Combining distributed ledgers and passkeys to enhance the security of multiparty information exchange systems.

Quirin Schlegel

2023

1 Context

The last decades have seen an exponential increase in the production of digital data. This is evident from the volume of the annually produced data worldwide, which increased by a factor of 51 between 2010 and 2022 [1]. Consequently, many methods of processing and analyzing data have been devised.

At the same time, digital data and information have become the target of increasingly sophisticated malicious actors[2], which raise cybersecurity requirements. These requirements are especially high for institutions, that provide critical infrastructures or other essential services because their outage would result in substantial damage [3]. However, to improve the quality of service many of these institutions need to expand information exchange capabilities, which satisfy confidentiality, integrity, and availability criteria.

One cornerstone to enable these capabilities is the decentralized management of identities and access rights [4]. In classical data warehouses, the identities would be managed by a single identity provider. In a later iteration of this system, users could also be shared between the domains of identity providers as federated identities [5] [6]. However, the federation of users still requires trust between the identity providers and substantial administrative overhead. Thus the problem arises of how to facilitate this functionality in a more secure and automated manner.

2 Problem

One approach to enable the previously described information exchange capabilities comes in the form self soverin identity solutions, which rely on the usage of blockchain systems. These systems allow storing a shared immutable state in a distributed ledger DL. Additionally, this type of ledger is not controlled by a single entity, but by a group of independent entities, which are incentivized to act according to the ledger's rules. However, these ledgers operate on the

principles of public key cryptography and have a low tolerance for human operator errors. This is caused by the wallet recovery mechanism's reliance on a set of words to be remembered, which are called passphrases. The passphrases are prone to phishing attacks and require their users to be aware of potential attack vectors[7]. Consequently, institutions have to train their staff and implement various security policies to optimize their cybersecurity posture.

Phishing attacks are the most commonly reported cyber crime[8] and should not be underestimated when designing a system with high cybersecurity requirements.

Another approach is to utilize physical security keys in a traditional identity provider setting to incorporate the benefits of public key cryptography. The problem with this approach is that multiple identity providers have to directly connect through a form of identity federation. Also, users would be managed within the constituencies of the individual identity providers, which are individually still dependent on proper IT administration.

3 Research Question

For this work, we want to focus on describing and implementing an identity and access management system for a decentralized dataspace. Also, we put an emphasis on reducing the likelihood of phishing attacks, because this is a major attack vector, that can be addressed from a system design perspective [9]. Thus our research question is: How to incorporate blockchain systems to improve and automate the security of multiparty identity and access management systems?

4 Approach

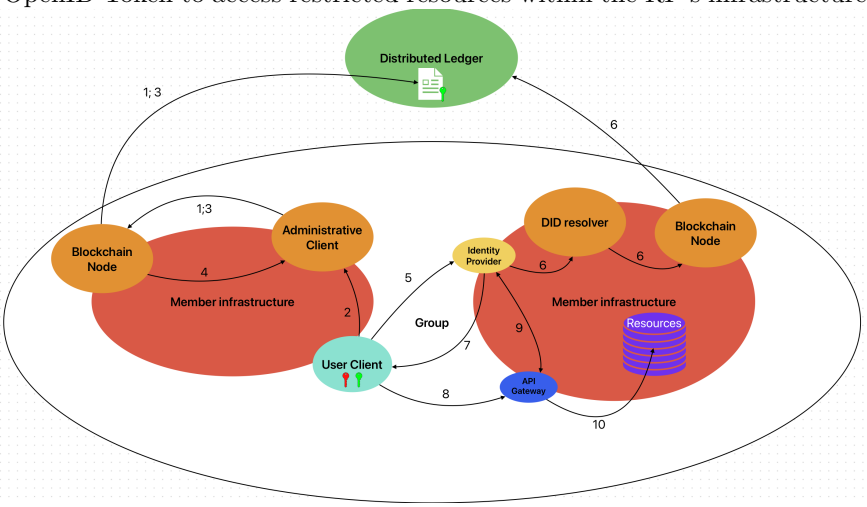
This approach focuses on the use of a DL to store a shared state between the members of a group, which is internally sharing data and computing power. While also leveraging traditional identity provider instances to maintain compatibility with the existing infrastructure of each member.

Initially, a user client has to be registered, to do so an administrative client has to initialize the creation of a verifiable credential (VC)[10] in the form of a decentralized identifier (DID)[11]. The complete conses process for creating the VC is beyond the scope of this work but could include a smart-contract-based poll between the group's members or other smart-contract logic.

Once the user has been issued a VC, which includes his public key the user can proceed to be authenticated by another member's identity provider. In this context, we will refer to the other member as the relying party (RP). The process for authentication is based on the Web Authentication specification [12] to maximize security and interoperability. Hence the identity provider will request the user's public key credential and attest the corresponding private key. However, instead of looking up the user's credentials in a local or federated database, the identity provider will look up the DID corresponding to the user's

public key on the specified distributed ledger. Now the identity provider asserts whether the VC contains access rights, which are relevant to its domain.

If this assertion returns positive the user will be issued an OpenID [13] Token in accordance with the OpenID Specification. Subsequently, the user can utilize the OpenID Token to access restricted resources within the RP's infrastructure.



1. The Administrative Client sends a transaction to the Distributed Ledger to get registered within the federation group's smart contract as an administrator. This process is facilitated by the Blockchain Node which is running in the member's infrastructure.

2. The User Client requests to be registered in the federation group's smart contract as a user.

3. The Administrative Client attests the user's public key, by sending a challenge to be signed by the user's private key. After a successful attestation, the user's public key is hashed and used as the DID of the user's VC. The VC is now written to the Distributed Ledger through the Blockchain Node.

4. success ack

5. Now the User Client can try to login to the Identity Provider of another Member's Infrastructure. Thus the User Client initiates a Web Authentication flow and the Identity Provider asserts the user's public key credentials.

6. The Identity Provider hashes the obtained public key and uses the DID resolver to look up the VC corresponding to the user.

7. If the VC holds access rights that are relevant to the Member's Infrastructure, the Identity Provider will generate an OIDC token and send it to the user.

8. The User Client tries to access the Member's Resources and passes the previously obtained OIDC token along the access request.
9. The API gateway passes the OIDC token to the Identity provider and awaits a response describing the validity of the token.
10. If the token is valid and the scope of the access rights is sufficient the user is granted access to the Member's Resources.

5 Implementation

To implement this Concept the following building blocks are required.

5.1 Distributed Ledger

The Distributed Ledger is a decentralized means of storing data, also it is possible to attach complex rules to the writing access of given data in the form of smart contracts. For this work, we will focus on storing data in the form of DIDs and we will defer the more complex governance methods.

5.2 User Client

The user client stores credentials in the form of Fido2 [14] / Webauthn passkeys on a user's device. Thus the user can be authenticated as the owner of a particular key set. The key set is also separately linked to an identity-providing DID.

5.3 Administrative Client

The administrative client is a wrapper around a full DL wallet and can write DIDs to a DL. These DIDs provide identities to the users under administration and will be read by the identity provider to assert a user's access rights.

5.4 Blockchain Node

The Blockchain Node directly connects to other peers in the blockchain network. It submits transactions to the blockchain and provides data for local applications such as the DID Resolver

5.5 DID Resolver

The DID Resolver[15] can read data from the Blockchain Node and parses DIDs. Additionally, it provides an interface to local applications that allows looking up DIDs.

5.6 Identity Provider

The Identity Provider is a server, which is connected to one or many user databases and handles the authentication of users as well as attesting the user's identity to an RP. For this work, we use Keycloak an open-source identity provider, which is extensible. This also leads to the core innovation introduced by this work. Because the extension to be implemented replaces the need for a classical user database with a DID look-up through the DID Resolver and the subsequent issuing of an OpenID Token. Hence enabling identity federation without a direct trust relationship between the identity issuer and the RP.

5.7 API Gateway

The API gateway handles the users' resource requests and verifies that the required access rights are provided by the OpenID Token.

6 Evaluation

To evaluate the approach that was described above, I would propose a security analysis of the individual components and a subsequent overview of the entire exposed attack surface. The security analysis will consist of an enumeration of common threats regarding the given component and a scenario analysis. The scenario analysis will expect that the given component has been compromised by an attacker. To conclude the evaluation I want to compare this approach with the current status quo, which does not utilize a distributed ledger.

References

[1] Volumen der jährlich generierten/replizierten digitalen datenmenge.

[2] Ömer Aslan, Semih Serkant Aktuğ, Merve Ozkan-Okay, Abdullah Asim Yilmaz, and Erdal Akin. A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics*, 12(6), 2023.

[3] Martti Lehto. *Cyber-Attacks Against Critical Infrastructure*, pages 3–42. Springer International Publishing, Cham, 2022.

[4] Mark Campbell. The road to decentralized identity: The techniques, promises, and challenges of tomorrow's digital identity. *Computer*, 56(6):96–100, 2023.

[5] Mahmoud ElGayyar, Hany El Yamany, Katarina Grolinger, Miriam Capretz, and Syed Mir. Blockchain-based federated identity and auditing. *International Journal of Blockchains and Cryptocurrencies*, 1:179, 01 2020.

[6] Yvonne Wilson and Abhishek Hingnikar. *Evolution of Identity*, pages 23–33. Apress, Berkeley, CA, 2023.

[7] Marwa Alyami, Reem Alhotaylah, Sawsan Alshehri, and Abdullah Alghamdi. Phishing attacks on cryptocurrency investors in the arab states of the gulf. *Journal of Risk and Financial Management*, 16(5), 2023.

[8] Most commonly reported cyber crime categories in the united states in 2022, by number of individuals affected.

[9] Abylay Satybaldy. Usability evaluation of ssi digital wallets. In Felix Bieker, Joachim Meyer, Sebastian Pape, Ina Schiering, and Andreas Weich, editors, *Privacy and Identity Management*, pages 101–117, Cham, 2023. Springer Nature Switzerland.

[10] W3c specification verifiable credentials. https://www.w3.org/TR/vc-data-model/.

[11] W3c specification decentralized identifiers. https://www.w3.org/TR/did-core/methods.

[12] W3c specification webauthn. https://www.w3.org/TR/webauthn-3/.

[13] Openid. https://openid.net.

[14] Fido. https://fidoalliance.org.

[15] Did resolver. https://github.com/uport-project/ethr-did-registry.