JOURNAL OF INFORMATION SYSTEMS ENGINEERING & MANAGEMENT

Journal of Information Systems Engineering & Management, 2018, 3(4), 32
ISSN: 2468-4376

lectito

Federation of Attribute Providers for User Self-Sovereign Identity

Pedro Coelho [1]*, André Zúquete [2], Hélder Gomes [3]

[1] *University of Aveiro,* PORTUGAL
[2] *Department of Electronics, Telecommunications, and Informatics/Institute of Electronics and Informatics Engineering of Aveiro, University of Aveiro,* PORTUGAL
[3] *Águeda School of Technology and Management / Institute of Electronics and Informatics Engineering of Aveiro, University of Aveiro,* PORTUGAL

***Corresponding Author:** toipacoelho@ua.pt

Citation: Coelho, P., Zúquete, A. and Gomes, H. (2018). Federation of Attribute Providers for User Self-Sovereign Identity. *Journal of Information Systems Engineering & Management*, 3(4), 32. https://doi.org/10.20897/jisem/3943

Published: November 10, 2018

ABSTRACT

In a world where people must subject themselves to high scrutiny in every process they initiate, and in a world where the digital environment grows incessantly, we anticipate more online services asking for personal attributes and their need to trust in such attributes. For tackling such need, we describe a proposal for a secure, decentralized and shared repository of certified personal attributes. Individuals benefit because they have full control over the disclosure of their set of certified attributes (e.g. to assert their identities to service providers). The certifying entities benefit from a blockchain-based, resilient and decentralized infrastructure, avoiding the higher costs of an always-on, centralized infrastructure, while retaining the power to issue and revoke certified attributes. Finally service providers benefit from the correctness and freshness of the certified attributes that individuals disclosed to them.

Keywords: self-sovereign identity, distributed ledger technology, personal attributes

INTRODUCTION

Today's technology makes us expect that companies and public institutions will provide many of their services through the Internet. This has a broad range of advantages. First, it makes it easier to fulfill needs without having to travel. Second, it may be designed to prevent a subject from having the fastidious task of gathering a whole set of paper-based information from several parties before applying to a given service. However, in both cases it calls for a general-purpose, attribute attestation (or certification) system, capable of allowing people to collect and remotely provide their digitally certified attributes to service providers on a needed basis.

The aforementioned attributes belong to the set that represents a person's identity (ISO/IEC, 2011). These attributes can be perpetual (e.g., a social security number), ephemeral (e.g., a bank account number) or may vary over time (e.g., the age of the subject). An attribute can also be defined as some piece of property or data about a person that often can be attested by some trusted party (Alpár and Jacobs, 2013). Such attestation is essential to allow a person to claim their ownership when interacting with service providers.

The concept self-sovereign identity rose from a user-centric identity turned into an interoperable, federated identity. A user-centric identity means that users must be central to the administration of their identity. This requires not only interoperability of such identity across multiple locations, with the users' consent, but also true control of identity. This control refers to how, when, for how long and to whom users will disclose portions of their identity, i.e. attributes (Allen, 2016).

With the major improvements on Distributed Ledger Technologies (e.g., Blockchain) in the last years, several use cases for them were identified; one of them is identity management, enabling users to control their identity (Swan, 2015). Those use cases map perfectly into the concept of self-sovereign identity.

This paper describes a system that was designed to provide trusted, personal attributes to service providers and to facilitate the organization and distribution of those attributes by their owners, while adhering to the principles of self-sovereign identity. The system recognizes four different groups of entities (roles): Users, Attribute Providers (APs), Regulation Bodies (RBs) and Service Providers (SPs). RBs are responsible to manage and supervise the set of APs that may participate in the system. APs certify and publish users' attributes upon their request, and may latter revoke them. Users provide trustworthy, certified attributes of their own to SPs on a need-to-know basis. All certificate requests and certified attributes are centrally stored in a permissioned blockchain, maintained by RBs and APs.

RELATED WORK

There were several proposals for using a public repository that people could use to store, validate and present their own attributes. Some of the proposals have a broader scope, while other focus on a specific type of attribute, such as academic certificates.

The Sovrin Foundation recognizes digital identity as one of the oldest and hardest problems on the Internet. Consequently, they propose a protocol and token for self-sovereign identity, based on the mathematical trust provided by a blockchain created with the exclusive goal of identifying someone (Tobin and Reed, 2018). Their solution uses decentralized identifiers (DIDs), which are independent from any centralized registry or certification authority. The Sovrin Foundation also intends to offer the identity owner the capability to selective disclosure verifiable claims using zero-knowledge proofs. With Sovrin, a person can publish their claims and have entities signing them for a proof-making process. Our approach is more limited and pragmatic than this one. In our system, people cannot publish any attribute about themselves, attributes are published by APs authorized by RBs, but only upon owners' requests. While this limits the freedom of people to publish anything about themselves, it also enables SPs to trust on published attributes by trusting on the roots of trust formed by the storage blockchain and the RBs.

The MIT Media Lab and Learning Machine started Blockcerts, a community-based, open-source project aiming at a decentralized implementation of an Open Badge Specification solution (Blockcerts, 2018). This project deals with the issuing, distribution and validation of certificates. These certificates can attest an academic degree, professional achievements or soft-skill acquisitions. They can also be issued individually or grouped in batches. Blockcerts uses the Bitcoin blockchain for verification proposes but aims to be blockchain agnostic, i.e., it should be able to fulfill its goals on top of any type of blockchain. This project lacks an effective way to revoke certificates. The current revocation method involves an amount of cryptocurrency controlled by both parties. A document becomes revoked when one or both parties spend that amount of currency. Their requirement of some amount of cryptocurrency to put attributes into the network and to manage their status increases the costs for institutions. This is negative, since a system that needs to draw fees to maintain itself can create a barrier for global participation. Even considering that fees can limit the amount of unwanted information in the system, our goal was to conceive a system free of fees.

uPort claims to be a secure and easy-to-use system for self-sovereign identity (Lundkvist et al., 2018). It uses the Ethereum blockchain and the InterPlanetary File System (IPFS), Azure, Amazon Web Services (AWS) or any other form of cloud storage to store the hash of an attribute data blob. This project is not limited to individuals, it can also be used to identify devices and entities (e.g., institutions). uPort functions by giving the attribute owner control over a key in its mobile device and by giving it the power to digitally sign and verify claims, actions or transactions. One of the most interesting features about uPort is the ability to recover attributes linked to a key in case of loss or theft; for that, they explore the Ethereum smart contract capability. The uPort blockchain exploitation is similar to ours, since it explores its possibility to provide trust and a kind of decentralized PKI (Public Key Infrastructure). However, it is anchored on a system using proofs of work to reach consensus, therefore depending on fees that can translate to unmanageable amounts of traditional currency given the unstable nature of cryptocurrencies. In our case, we use a permissioned blockchain.

All these projects have different approaches to the identity problem at hand. This heterogeneity provides an opportunity to look at the strong features and possible weakness that each one presents to learn and devise a better solution to the problem of self-sovereignty.

PROPOSED SOLUTION

In this section we describe the basis of a self-sovereign identity system capable of delivering the goal mentioned.

Roles

The system that we propose encompasses four types of roles. We will have the **User** role, which is to be taken by the people (whose identity is comprised of a set of attributes); the **Attribute Provider** (**AP**), which is to be taken by entities entitled to certify that a given attribute is true and belongs to a given subject; the **Service Provider** (**SP**) which is to be taken by entities consumes the User attributes, i.e., that depend on the User attributes to provide some service and, for that reason, may be interested in verify the authenticity and trustworthiness of claimed User attributes, and the **Regulatory Body** (**RB**)**,** which is to be taken by entities in charge of regulating (or supervising) particular business sectors (e.g., the FCC, the regulatory body for electronic communications in the United States of America). This role also includes the management (storage and update) of the public attribute repository.

In addition to these roles we also consider an **Enroller**, an entity responsible for enrolling users into the system, a pre-condition so they can use the system.

Attributes

A personal attribute is a tagged value; the attribute tag describes its semantics. Examples of attribute tags are birthday, weight, address, driver license number, etc.

An aggregated attribute is a set of attributes the make sense together, either because they refer a multi-attribute object bound to a User (e.g., driver license number, type and validity) or because they refer a set of objects that form another object (e.g., the name of the parents).

Hereafter we will always assume that attributes can be aggregated, but we will avoid stating it explicitly, except when necessary, for clarity sake.

Policies

In our system, trust is guaranteed by a set of policies intended to govern the interactions between the subjects bound to the roles described. These policies address the requirements that entities must fulfill in order to be able to participate in system.

The APs are entities that are entrusted to certify and publish Users' attributes. This trust is crucial for AP operation and for the certified attributes acceptance by the SPs. This calls for the inclusion of Regulatory Bodies (RBs) in our solution, which are the regulatory entities responsible for the supervision of specific business sectors, namely for licensing APs and to verify that they operate according to that sector rules. Our solution eases the RB monitoring task as they are responsible for the enrollment of new APs and they are able to verify the correctness of certified attributes issued by the APs operating in their respective business areas. With this we believe we are increasing the overall trust level of the solution.

Regarding attribute certification process, the User (role) is the client of AP (role). In order to request the certification of an attribute, users can proceed in one of two ways: (i) they can directly (via a channel external to our solution) request the AP to certify their attribute, in which case it results in the issuance, by the AP, of an attribute certificate that is published to the Users address (i.e. their Identifier) in the blockchain based repository; or (ii) they can submit a request for an attribute certificate, addressed to a specific AP, directly in the blockchain. Before requesting an attribute certificate through any of the two methods, first the user needs to enroll in the system by an Enroller.

Regarding the exploitation of certified attributes, we have the following set of policies. Upon the issuance of an attribute certificate, the owning User stores the certificate in a personal repository (wallet) linked to the reference (or address) that he provided the AP. The wallet also contains a set of secret credentials (private keys) that allow the User to reclaim the ownership of his certificates. Thus, when a User is interacting with an SP, to prove the possession of a certified attribute he wants to disclose, the User must provide the reference to the attribute certificate and use the keys in his personal wallet to prove the ownership. To accept an attribute certificate provided by a User, SPs need only to trust in the issuer AP (and respective RB) and verify that the interacting user is in fact the certificate's owner. However, in no case an AP can limit, or even know, the SP to whom an attribute certificate is going to be presented.

Attribute certificates can have a lifetime that restricts the time interval where it is valid. Also, attribute certificates may be presented many times to different SPs, or they may be restricted to be used only a limited number of times. Certificates can evolve along time, but such evolution must be agreed upon both by the owner (User) and the issuer (AP). Certificates can, at any time, be revoked solely by the issuing AP. The revocation, however, does not eliminate the certificate; it simply cannot be used anymore as a valid attribute after the

revocation date. In other words, attribute certificates, once disclosed to an SP, can still be checked by that SP, regardless of its posterior revocation, this can also serve to prove that the user once owned a certain attribute. Thus, certified attributes form a chain of values and dates with a single reference. The chain can include several attribute updates or a single and final revocation.

Regarding privacy issues related with the exposition of attribute certificates in a public repository, we have the following policies. Each certified attribute is bound to a unique User pseudonym. The set of pseudonyms of each User is managed through their wallet. Thus, it is impossible to know from the public repository who is the real owner of an attribute and the set of attributes belonging to a single User. Furthermore, attributes values in attribute certificates may be exposed (i.e., in cleartext) or obfuscated. This last case allows the attribute owner to selectively expose the attribute value only when dealing with a given SP. For disclosure there are two access control mechanism that can overlap one another, a mechanism of authorization provided by the repository and the obfuscation mechanism . Note that the real attribute value does not need to be conveyed to the RB that will audit the publication of the certificate, since RBs care about attribute tags, not their values, to perform their supervision task.

User Wallet

A User wallet is a fundamental piece for managing the issuing and the exploitation of the user certified attributes. The wallet assures the secrecy of the credentials used by its owner, both when requesting the issuing of a certified attribute (to an AP) or when claiming its possession (to an SP). The wallet also keeps, directly or indirectly, the references to all the certified attributes belonging to its owner. Finally, the wallet contains the elements that enable the revealing of obfuscated attributes.

Technological Solutions

For implementing this system, we are considering the following approaches.

Regarding the request of an attribute, this can happen through a bit commitment submitted as a transaction to the network.

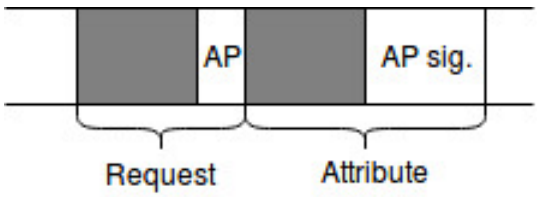

Figure 1. Bit commitment with request and later attached attribute

The goal of this bit commitment is to create further proof of intent of having is attribute in the ledger. This bit commitment is constituted by a AP ID field, which is clear in order to reach the designated AP, and a value field, which is obfuscated in order to protect the users and not letting through their intentions.

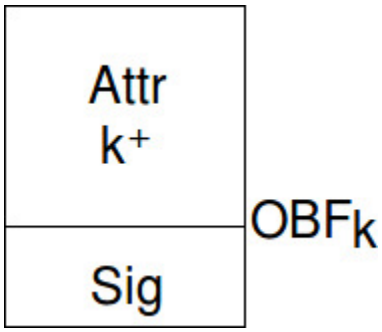

Figure 2. Bit commitment obfuscated portions

The value field as illustrated above includes several information necessary for attribute certification. It has the needed information to identify the attribute that is being requested; the address of destination of the attribute (it may be the same or a different one); and a signature; this obfuscation in made with a symmetric key negotiated previously between the two parties that can be revealed to authorities later in a dispute case.

Even though the request can happen within the ledger, there is the need for a different and direct channel where the AP can authenticate the user. In the figure below the users are authenticated through a secure channel that they had previously established (e.g., a mobile app), through this channel the user can then obtain a Request Code to embed in the obfuscated portion of the bit commitment.

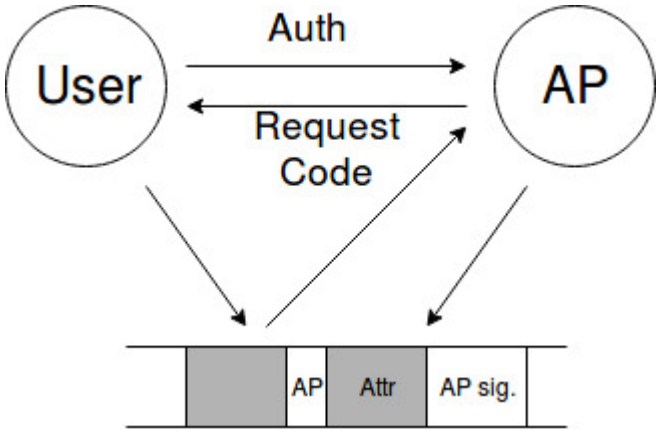

Figure 3. Authentication and flow of request

Each certified User attribute is bound to a unique public key. The corresponding private key is kept in the User wallet and is used to assert the possession of the certificate. The public key is, therefore, the reference to a certificate.

Obfuscated attributes are the digest of the actual attribute with a random, secret key. The secret key is managed by the wallet and can be generated from the private key bound to the attribute's certificate. This key is conveyed to the AP, during the certificate issuing, and to the RB, in order to allow it to check the suitability of the certification request.

For tackling the complete auditing of the system upon a dispute, APs keep a log with all issued certificates, as well as their secret key when the certified attribute is obfuscated. This log must also contain some proof (e.g., a digital signature) of the User that requested the certificate.

Each certificate contains a digital signature of the AP that issued it, together with a digital signature of the RB that published it. The first signature asserts the correctness of the attribute binding (to a particular User), the second signature asserts the suitability of the attribute for the RB's business.

Since each certificate will have a unique reference (its public key), this can represent a problem for the wallet of its owner, since it may have many references to manage. A possible solution to this problem is the use of hierarchical deterministic wallets, such the ones introduced by Bitcoin Improvement Proposal (BIP) 0032. These are able to generate a tree of key pairs from a single seed, which the User owns and controls. The value of the attribute, which can be stored in the wallet, can be used to derive the key pair associated to its certificate (Robles and Appelcline, 2018).

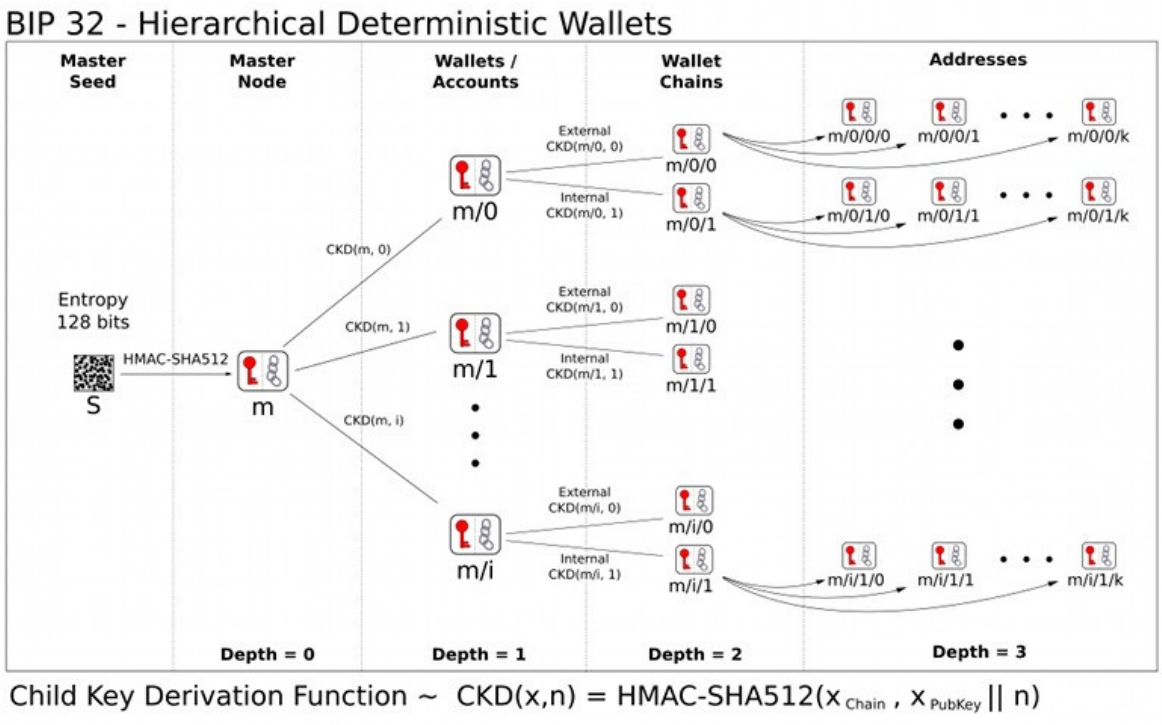

Figure 4. Hierarchical Deterministic wallet derivation tree

The system detailed so far uses digital signatures as a one-to-many form of asserting the authorship of a particular piece of information. In particular, signatures of APs and RBs are necessary to unequivocally identify the certificate issuer and to guarantee a certain degree of trust on the quality of the certified attribute. Anyone can be a signature validator, but Users and SPs and main interested parties in doing so. Therefore, we need some kind of PKI to support the issuing of public key certificates (PKC) for both APs and RBs. One way to achieve this is by setting the roots of the PKI within the blockchain itself, and by allowing SPs and Users to trust on at least one RB. This way, the Users and SP only need to trust on some RB to build trust on the blockchain and, transitively, on the PKCs of both RBs and APs.

Trust Model

The society, in general, trusts on RBs to do a proper supervision of their business sector and to keep a coherent blockchain among themselves. Within RBs, they only trust on authenticated RB peers to enter on blockchain definition update agreements.

Each RB has a list of federated APs, which are not fully trusted. Namely, attribute certifications proposed by APs may be checked by the RB to assert its suitability. On the other hand, an RB trusts that a federated AP performs a correct certification, i.e., that when the AP says that attribute A belongs to User U such binding is true (at least, at the time of the certification). Abusive AP certifications cannot be immediately detected by RBs but can be later identified by auditing APs' certification logs.

Each certificate must possess an AP digital signature, which prevents RBs from forging certificates issued by APs. Therefore, APs do not need to trust on the RBs on which they are federated. However, this also implies that RB cannot issue the PKC of the AP they federate, otherwise RB could forge attribute certifications on behalf of APs. This trust issue can be solved in different ways (for instance, by requiring each AP PKC to have the signature of more than one RB).

CONCLUSION

In this paper we proposed the foundations for a federated ledger for regulated self-sovereignty. The system allows people to keep full control over the ownership and distribution of their attributes, which are critical self-sovereignty requirements, while restricting the set of certified attributes to the ones issued by APs authorized by RBs, also a critical requirement for ensuring SPs about the quality of the attributes presented by people.

Self-sovereignty is a subject on which interest is growing fast and we will see a rise in research initiatives such as ours. With this in mind, we must be permeable to new findings and concurrent advances relatively to the one presented.

As feature work, we will identify some use cases and we will start to detail and implement a prototype, properly integrated with the applications used by people and a set of SPs. These will be very important for asserting and demonstrating the usefulness of the system to a future world were many services could be remotely provided to properly identified individuals.

REFERENCES

Allen, C. (2016). The Path to Self-Sovereign Identity. Available at: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html (Accessed 28 June 2018).

Alpár, G. and Jacobs, B. (2013). Credential Design in Attribute-Based Identity Management. In *Bridging distances in technology and regulation*, R. Leenes and E. Kosta, Ed. Wolf Legal Publishers, pp. 189-204.

Blockchain Credentials. (2018). *Blockcerts*. Available at: https://www.blockcerts.org (Accessed 29 June 2018).

ISO/IEC. (2011). Information technology -- security techniques -- a framework for identity management -- part 1: Terminology and concepts. *International Organization for Standardization, Geneva, Switzerland.*

Lundkvist, C., Heck, R., Torstensson, J., Mitton, Z. and Sena, M. (2018). *UPort: A Platform for Self-Sovereign Identity*.

Robles, K. and Appelcline, R. (2018). Web off Trust Info. Available at: https://github.com/WebOfTrustInfo/ID2020DesignWorkshop/blob/master/draft-documents/hierarchical-deterministic-keys-for-bootstrapping-a-self-sovereign-identity.md

Swan, M. (2015). *Blockchain: Blueprint for a New Economy*, 1st ed. Sebastopol: O'Reilly.

Tobin, A. and Reed, D. (2018). *Sovrin ™: A Protocol and Token for Self- Sovereign Identity and Decentralized Trust*. Sovrin Foundation.