TrustVault: A privacy-first data wallet for the European Blockchain Services Infrastructure

Sharif Jacobino
Department of Software Technology
Distributed Systems
Faculty of Electrical Engineering, Mathematics & Computer Science
Delft University of Technology

Johan Pouwelse
Department of Software Technology
Distributed Systems
Faculty of Electrical Engineering, Mathematics & Computer Science
Delft University of Technology

I. Introduction

Internet users today have very little control over where and how their data is stored and used online. Big Tech companies store gigabytes of data about you, and know which online services you use [1]. User data is an extremely valuable asset and is the main source of income for these companies. Public and policy trust in Big Tech has been breaking down in recent years (also called the ”techlash”) following major scandals, rampant misinformation campaigns, and a perceived consolidation of power [2]. There are various movements aiming at halting the power of Big Tech and giving back control to the users. These movements are powered by technologies like blockchains and self-sovereign identity which promise to improve the way we interact online services and with each other.

The European Commission is ramping up its efforts for bringing transformation in the digital sphere with projects such as ”Path to the Digital Decade”. One of their goals is to improve the way citizens, businesses and public administrations share information and trust each other, and simplify verification processes for cross-border services using blockchain technology [3]. Its proposed solution to reduce our reliance on Big Tech is the European Blockchain Services Infrastructure (EBSI). As at May 2022, there was €57 million in funding for large scale EBSI trials [4]. Each European citizen will have its own digital wallet to interact with EBSI.

This work aims to accelerate the European identity and data self-sovereignty movement by providing a EBSI-certified data wallet with advance data sharing capabilities. Citizens can issue credentials within the EBSI framework and define access policies that are automatically enforced for their data based on these credentials. This fine-grained access control for both read and write operations to the user's data wallet gives the user even more control over their data. Our approach preserves user privacy by enabling selective disclosure of only the credentials necessary to satisfy an access policy.

Using verifiable credentials as a basis for attribute-based access control for personal data storage is a novel concept that extends the notion of self-sovereignty over personal identity to personal data. The question that this work aims to answer is: How can a secure personal data storage be created, that gives granular access control to the owner based on attributes extracted from verifiable credentials on the European Blockchain Services Infrastructure. This research question is divided into the following sub-questions:

- How can access to data in a personal data storage be controlled granularly using policy rules?
- How can self-issued credentials be used as access tokens in a privacy-preserving manner?
- How does a privacy-first data wallet add value to EBSI?

The result of this work is called a proof of concept called TrustVault: A privacy-first data wallet deployed on the TrustChain Super App. TrustVault builds upon the concept of personal data Pods first proposed by the Solid team [5]. TrustVault is hosted on the user's smartphone rather than in the cloud. TrustVault also consists of a EBSI conformant SSI wallet able to issue and receive credentials to and from the EBSI network. Besides EBSI credentials, credentials from TrustChain's internal SSI framework can also be used as access tokens. In this proof of concept photos can be grouped together with access policy rules on photo and on group level. Peers on the TrustChain network can share photos using the IPv8 peer-to-peer protocol.

In section II related work is discussed.

II. Related Work

A. Solid

Solid is a protocol developed at MIT that let's people store their data securely in decentralized data stores called Pods [6]. Pods are personal web servers that can store any kind data. The Pod owner has granular control over who has access to the data. Solid uses Access Control Lists (ACL) based on WebID [7] to grant and revoke access to any slice of data contained in a Pod to individuals, organizations, or applications.

B. Self-Sovereign Identity

Self-sovereign identity (SSI) is a decentralised model of digital identity developed to address the shortcomings of the previous internet identity models [8]. With centralised identities, centralized institutions such as governments and banks issue credentials that allow citizens to interact with services and each other. On the internet you would establish an account with every website, service or application. In this model, all the data about you belongs to the issuing party, can't be reused, and is out of your control.

The federated identity model introduces identity providers (IDP). IDPs allow you to have one account that can be used to interact with any service that supports that IDP. This is the mechanism behind the social login buttons (Login with Facebook) widely found on the internet today. Federated identity simplified managing accounts for every service to managing a few accounts at a few IDPs. All our identity data, and information about when or how we use our federated identities is now concentrated in these Tech Giants, raising a lot of privacy concerns.

The rise of blockchain technology inspired the decentralised identity model. This model is not based on accounts with centralised institutions or IDPs but on direct relationships between peers. No party controls or owns the relationship. Users are in full control of their identity data, how it shared and with whom. Peers establish private connections by securely exchanging public keys whereby blockchains serve as decentralised public key infrastructures. This model closest resembles how we manage our identities in the real world: with wallets containing credentials obtained from trusted parties which can shown to other parties to initiate an interaction.

C. European Blockchain Services Infrastructure

The European Blockchain Services Infrastructure (EBSI) is a distributed network that runs a public blockchain to host public and private services that want to leverage the benefits of blockchain technology. The main services that EBSI aims to facilitate are:

1) Notarization: using the blockchain to make digital audit trails and automate compliance checks.
2) Diplomas: giving citizens control over their educational credentials and lowering the cost of verifying documents.
3) European Self-Sovereign-Identity Framework (ESSIF): serve as a verifiable registry and communication channel for an SSI framework across Europe.

Most relevant to this work is ESSIF, as it serves as the network used to issue and receive verifiable credentials. The EBSI blockchain serves as verifiable registries for users and trusted applications. TrustVault users can issue credentials to other participants in the network and receive and verify credentials from other participants using EBSI.

D. Walt.id SSI Kit

The SSI Kit by walt.id is a Self-Sovereign-Identity open source solution, primarily focused on the European EBSI/ESSIF ecosystem [9]. It provides building blocks for key management, issuing, presenting and verifying credentials, and specific EBSI-related functions. Walt.id developed one of the earliest EBSI conformant wallets.

E. Attribute-Based Access Control

Attribute-based access control (ABAC) is an access control model that controls access to objects by evaluating rules against attributes of entities [10]. This allows for more precise access control because of the large set of possible combinations of attributes and consequently large set of possible rules for policies, only limited by the available set of attributes.

F. Tribler IPv8

IPv8 is a peer-to-peer communication protocol developed by Tribler for private and authenticated communication. IPv8 abstracts away physical addresses and allows peers to be identified by their public keys[11][12]. IPv8 is used for communication in the decentralised applications on the TrustChain Super App

III. Methodology

This section discusses our approach to solving the research sub-questions. In section III-A we describe our system for defining access policies at different levels of granularity. In section III-B we describe how verifiable credentials, specifically self-issued credentials can be used as access tokens. Lastly, in section III-C we describe how we integrated EBSI into the TrustChain Super App.

A. Granular access policy

TrustVault uses a tree-structured directory to store data. Every file has a unique path starting from the root directory. Files and folders (including the root folder) have an associated meta-data file that includes access policy rules. To access a file, every policy rule along the the file's path must be satisfied. Practically this means that policy rules are inherited from all parent folders as shown in figure 1.

An access policy is a binary boolean expression tree and the leaves are attribute rules that are evaluated at access time. Figure 2 depicts the access policy: *Age $\geq$ 18 AND (AlumniOf TU Delft OR isFriend)*. To satisfy this policy, the requester has to provide a verifiable credential that asserts that their age is 18+, e.g. a government ID with a predicate proof over the age, and either a proof-of-enrolment from the TU Delft or a friendship token from the TrustVault's owner. A file's complete policy is the conjunction of all the policies on it's path.

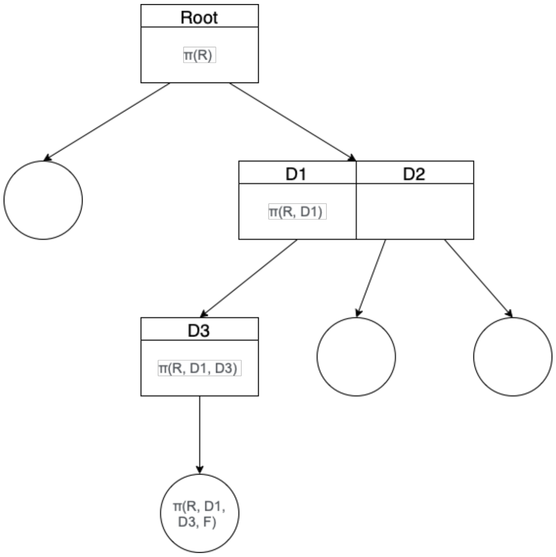

Fig. 1. Directory tree with inherited policies

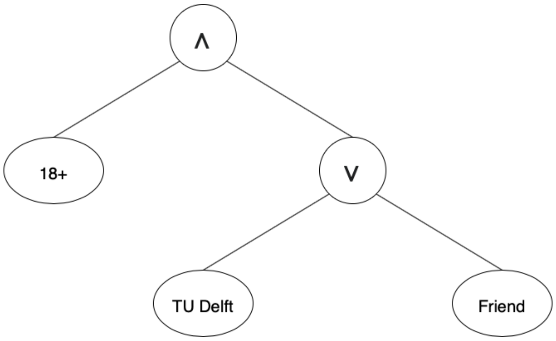

Fig. 2. Access policy tree

The user interface lets the user add or remove nodes to the policy tree. Additionally, the user can define joint read+write access policies or separate read and write policies for more fine-grained control.

B. *Self-issued access token*

As mentioned in the previous section, every policy along a file's path must be satisfied in order to access the file. This path is evaluated against a set of attributes presented by the requesting party in the form of verifiable credentials. Using verifiable credentials enables the requesting party to selectively disclose attributes, and allows the TrustVault to verify the authenticity of the credential, including verifying if the credential was issued by the controller of the vault. Self-issued credentials can serve a similar function as friend requests in traditional social networks. The friendship token in the previous example is a self-issued credential that grants vault access to those that are given one. This makes TrustVault well-suited for social applications.

C. *EBSI data wallet*

EBSI will serve as the main network to issue and receive credentials to and from. This makes TrustVault a secure data wallet for EBSI users. The process of getting TrustVault EBSI conformant was not straight forward. The early prototypes were built using the early versions of the TypeScript cef-ebsi packages for EBSI v1 [13] as part of the EBSI Early Adopters programme. In v1, all operations were API calls to test endpoints. In v2, critical actions including creating, signing, and verifying credentials were moved from the endpoints to libraries running on the user's device. At this point, there were 3 documentation sources for implementing an EBSI that were out of sync in several places meaning that there was a lot of trial-and-error to get the API connection working. In the end, walt.id SSI kit was chosen to handle the issuing and verification of credentials as it is more feature complete in that area.

When initializing TrustVault, the user needs to complete the EBSI on-boarding process which entails scanning a QR-code on the on-boarding page to get an authentication token that is used to get permanent authorisation. In subsequent sessions, the user needs to provide the authorisation token to get short-term session token. The initial authentication step will be dropped once EBSI goes into full production. The ontology used for EBSI credentials can be used to devise appropriate access policies.

D. *Security*

Security of data in transfer and at rest

IV. Evaluation

This section describes our method for evaluating TrustVault's implementation, performance, privacy and security, and outlines the results.

A. *Implementation*

TrustVault is made for Android and is thus completely implemented in Kotlin. The codebase includes a fork of walt.id SSI kit. The open source code for SSI kit is also written in Kotlin. However it is not set up to work with Android. Major modifications were made to make it compatible with Android, including changing IO operations and replacing code that uses packages and libraries not supported by Android.

B. *Performance*

Measuring performance of evaluating policies with full credentials vs derived access token, transfer rate of file.

C. *Privacy*

Evaluate privacy protection and concerns for TrustVault owner and requesting party.

D. *Security*

Evaluate security in terms of data protection, potential attacks. Security properties inherited from SSI in comparison to traditional access control in Solid.

V. Discussion

VI. Conclusion and Future Work

A public blockchain could be used to keep record of every access request made to the TrustVault in a manner that is irrefutable. This further improves the security of TrustVault by making it possible to trace back malicious behavior on an auditable, immutable, publicly verified access log. TrustChain does not support writing arbitrary data to the chain. This functionality would have to be implemented in TrustChain, or a separate ledger could be used solely for this purpose. The storage costs and transaction costs for logging every request on chain may be prohibitive without proper scaling solutions.

References

[1] D. Curran. (2018) Are you ready? Here is all the data Facebook and Google have on you. [Online]. Available: https://www.theguardian.com/commentisfree/2018/mar/28/all-the-data-facebook-google-has-on-you-privacy

[2] K. Birch, D. Cochrane, and C. Ward, "Data as asset? the measurement, governance, and valuation of digital personal data by big tech," *Big Data & Society*, vol. 8, no. 1, p. 20539517211017308, 2021.

[3] European Commission. (2022) European Blockchain Services Infrastructure. [Online]. Available: https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/Home

[4] ——. (2022) EBSI Grants. [Online]. Available: https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/EBSI+Grants

[5] T. Berners-Lee. Solid. [Online]. Available: https://solidproject.org

[6] E. Mansour, A. V. Sambra, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulnaga, and T. Berners-Lee, "A demonstration of the solid platform for social web applications," in *Proceedings of the 25th International Conference Companion on World Wide Web*, ser. WWW '16 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2016, p. 223–226. [Online]. Available: https://doi.org/10.1145/2872518.2890529

[7] W3C. Webid. [Online]. Available: https://www.w3.org/wiki/WebID

[8] A. Preukschat and D. Reed, *Self-sovereign identity*. Manning Publications, 2021.

[9] Walt.id. walt.id ssi kit. [Online]. Available: https://github.com/walt-id/waltid-ssikit/blob/master/src/main/kotlin/id/walt/services/jwt/WaltIdJwtService.kt

[10] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas, "Attribute-based access control," *Computer*, vol. 48, no. 2, pp. 85–88, 2015.

[11] Tribler. IPv8. [Online]. Available: https://github.com/Tribler/kotlin-ipv8

[12] ——. IPv8. [Online]. Available: https://github.com/Tribler/py-ipv8

[13] European Commission. cef-ebsi packages. [Online]. Available: https://www.npmjs.com/search?q=cef-ebsi