Self-Sovereign Identity (SSI) Attribute-Based Web Authentication

Biagio Boi[a], Marco De Santis[b] and Christian Esposito[c]
University of Salerno, Fisciano, Italy

Keywords: Self-Sovereign Identity, Hyperledger Aries, Attribute-Based Authentication.

Abstract: Web authentication is primarily based on password usage, representing the weakest link in the entire security chain. The number of services offered over the web is continuously increasing, and with them also the number of required passwords that users need to create and securely store. Despite various standards for password-less or multi-factor authentication, another issue is that most web authentication means use an identity provider (or a federation of providers) advocated to create, manage and check digital identity claims; able to profile user habits related to web navigation and violate rights in terms of privacy. Recently, we are witnessing a radical change of perspective, where identity checks and enforcement are moved away from the providers and more focused on users. Within such user-centric approaches, Self-Sovereign Identity (SSI) has faced progressive popularity, and some authentication mechanisms based on SSI have been proposed. This paper aims to describe a solution based on Hyperledger Aries which is capable to achieve zero-knowledge proof to make an attribute-based authentication and authorization for the web able to cope with the recent legal obligations in terms of privacy.

1 INTRODUCTION

User authentication is among those solutions needed by processors to protect personally identifiable information, as required by the General Data Protection Regulation (GDPR) (Tamburri, 2020), and the current legislation and the recent changes at the national and European level w.r.t. data protection are promoting a considerable evolution of the authentication and authorization solutions. In fact, over the past few years, these mechanisms have undergone significant changes due to technological advances, user behavior changes, and an increase in cyber threats.

Password-less authentication has emerged as an alternative to traditional username and password authentication, which moreover aims at solving problems related to centralized authentication providers. It uses methods such as biometric authentication, hardware-based security keys, or one-time codes to verify the identity of a user without requiring them to enter a password. Some of the notable changes in authentication mechanisms over the last years concern the usage of multi-factor authentication (MFA) (Ometov et al., 2018), which aims at increasing the overall level of security of the authentication process. While the improvement of the authentication schemes is thought to aim for more robust and effective user identification, from the architectural point of view, they are strongly centralized. Every time requests are made, identity attributes and access claims must be directly or indirectly processed by a provider, which can record all the incoming requests. This represents a serious concern as it has non-negligible privacy consequences: the provider can profile user habits in the web navigation. There is a need to move away from this provider-centric practice to decentralize the identity and claim management.

Self-Sovereign Identity (SSI) (Mühle et al., 2018) is a password-less and decentralized mechanism that allows identity verification without relying on a centralized authority. The concept of Verifiable Credentials (VCs) completely matches such a schema, providing self-verifiable attributes using some cryptographic property. In addition, certain VCs schemes also introduce *Zero-Knowledge Proof (ZKP)* to solve the above problems related to privacy.

In this work, we want to exploit existing Hyperledger solutions, such as Aries for proposing attribute-based authentication based on SSI and ZKP in the context of web authentication. The novelties are not limited to the combination of SSI and ZKP

[a] https://orcid.org/0000-0003-3044-5345
[b] https://orcid.org/0009-0004-6514-4168
[c] https://orcid.org/0000-0002-0085-0748

Boi, B., De Santis, M. and Esposito, C.
Self-Sovereign Identity (SSI) Attribute-Based Web Authentication.
DOI: 10.5220/0012121400003555
In *Proceedings of the 20th International Conference on Security and Cryptography (SECRYPT 2023)*, pages 758-763
ISBN: 978-989-758-666-8; ISSN: 2184-7711

but we demonstrate how well-established session handling mechanisms, such as JSON Web Token (JWT) (Jones and Sakimura, 2015), can still be applied in this context.

2 STATE OF ART

To solve problems related to user tracking on activities done with SSO credentials, the concept of giving users the possibility to store their credentials for authentication over multiple platforms started to take place. The first provider moving in such a direction is *FIDO (Fast Identity Online)*, which aim is to create a series of open standard authentication protocols enough stronger, and able to cut out every kind of password-based mechanism. The first proposed protocol has been FIDO UAF (Universal Authentication Framework Protocol), able to use a fingerprint scanner as an authentication mechanism. The real change happened in 2019 when FIDO2 was presented. It is an evolution of the first version in terms of use cases and security, such protocol is based on the paradigm of public/private key pair for user authentication in a secure way, without the need for a password.

A study on possible problems coming from the adoption of FIDO2 has been conducted by (Ghorbani Lyastani et al., 2020), which highlights the problems in recovery at scale and authenticator revocation. These two problems are related to the implementation of FIDO2 which does not support a verifiable registry containing the valid keys, making it impossible for users to invalidate the authenticator device in case of steal. Blockchain support may solve such problems, by adopting a verifiable data registry in which stores released and valid keys and revoke stolen identity authentication data. A different implementation of SSI is based on VCs and Blockchain, which aim is to provide a digital identity using such verifiable credentials in conjunction with a revocation registry, able to be updated when a data branch occurs.

(Ferdous et al., 2023) describe an identity manager framework: SSI4Web, based on blockchain and SSI for Web. It creates an immutable distributed register of user transactions. In such a way, users hold their VCs and can share them based on their volunteer, with a blockchain system that registers every operation done over them. For example, a service may ask the user to present his credentials to conclude a transaction or access a service. A commercial service exists, namely, Sovrin (Reed et al., 2016), which is the first public-permissioned blockchain designed to support SSI and VCs using the architecture offered by Hyperledger Aries and Indy. VCs and SSI are taking the advent of the market of authentication systems; digital identity demands are increasing and with them, the guarantee of privacy, which must agree with GDPR, too.

Anyway, poor studies have been conducted on the usage of such solutions as means for attribute-based authentication and on the security of credentials exchange mechanisms, which could be the weakest link if proper security methods are not implemented. The proposed system aims at considering SSI for attribute-based authentication in conjunction with ZKP; the overall aim is to reduce the data shared by users using a decentralized approach while guaranteeing access to reserved resources.

3 DESIGN

The proposed architecture takes into consideration the SSI paradigm to guarantee the users' privacy and security; creating an ecosystem in which users are encouraged to use digital services thanks to more secure, reliable, unique, and fast authentication. In what follows the architectural design is discussed, and we will refer to the VC as that defined by W3C (Sporny et al., 2022).

3.1 Decentralized Trust Management

To realise decentralized trusted authorities, a verifiable, always available, and tamper-proof registry is needed. Two main approaches exist: Blockchain or Distributed Public Key Infrastructure (DPKI).

Blockchain is an immediate alternative that jumps to the eye when talking about decentralization, despite it isn't the only available one, it can be considered the best choice in comparison with an approach based on DPKI. As discussed in the study conducted by (Li et al., 2020), the major solutions based on DPKI which do not use Blockchain are log-based PKI and Web of Trust, but problems exist for both solutions. The first one ignores data consistency in the log server, while the second one does not provide identity retention and is not friendly to new incoming members.

In such a Blockchain, or a Key Infrastructure, a public and verifiable registry, must always be available for the verification of proofs presented by the users. This register will contain information about the validity of released Verifiable Credentials (VCs), and more importantly, information about the identity of issuing agents.

Optionally, based on the implementation choice is possible that such a registry will also be used for the users' identification, as explained in the following subsection.

A Blockchain solution is preferred on DPKI. A possible considered solution that follows the described requirements is more generally known as *Public Permissioned Blockchain*, where a *Trusted Authority (TA)*, for the security of the proposed system, manages validation nodes by choosing only the trusted nodes and admitting new nodes by checking their trustworthiness. Validation nodes are those responsible for certifying an issuer's identity and validating new credentials issuing. Notice that TA so defined must not be confused with a Certificate Authority (CA) widely known within the context of Public Key Infrastructure (PKI); TA is the node of the network responsible for managing the Blockchain and does not certify or validate any Credentials.

The public part consists of Public Ledger: a registry containing all the information needed from VCs by providing a decentralized and tamper-proof way to store and verify verifiable credentials, enabling organizations and individuals to establish trust and securely share information without relying on a central authority.

3.2 Decentralized Identification

In order to deploy a decentralized authentication mechanism, it's necessary to define how to identify users within the decentralized context. In a traditional centralized system users are identified by a username, an email, or a context-related identifier released by a central entity, which associates the released identifier with a key, which is the demonstration of being the owner of such identity. In our context, there is no central entity able to do this, but instead, there is a set of issuers that release identity and related identifiers.

The World Wide Consortium (W3C) has defined the *Decentralized Identifiers (DID)* (Sporny et al., 2021) completely compliant with the VC data model. The DID Method is used for referring to a precise implementation of DID specification, often associated with a particular verifiable data registry. Some methods make use of a Distributed Ledger Technology (DLT), such as the case of *SOV* using the Sovrin Network; while other ones use the cryptographic property of the method, such as the case of *key* based on public/private key pairs, to verify the ownership of a DID.

Since a verifiable data registry has been adopted by architectural choice, a method based on such a registry could be better in terms of security and performance in the verification of identity, but thanks to the structure of the proposed architecture, also other kinds of DID Methods are implementable.

3.3 Credentials Exchange & Authentication

Following what is defined in data model (Sporny et al., 2022), a VC must include Credential Metadata, Claim(s) and Proof(s). The format used for the proposed architecture is JSON-LD. VC data model shows the main flow, which can be adopted in our architecture. A verifiable data registry is used for maintaining identifiers and schemas; while Issuer, Holder, and Verifier will interact with such register at different levels. Three main phases must be analyzed: Credentials Issuing, Credentials Presentation, and Credentials Verification.

The *issuing* phase is responsible for the creation of VCs and related claim(s) and proof(s). An authorized Issuer can release such credentials by referring to a created schema, or by reusing an existing one, which location depends on the implementation of Decentralized Trust Management. VC Data Model agrees on multiple possible representations of such credentials, their description is out of the scope of this paper; we will consider only the case of JSON representation, which is also the most common. A credential always refers to a context, which contains the field that must be filled for the release of valid credentials. It can be found in a vocabulary way, not blinded to any Blockchain or registry, which is the case of JSON-LD credentials, or in a schema, which is the case of Indy credentials. The currently considered architecture, which is based on Blockchain, supports a verifiable data registry, namely the Public Ledger. Such a choice makes us move in the direction of Indy credentials which gives us the possibility to handle revocation. To issue new credentials, the Issuer uses a previously declared schema, available on the public registry, by asking the Decentralized Trust Management to validate this transaction. Once such a transaction has been validated it will be published in the verifiable data registry, making it available for verification by Verifier. The Holder receives a VC and stores it in a secure space. It's possible to consider the Public Ledger as the verifiable data registry in the case of the adoption of a Blockchain, where only authorized agents can release credentials by interacting with the registry under the supervision of trustable nodes. More in detail, this paper is interested in exploiting ZKP for verifiable credentials, for this reason, the Issuing phase must follow the algorithms defined in 5.2 (Lodder and Khovratovich, 2019), while the information included in a so-defined credential can be found at 3.2 (Sporny et al., 2022).

The *presentation* phase must be done transparently, and the Holder must be informed of the information which is requested by the verifier. The proposed system takes into consideration also ZKP; this means that a method able to prevent information disclosure must exist, this can be an added value in SSI and more in general in attribute-based authentication. In such a schema the Verifier will send a proof request to the Holder, which is able, by using a series of cryptographic techniques, to create a ZKP of possession for each attribute requested from the Verifier, formal algorithms can be found at 7.1, and 7.2 in (Lodder and Khovratovich, 2019).

In the *verification* phase the Verifier verifies the quality of a presented credential. Verifiers do not interact with Issuers during the credential verification process; this is because the proposed SSI system is designed to enable individuals to control their digital identities and personal data, including the credentials that verify their attributes, without relying on centralized intermediaries. Some verification methods require the usage of a decentralized registry - which is the case of Indy credentials. Moreover, an example of self-verifiable credentials is done by JSON-LD credentials which use cryptographic properties for the verification of the validity of a credential. Indy credentials, instead, directly support a revocation registry by making them authenticated also in terms of validity in a given time; for this reason the verification step will involve the Public Ledger.

The attribute that users will share with the Verifier can directly be an email or a role within an organization. As it's possible to see in Figure 1, where the Holder and Verifier are substituted to Browser and Server resp. - since we are considering the case of web authentication - the server creates the JWT after checking the validity of verifiable credentials. After receiving such a signed token, the client, namely the browser will use this token for every request; this is because the JWT is a self-contained token since the payload contains all requested information about users, preventing database interrogation at each iteration. In our system, a JWT will contain the value of verifiable credentials, or more properly, in the considered system, the assertion about a predicate.

4 IMPLEMENTATION

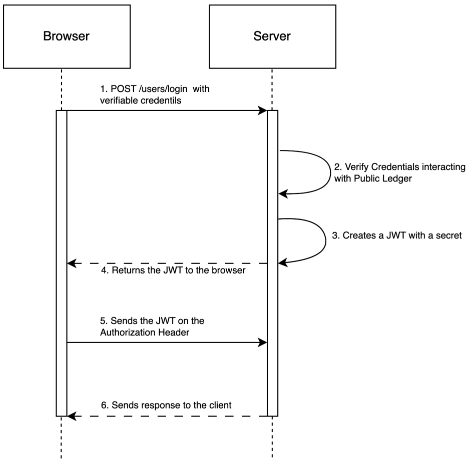
Browser
Server
1. POST /users/login with verifiable credentils
2. Verify Credentials interacting with Public Ledger
3. Creates a JWT with a secret
4. Returns the JWT to the browser
5. Sends the JWT on the Authorization Header
6. Sends response to the client

Figure 1: JWT authentication and request schema.

Hyperledger Aries will be considered for the implementation of our proposed architecture. It provides a shared, reusable, interoperable tool kit designed for initiatives and solutions focused on creating, transmitting, and storing verifiable digital credentials. Such a solution also includes direct support for Indy wallet and ZKP thanks to the adoption of AnonCreds. AnonCreds uses a combination of cryptographic techniques, including ZKP, digital signatures, and encryption, to enable users to prove certain claims about themselves without revealing their actual identity. Hyperledger Aries, with the related implementation named Aca-py, offer the possibility to create and handle revocation using an HTTP API-based communication. It also provides API for the verification of credentials using the Public Ledger and for interacting with the Indy wallet in order to produce *Predicative Proof Presentation (PPP)*. Since all communications must be secured, the public key contained in the DIDDoc is used to cipher the requests coming from a new user.

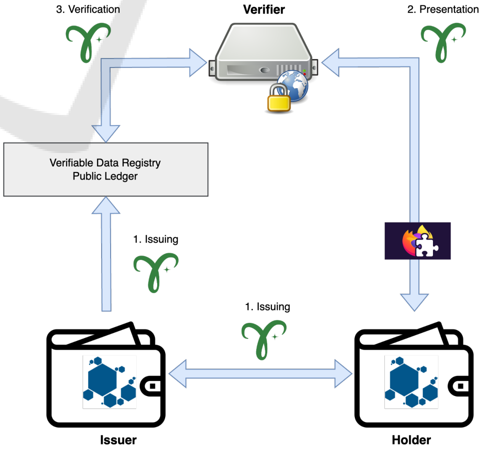

Figure 2: Overall implementation, characterized by the three credentials exchange phases.

Figure 2 represents the overall proposed implementation based on previously described technologies; three main phases exist, whose description can be found in the Design section. Our focus will be on phases 2 and 3, related to the Presentation and Verification, which are the phases needed for attribute-based authentication. The Issuing phase can be considered as already implemented by Sovrin Network, meaning that Holder already has its SSI, which can be used to access to Verifier server. Services that want to use SSI mechanisms for authentication must implement a JWT mechanism within the server and expose their API for the connection and for the exchanging of credentials. The service detects the existence of an extension and will ask the user to produce PPP about one or more attributes.

The presentation begins with a Verifier server that checks if the Holder browser has a Firefox extension; such an extension is able to communicate with the server to transmit the Holder agent end-point needed for using Hyperledger Aries. The extension is configured for creating a listener on presentation requests coming from a Verifier; in such a way, each time a Verifier wants to access Holder VCs, the extension will be informed and the request will be shown to the user using the respective interface. At this point, the Holder will interact with the extension in order to check which fields are requested by the Verifier, and what is more important, the visibility of requested attributes. To correctly deploy a ZKP of possession is necessary that Holder does not share information about the real value of requested attributes but only assertion about a predicate, in a verifiable way. Once the Holder agrees on producing the PPP, the Firefox extension will call the local Aca-py end-point for accept the transmission of PPP. At this point, the Verifier server has the PPP sent by Holder and needs to verify it before giving access to the reserved resources. This verification is simply done by using the end-point offered by Aca-py, which consults the Verifiable Data Registry assessing the validity both in terms of truth and expiration. Once Verifier has verified the VCs can respond to the extension with a JWT; such a token can be used by extension for redirect operations to reserved pages.

From the security perspective, the communication between the Holder and the Verifier is secured using Aries, whose analysis will be considered in the next section. The Firefox extension does not exchange any packets with the Verifier, except the communication of the end-point needed for creating a secure communication using Aries. The extension can be seen as a mediator or a means for improving the user experience. Once the service declares its end-point, the Verifier searches to such end-point for the DIDDoc, which will be used by the Aca-Py client to encrypt the communication and exchange the credentials following the defined schema. The performance evaluation of such an approach is out of the scope of current work but depends on the performance of public permissioned Blockchain and Hyperledger Aries, whose evaluation has been already conducted and reported in (Pflanzner et al., 2022).

5 SECURITY

The implementation of the proposed architecture has been explained by considering the Aca-Py agent, which interacts with the Indy wallet by communicating over an HTTP channel in according with Hyperledger Aries standards. In what follows, we refer to a typical user as a Holder, which is a user with VCs that uses the Aca-Py agent for interacting with the Verifier. Starting from the STRIDE model (Shostack, 2014) we highlight all the possible threats, not only referring to the single agent but considering the overall implementation. In addition, *Forward Secrecy* property and resistance to *Replay Attacks* can increase the overall security of communication, in particular referring to the insecure communication channel.

Spoofing attacks are prevented by securing the communication between agents. Hyperledger Aries uses DIDComm v1 Encrypted Envelope - and is currently working on a second version of such a protocol. The DIDDoc defines the key-agreement mechanisms used for the encryption. Tampering attacks are mitigated thanks to the Blockchain and the messages being digitally signed. An attacker cannot change the value of an attribute since it is associated with a public-facing proof, which is published on the verified data registry. Such registry also contains information on Issuers, namely their DIDs, which means that if the attackers want to behave as an Issuer must be able to gain access to their Indy wallet. Wallet access is protected using a secret key, whose security depends on the Issuer's choice; elevation of privilege is prevented depending on such choice. The only way to obtain information for an attacker from a typical user is by using the Presentation request, which indicates the requested attributes. Information disclosure is prevented thanks to the characteristics of such a request signed by the Verifier and consequential re-signed by the Holder, making it impossible for an attacker to steal information since the Holder can verify the requesting party's identity. Moreover, the information in the verified data registry is not associable with any attribute since it only contains data related to the public-facing proof, not dependent on VCs. Forward Secrecy is guaranteed by adopting session keys; at each session, before communication begins, the parties share a sort of local DIDDoc, which present a local public key, which is different from the public key existing in DIDDoc and which is used for the encryption of the current session communications. Replay and Repudiation Attacks are mitigated using the mechanisms of the message signature, or more precisely, the DIDComm V1 Signed Envelopes. The credential presentation is signed by both parties, in such a way the typical user is sure to answer to the Verifier server, which is unable to behave maliciously, by re-using the proposed credentials since the presentation is signed in conjunction with a challenge. DoS attacks depend on server implementation; a filter or firewall on requests of such a server can be enough to guarantee a good level of prevention.

6 CONCLUSIONS

A distributed approach based on Blockchain for handling authentication in the context of attribute-based authentication has been proposed, in conjunction with a preliminary security analysis. We have planned to use formal methods, such as ProVerif (Blanchet et al., 2018), to verify the security of the used communication protocols. SSIs are increasingly widespread and interoperable (Yildiz et al., 2022). A more specific use case can be found within the context of the Solid project, which has been considered a promising solution for e-government services (Sambra et al., 2016). A schema of ZKP it's been taken into consideration for increasing the overall level of privacy. The main advantages in adopting a Blockchain-based solution are related to the possibility to revoke and update VCs when not still valid; on the counter limitations of the proposed architecture are related to the scalability of permissioned Blockchain, required for secure implementation. Future works may include these kinds of credentials also in other projects, like Algorand (Gilad et al., 2017), which offer different consensus mechanisms, able to guarantee both security and scalability.

ACKNOWLEDGEMENTS

This work was partially supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

REFERENCES

Blanchet, B., Smyth, B., Cheval, V., and Sylvestre, M. (2018). Proverif 2.00: automatic cryptographic protocol verifier, user manual and tutorial. *Version from*, pages 05–16.

Ferdous, M. S., Ionita, A., and Prinz, W. (2023). Ssi4web: A self-sovereign identity (ssi) framework for the web. In Prieto, J., Benítez Martínez, F. L., Ferretti, S., Arroyo Guardeño, D., and Tomás Nevado-Batalla, P., editors, *Blockchain and Applications, 4th International Congress*, page 366–379, Cham. Springer International Publishing.

Ghorbani Lyastani, S., Schilling, M., Neumayr, M., Backes, M., and Bugiel, S. (2020). Is fido2 the kingslayer of user authentication? a comparative usability study of fido2 passwordless authentication. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 268–285.

Gilad, Y., Hemo, R., Micali, S., Vlachos, G., and Zeldovich, N. (2017). Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th symposium on operating systems principles*, pages 51–68.

Jones, M. and Sakimura, N. (2015). Json web key (jwk) thumbprint. Technical report.

Li, Y., Yu, Y., Lou, C., Guizani, N., and Wang, L. (2020). Decentralized public key infrastructures atop blockchain. *IEEE Network*, 34(6):133–139.

Lodder, D. M. and Khovratovich, D. (2019). Anonymous credentials 2.0.

Mühle, A., Grüner, A., Gayvoronskaya, T., and Meinel, C. (2018). A survey on essential components of a self-sovereign identity. *Computer Science Review*, 30:80–86.

Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., and Koucheryavy, Y. (2018). Multi-factor authentication: A survey. *Cryptography*, 2(1):1.

Pflanzner, T., Baniata, H., and Kertesz, A. (2022). Latency analysis of blockchain-based ssi applications. *Future Internet*, 14(10):282.

Reed, D., Law, J., and Hardman, D. (2016). The technical foundations of sovrin. *The Technical Foundations of Sovrin*.

Sambra, A. V., Mansour, E., Hawke, S., Zereba, M., Greco, N., Ghanem, A., Zagidulin, D., Aboulnaga, A., and Berners-Lee, T. (2016). Solid: a platform for decentralized social applications based on linked data. *MIT CSAIL & Qatar Computing Research Institute, Tech. Rep.*

Shostack, A. (2014). *Threat modeling: Designing for security*. John Wiley & Sons.

Sporny, M., Longley, D., Sabadello, M., Reed, D., Steele, O., and Allen, C. (2021). Decentralized identifiers (dids) v1. 0. w3c.

Sporny, M., Noble, G., Longley, D., Burnett, D. C., Zundel, B., and Hartog, K. D. (2022). Verifiable credentials data model v1.1.

Tamburri, D. A. (2020). Design principles for the general data protection regulation (gdpr): A formal concept analysis and its evaluation. *Information Systems*, 91:101469.

Yildiz, H., Küpper, A., Thatmann, D., Göndör, S., and Herbke, P. (2022). A tutorial on the interoperability of self-sovereign identities.