returns a boolean value indicating whether or not they're in the access list.
- **Time_call()**: This function returns the current timestamp (block's timestamp).

Execution costs. In considering the costs associated with the functions, we have considered the functions `evaluate_attributes` and `check_ACL`. Given that the function `check_ACL` does not alter the state of the contract, it could be executed without incurring any costs. However, we have also considered the possibility of maintaining a record of who has read the ACL via the transactions made and have therefore calculated the cost of this latter possibility. Given the earlier considerations, we calculated the function call cost as GasUsed $\times$ GasPrice. The gas price is determined based on the cost of gas units in Gwei, which is equivalent to one gas unit equal to 15.49 Gwei in April 2024[6]. The gas used for a transaction is retrieved by Table 2, which shows the cost of performing each function. It shows the cost in gas units and the corresponding value in Gwei, also calculated in Ether.

About the data structure employed for the ACL, it is also possible to use an array. This stores not only the most recent access attempt made by a verifier but also the entirety of the access history for the VC. Consequently, in this instance, it would be unnecessary to utilise events

[5]Solidity: https://soliditylang.org/
[6]Source: https://ycharts.com/indicators/ethereum_average_gas_price

Function	Gas Used	Gwei	Ether
evaluate_attributes	232262	3597738	0,003597738
check_ACL	42092	652005	0,000652005

Table 2
Execution cost of DP methods.

to ascertain the access history of a verifier. The primary distinction is the cost of executing the functions. Indeed, we found that the function `check_ACL` exhibited a higher cost in their execution. The iteration cost of the search function probably causes this. Table 3 shows the respective costs of functions performed on the local environment.

Function	Gas Used	Gwei	Ether
evaluate_attributes	229148	3549502	0,003549502
check_ACL	68858	1066610	0,00106661

Table 3
Execution cost of DP methods with ACL as array.

3.2. Example of application scenario.

To better understand the application scenarios of our proposal, we report an example related to bank accounts. Firms are addressing the financial crime business by extending their *Know Your Customer* (*KYC*) initiatives. The KYC strategy is a collection of standards financial institutions and enterprises use to assess the identity, suitability, and risks of present or future clients to detect suspect conduct such as money laundering and financial terrorism before it occurs [22]. KYC regulations, which originated with the *Bank Secrecy Act* (*BSA*) in 1970 [23], have been amended several times since then, including by the *Anti-Drug Abuse Act* of 1986 and the *Money Laundering Suppression Act* of 1994. The KYC structure consists of three steps: *Customer Identification Programme* (*CIP*), *Customer Due Diligence* (*CDD*), and *Enhanced Due Diligenc*e (*EDD*). CIP requires enterprises to collect four pieces of identifying information about a client: name, date of birth, address, and identity number. Additional precautions include verifying that clients are not on government sanction lists, politically exposed persons (PEP) lists, or known terrorist lists. Financial activities are also thought to distinguish potentially dangerous behaviour from normal corporate activity. In our case, suppose the issuer is a bank where the holder has an account. The bank creates a DP for every customer to be applied to their financial reports. The DP defines that only specific banks/companies or government institutions can access this information. The holder wants to create an account with a crypto company. Also, cryptocurrencies incorporate regulations such as *Crypto Anti-Money Laundering* (*AML*) for licenced exchanges to prevent criminals from conducting transactions, which includes KYC. The crypto company, also known as the verifier, requests certain information from the holder, including government-issued identification and financial reports from the holder's bank. However, to access the holder's financial reports, the verifier must first demonstrate that it has the requirements through the DP assessment. If the DP assessment yields a positive result, the verifier may then access the information and proceed with the other two KYC steps.

4. Related Work

In this section, we are going to mention some related works. In particular, the AC model approach with an ABAC methodology applied in an Ethereum blockchain was defined in Maesa et al. paper [24]. This article implements an access control policy as a smart contract to control the holder's access to a verifier resource. The verifier resource could be a smart contract or an off-chain service. This resource is protected by a Smart Policy produced by the verifier, so when a holder requests access to the verifier service, they must meet certain criteria defined in the policy. Additionally, attribute sharing is mediated by ZKP; thus, VCs are not sent plaintext, and the Smart Policy only receives proof of owning particular attributes. Our paper proposes a DP defined by the issuer on the holder VCs. In this case, ZKP was not used during VC sharing, but it may be implemented in the future, particularly for sensitive attributes.

Karthikeyan's master thesis [25] proposes a cryptographic method employing *Ciphertext-Policy Attribute-Based Encryption* (*CP-ABE*) tecnique [26] to implement issuer policy for SSI systems. The VCs are encrypted using a policy that consists of attributes and logical operators, such as "or" and "and" Verifiers can decrypt the credentials only if their attributes match the issuer policy requirements. In our case, a CP-ABE method is not considered for creating an issuer policy. Instead, the policy is stored in a smart contract, which automatically authorises access when a user's characteristics match the policy.

The paper by Belchior et al. [27] also addresses access control models employed with SSI. This paper introduces SSIBAC, which offers decentralised authentication and authorisation for cross-organisation identity management without keeping user-sensitive data. In this case, they require VPs to encode user attributes since their access control engine will determine an access control decision based on ABAC/XACML access control policies. By analysing the schema fields from the VC(s), the access control policy, and the prerequisites for an ALLOW decision, this system employs a function to convert a verifier's access control policy, which contains the rules to access a verifier's resource, to a Verifiable Presentation Request (VPR). In this scenario, the verifier is the policy creator for a resource they own. Furthermore, they do not utilise a smart contract to conduct the authorisation process. Instead, they employ a single access control engine from the verifier's side.

Wu et al. [28] offer an attribute-based access control strategy that uses several blockchain nodes to decrypt data, employs zero-knowledge proof technology to guarantee the accuracy of the decryption result, and encrypts attributes and access policies using an additive homomorphic cryptosystem. The scheme is implemented on Hyperledger Fabric, demonstrating reasonable computation overhead. In contrast, we considered an Ethereum blockchain but did not use ZKP techniques or homomorphic encryption. This is because we assume that the attributes we manage are not sensitive.

5. Conclusion

In this paper, we studied several SSI authentication and authorisation problems and a potential AC control solution for monitoring holder's VC access. During our discussion, we identified various problems with the communication between the verifier and the holder in SSI. These issues, such as potential security risks, can significantly affect the overall system. To address these concerns, we explored access control techniques that can be applied to SSI for traceability. The issuer, also known as the policy originator, can control the authorised access to the holder's VC. If the holder sends credentials to unauthorised users, this can be verified by checking for a transaction to the DP and the record in the ACL.

This section also looks into future developments that might be included in the model. A ZKP may be implemented in the attributes sharing from the holder to the DP, as previously mentioned in Section 4. Instead of receiving unencrypted data, the policy receives proof that the verifier possesses particular attributes. Our model does not handle this option, which might benefit verifier/holder and DP/holder communications.

Additionally, we suggest that specific negotiating strategies may be applied in the holder-verifier exchange. After a positive policy verification, negotiation can also be employed in an AC model that has been proposed. Negotiation techniques can help both parties reach a mutually beneficial agreement on the terms of access control. By engaging in negotiation after policy verification has been successful, the holder and verifier can ensure that the credentials shared are appropriate and sufficient for the requested access level. This additional layer of negotiation can help establish trust and improve the overall security of the access control process.

Another interesting aspect to study is the problem of inference. The inference problem is the intentional disclosure of sensitive information from non-sensitive information. For example, suppose a verifier asks first if the holder is over 18 and then if he is under 20 years old. If they receive a positive response to both questions, the verifier could mistakenly assume that the holder is 19 years old, even if a ZKP is used. The sensitivity of the information, mainly when combined, and the extent to which a verifier can infer from it could be analysed.

Acknowledgments

The authors are members of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM). This work has been partially supported by:

- GNCS-INdAM, CUP_E53C22001930001 and CUP_E53C23001670001;
- European Union - Next Generation EU PNRR MUR PRIN - Project J53D23007220006 EPICA: "Empowering Public Interest Communication with Argumentation";
- University of Perugia - Fondo Ricerca di Ateneo (2020, 2021, 2022) - Projects BLOCKCHAIN4FOODCHAIN, FICO, AIDMIX, "Civil Safety and Security for Society";
- European Union - Next Generation EU NRRP-MUR - Project J97G22000170005 VITALITY: "Innovation, digitalisation and sustainability for the diffused economy in Central Italy";
- Piano di Sviluppo e Coesione del Ministero della Salute 2014-2020 - Project I83C22001350001 LIFE: "the itaLian system Wide Frailty nEtwork" (Linea di azione 2.1 "Creazione di una rete nazionale per le malattie ad alto impatto" - Traiettoria 2 "E-Health, diagnostica avanzata, medical devices e mini invasività").

References

[1] F. Schardong, R. Custódio, Self-sovereign identity: A systematic review, mapping and taxonomy, Sensors 22 (2022) 5641.

[2] C. Allen, The path to self-sovereign identity, 2016. URL: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html.

[3] CWE Content Team, E. Dalci, CWE-200: Exposure of Sensitive Information to an Unauthorized Actor, Technical Report, MITRE, 2008. https://cwe.mitre.org/data/definitions/200.html.

[4] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, Attribute-based access control, Computer 48 (2015) 85–88.

[5] K. Demsey, N. Chawla, A. Johnston, R. Johnston, A. Johns, A. Orebaugh, M. Scholl, K. Stine, Information security continuous monitoring for federal systems & organizations, NIST US Dept. of Commerce, NIST Special Publication SP800-137, available online at http://csrc. nist. gov/publications/nistpubs/800-137/SP800-137-Final. pdf, last accessed June 25 (2011) 2015.

[6] K. Stouffer, J. Falco, K. Scarfone, et al., Guide to industrial control systems (ics) security, NIST special publication 800 (2011) 16–16.

[7] A. Mühle, A. Grüner, T. Gayvoronskaya, C. Meinel, A survey on essential components of a self-sovereign identity, Comput. Sci. Rev. 30 (2018) 80–86.

[8] D. L. Manu Sporny, D. Chadwick, Verifiable Credentials Data Model v2.0, Technical Report, World Wide Web Consortium, 2023.

[9] Q. Stokkink, J. Pouwelse, Deployment of a blockchain-based self-sovereign identity, in: IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), iThings/GreenCom/CPSCom/SmartData 2018, Halifax, NS, Canada, July 30 - August 3, 2018, IEEE, 2018, pp. 1336–1342.

[10] A. S. Rajasekaran, M. Azees, F. Al-Turjman, A comprehensive survey on blockchain technology, Sustainable Energy Technologies and Assessments 52 (2022) 102039.

[11] M. Sporny, D. Longley, M. Sabadello, D. Reed, O. Steele, C. Allen, Decentralized Identifiers (DIDs) v1.0, Technical Report, World Wide Web Consortium, 2022.

[12] V. C. Hu, D. Ferraiolo, D. R. Kuhn, et al., Assessment of access control systems, US Department of Commerce, National Institute of Standards and Technology, 2006. URL: https://csrc.nist.gov/pubs/ir/7316/final.

[13] D. Ferraiolo, J. Cugini, D. R. Kuhn, et al., Role-based access control (rbac): Features and motivations, in: Proceedings of 11th annual computer security application conference, 1995, pp. 241–48.

[14] R. S. Sandhu, P. Samarati, Access control: principles and practice, IEEE Commun. Mag. 32 (1994) 40–48.

[15] J. T. Force, T. Initiative, Security and privacy controls for federal information systems and organizations, NIST Special Publication 800 (2013) 8–13.

[16] J. Park, R. S. Sandhu, Originator control in usage control, in: 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY 2002), 5-7 June 2002, Monterey, CA, USA, IEEE Computer Society, 2002, pp. 60–66.

[17] R. Graubart, On the need for a third form of access control, in: Proceedings of the 12th National Computer Security Conference, 1989, pp. 296–304.

[18] C. D. McCollum, J. R. Messing, L. Notargiacomo, Beyond the pale of MAC and dac-defining new forms of access control, in: Proceedings of the 1990 IEEE Symposium on Security and Privacy, Oakland, California, USA, May 7-9, 1990, IEEE Computer Society, 1990, pp. 190–200. doi:10.1109/RISP.1990.63850.

[19] Control of Dissemination of Intelligence Information Directive No. 1/7., Director of Central Intelligence, May 4 1981. URL: https://www.cia.gov/readingroom/docs/CIA-RDP02B05208R000100180004-1.pdf.

[20] S. Bistarelli, C. Luchini, F. Santini, A military idam system based on ssi and orcon, in: 2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense), 2023, pp. 83–88. doi:10.1109/TechDefense59795.2023.10380863.

[21] Y. Zhang, D. Zheng, R. H. Deng, Security and privacy in smart health: Efficient policy-hiding attribute-based access control, IEEE Internet of Things Journal 5 (2018) 2130–2145. doi:10.1109/JIOT.2018.2825289.

[22] G. Bilali, Know your customer-or not, U. Tol. L. Rev. 43 (2011) 319.

[23] Richard Nixon, Bank secrecy act, 12 U.S.C.: Banks and Banking, 15 U.S.C.: Commerce and Trade, 1970. URL: https://www.govinfo.gov/content/pkg/STATUTE-84/pdf/STATUTE-84-Pg1114-2.pdf.

[24] D. D. F. Maesa, A. Lisi, P. Mori, L. Ricci, G. Boschi, Self sovereign and blockchain based access control: Supporting attributes privacy with zero knowledge, J. Netw. Comput. Appl. 212 (2023) 103577.

[25] N. Anaigoundanpudur Karthikeyan, Cryptographic Implementation of Issuer Policy for Self Sovereign Identity Systems, Master's thesis, University of Twente, 2021.

[26] V. Goyal, A. Jain, O. Pandey, A. Sahai, Bounded ciphertext policy attribute based encryption, in: L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, I. Walukiewicz (Eds.), Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations, volume 5126 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 579–591.

[27] R. Belchior, B. Putz, G. Pernul, M. Correia, A. Vasconcelos, S. Guerreiro, Ssibac: Self-sovereign identity based access control, in: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2020, pp. 1935–1943.

[28] N. Wu, L. Xu, L. Zhu, A blockchain based access control scheme with hidden policy and attribute, Future Generation Computer Systems 141 (2023) 186–196. doi:https://doi.org/10.1016/j.future.2022.11.006.