Blockchain's Relationship with Sovrin for Digital Self-Sovereign Identities

Daan Weller, Raoul Dijksman

***Abstract*—Sovrin is a blockchain technology for self-sovereign identities. This paper investigates what the blockchain technology accomplishes for the Sovrin network. Three cases are analysed to determine the differences between the physical, digital, and Sovrin domains. The Sovrin network is able to implement a system that is user-centric and follows the principles proposed by Sovrin's self-sovereign identity model. Credentials on the Sovrin network allow users to identify themselves in a decentralised manner and privacy-preserving manner. Sovrin's transaction flow of credentials incorporates many characteristics that are also present in the physical world. On top of that, Sovrin allows for better verification processes and real-time revocations compared to the physical world.**

We analyse the characteristics of the Web of Trust and compare it to the Sovrin network. Then, the main problems are identified that are solved by Sovrin. These problems are further analysed, and comparisons are made to the Sovrin implementation. We discuss if blockchain technology is a required part of the implementation of Sovrin to solve the identified problems. Finally, we conclude that blockchain technology is one of the possible solutions to the problems found.

I. Introduction

Ownership of user-centric identities today remains a hard problem to solve and suffers from centralised control [1]. Entities that provide digital identities in a centralised manner have the power to take away an identity at any time. Self-sovereign identity is a concept where users are not only at the centre of the identity process but have complete ownership of their own identity as well. Christopher Allen has proposed a set of principles for self-sovereign identities [2]. He preludes that being the ruler over one's own identity "requires not just the interoperability of a user's identity across multiple locations, with the user's consent, but also true user control of that digital identity, creating user autonomy". In order to accomplish this, a self-sovereign identity must not be locked down to one entity.

The Sovrin network is a blockchain technology for self-sovereign identities. It enables secure exchange and verification of identity information (e.g. subject is 18 years of age or older) for business transactions. Potential savings are estimated to be over 1 billion euro per year for just the Netherlands [3]. Sovrin claims that blockchain technology enables the implementation of self-sovereign identities [4]. At the time of writing, it is unclear what the blockchain technology accomplishes for the Sovrin network.

In this paper, we will answer the following question: To what extent does the Sovrin network require blockchain technology for Self-Sovereign Identity? To answer the main question, we will also answer the following sub-questions:

- What are the characteristics of a credential transaction on the Sovrin network?
- What is the difference between a traditional web of trust and Sovrin's implementation using blockchain?
- How are the main properties of the Sovrin network supported by a blockchain?

The remainder of this paper is structured as follows: In section II, we give a literature review of relevant research. The methods of our research are described in section III. Section IV describes a model for the flow of credentials between parties and three cases to analyse the difference between them. Sections V and VI compare Sovrin to a Web of Trust to analyse the association with a blockchain. Finally, we discuss our findings in section VII and provide a conclusion in section VIII, as well as suggestions for future research topics.

II. Literature Review

First, the current knowledge including substantive findings to the term Self-Sovereign Identity will be investigated, followed by the work done by the W3C to standardise identity identifiers and the exchange of credentials. The underlying system of Sovrin, called DKMS, is investigated. Then related research to the topic of Blockchain and user privacy is reviewed. Finally, a summary of the current theoretical knowledge of Sovrin is given.

A. Self-Sovereign Identity

Self-Sovereign Identity (SSI) is a concept introduced at the beginning of the twenty-first century, and originates from the definition and laws of digital identity by Kim Cameron in *The Laws of Identity* [5]. In this work, Cameron proposes definitions of identity and *claims*, as well as a set of laws to abide by in favour of the user. Identity is defined as a set of claims, where claims are "an assertion of the truth of something, typically one which is disputed or in doubt", according to Cameron. A few years later, the application of these definitions are expounded upon in *A User-Centric Identity Metasystem* [6], which describes possible implementations of such a system on a high level.

One of the definitions of SSI was introduced by Christopher Allan, who provides ten principles of SSI based on the previous definitions of identity provided by Cameron and the W3C Verifiable Claims Task Force [2]. Allen argues that the centralised identities in use today respect user consent to some degree. The ten principles proposed are therefore mainly centred around and to the benefit of the user.

Finally, another definition of SSI is given by Phil Windley, the chair of the Sovrin Foundation Board of Trustees [7]. In order to explain the basis of SSI in an online environment,

Windley introduces the term Multi-Source Identity (MSI) [8]. MSI is the concept of having a collection of relationships and credentials instead of an identifier oriented approach. So instead of performing authentication based on shared secrets between the user and each service provider, a set of claims made by the user are taken into regard to validating the posed identity. According to Windley, MSI involves new relationships between three different actors: credential issuers, credential holders and credential verifiërs. In a MSI system, any person or organisation can fulfil these roles. Credential issuers determine what credentials are used to what end, and specify the format of the credential. Credential holders are the end users of the system, e.g. a person, organisation or a device. The credential holders determine what credentials they need and are responsible for managing them. Credential verifiërs are the actors that determine what credentials to accept and thus what credential issuers and holders to trust. The set of necessary credentials to trust a given credential holders differs per situation and is generally referred to as taking a risk.

Windley states that the difference between MSI and SSI is the degree of control an identity holder has over his credentials. SSI supposedly implies that the identity holder has full control over its credentials. This means that the identity holder gains control over who has access to the credentials and can use them in a privacy-preserving manner. The identity holder may also acquire a multitude of credentials from different issuers. The involved parties in the network behave like peers and the roles these parties employ differ per transaction. Also, every party determines who (or what) to trust individually.

B. Verifiable Credentials

The W3C Verifiable Claims Working Group also provides a short definition of self-sovereign system by identifying attributes a given *verifiable credential* system should have [9]. The Verifiable Claims Working Group are working on a definition in an attempt to standardise the way to represent and exchanging credentials [10]. The Verifiable Credentials Data Model Editor's Draft, at the time of writing the latest editors draft is from 25 January 2019. Claims are statements about subjects, and credentials are sets of claims issued by a party. A verifiable credential is a tamper-evident credential that can be cryptographically verified.

In order to provide contrast, an initial comparison is made: in a self-sovereign system, the users exist independently from the services as opposed to service-centric systems where users are bound to some particular service. The working group identifies three different parties: credential issuers, credential holders and credential verifiers. A list of principles of such a system is then given with the focus on the credential holder, or the user of the system, from which self-sovereignity supposedly follows. This boils down to the user being able to "control" and "own" their credentials and identifiers, as well as how to manage them and when to use them. Another distinction is made for the software used to store and manage the credentials, named agents. Users should be able to choose and change agents used freely and may use the credentials stored by an agent in transactions without revealing who the other party is. Also, issuers and verifiers do not need to trust the agent, only each other.

The exchange of verifiable credentials is made possible by having a unique identifier. The *Decentralized Identifier* definition by the W3C defines a globally unique identifier in an URI format called a DID[11]. The holder can self-register a DID on a DID-compatible distributed ledger. This means that no central registration authority is required. Each DID points to a DID document, a JSON object containing public verification key(s) and addresses of off-ledger agent(s). The subject is able to prove ownership of a DID using these public verification keys.

C. Decentralised Key Management System

Decentralised Key Management System (DKMS) is an approach to cryptographic key management where there are no centralised authorities [12]. It is an alternative to the centralised public key infrastructure (PKI) architecture. DKMS allows anyone to manage their authoritative keys without other parties having to approve. While at the same time making these changes immediately viewable to everyone without the need to update a certificate store. This is done by using self-registered DIDs, so no central registration authority is required. This creates a trustless starting point from which trust between DID-identified peers can be built up through the exchange of verifiable credentials.

DKMS works on top of blockchain technology for decentralised access to cryptographically verifiable data. This ledger is the initial *single source of truth* of the system. It then adds on top of it a *web of trust* for the exchange of keys, formation of connections, and to issue/accept verifiable credentials from any other peer. This changes PKI into decentralised PKI (DPKI) [13]. DPKI's purpose is to "provide a simple, secure, way to generate strong public/private key pairs, register them for easy discovery and verification, and rotate and retire them as needed to maintain strong security and privacy".

D. Blockchain

Satoshi Nakamoto, founder of Bitcoin, a digital currency created in 2008, introduced the concept of blockchain technology [14]. A blockchain is a distributed ledger on a peer-to-peer network with a consensus algorithm to ensure replication across nodes. In his book, *Blockchain 101*, Sebastien Meunier describes blockchain as being a "shared trusted registries as immutable source of truth" [15]. In the context of decentralised identity, a blockchain can be the shared trusted source of truth between parties. Guy Zyskind et al. show that the blockchain can indeed be used in a decentralised manner while improving the privacy of its users [16]. Nir Kshetri concludes that "blockchain's decentralised nature is likely to result in a low susceptibility to manipulation and forgery by malicious participants" [17].

On the other hand, Marco Conoscenti et al. have written a paper discussing if the permissionless blockchains and peer-to-peer approaches could play a role in the development of decentralised applications while preserving the privacy of the users [18]. They analyse current uses of blockchain and

its current degree of integrity, anonymity and adaptability. They conclude that the blockchain by itself only guarantees pseudonymity. This means that an identity platform built on top of a blockchain would still need to be built with *privacy by design*.

Roman Beck et al. have shown that a traditional analogue transaction can be performed digitally with the help of blockchain technology in a decentralised manner [19]. They call it "a solution that offers the realisation of distributed trust-free systems, where economic transactions are guaranteed by the underlying blockchain". In their paper, Karl Wüst and Artur Gervai, describe a model that helps determine if a blockchain is required for a particular application [20].

E. Sovrin

Sovrin is an SSI network introduced by the Sovrin Foundation and is based on the open source Hyperledger project [21]. Specifically, the blockchain technology that Sovrin operates on is maintained in the Hyperledger subproject, named Hyperledger Indy. Hyperledger Indy provides the implementation of the distributed ledger used in the Sovrin network. This ledger is explicitly designed to be a public utility which can be used by any person or organisation. It supposedly does this by using blockchain technology to provide consensus.

Evernym, the software company mainly responsible for providing the implementation of Sovrin, has published a paper clarifying what is written in the Sovrin distributed ledger [22]. Although descriptive in technical and functional requirements, the paper does not provide arguments to why a blockchain is required for the operation of the Soverin network. Sovrin has posted a technical paper describing how the flow of the system should work using technical details [23]. It covers all related entities within Sovrin such as DIDs, Keys, Credentials, and Agents.

There are a number of alternatives available for decentralised identity that are also built using blockchain such as SecureKey Verified.Me, Uport, Jolocom and many more. These alternatives, however, all have a slightly different goal, technical implementation and definition of SSI. SecureKey Verified.Me is a product developed by a company called SecureKey in collaboration with IBM [24]. It works on the Hyperledger Fabric project from the Linux Foundation. It is developed for Canadian government services and organisations like banks. Uport is a framework that works on the ethereum blockchain. It allows users to authenticate them selfs and allows users to manage their identities across multiple applications. Uport itself is a DApp and conforms itself to be developed against other DApps. It does this using smart contracts on the ethereum blockchain.

IRMA is another decentralised solution to identity, but it is not blockchain based [25]. It is an attribute-based system. This means it attempts to use attributes, such as "is older than 18?", instead of actual properties, such as date of birth.

III. Research Method

First, the characteristics of credential transactions over the Sovrin network will be investigated. We will start by applying the model as defined by the W3C Verifiable claims working group. This model is then used to analyse three cases. A case in the physical world, the digital domain, and in the digital domain using the Sovrin network. Using these cases we determine the differences between the physical, digital, and Sovrin domains. In order to construct the Sovrin case, existing documentation is used [4] [22] [23] [26]. From there we will analyse characteristics of the Web of Trust and compare it to Sovrin's implementation. To do this, we will mainly use the review of Philip Zimmermann and Jon Callas on PGP's history [27]. In addition, we will primarily focus on the main problems of the traditional Web of Trust (PGP) in order to identify the main additions of Sovrin's implementation. For each problem encountered, we analyse if blockchain technology provides a solution and what the implications are. Finally, we discuss if blockchain technology is a required part of the implementation of Sovrin to solve the identified problems.

IV. Credential Transaction Model

The Verifiable Credentials Data Model definition by the Verifiable Claims Working Group gives a model that defines a few terms and entities [10]. A short list of definitions is given, as well as a depiction of the relationships between roles in figure 1.

Credential Entities consisting of attributes that attest a qualification, competence, or authority issued to an individual.

Inspection The act of comparing the attributes of a credential to the context of the party performing the inspection.

Verification The act of comparing its attributes to the context of a third party.

Issuer The party that supplies or distributes credentials.

Holder The holder of issued credential. The holder is often, but not always, the subject of the claims.

Verifier The party that requests and receives credentials and possibly wants to verify the proposed claims.

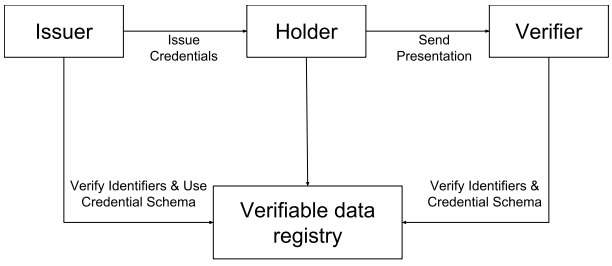

Fig. 1. The diagram displays the relationship of the transferred credentials

Issuing The act of declaring and delivering a credential to the holder.

Presentation The act of presenting the credential and the proof that the credential requested is indeed valid.

Verification The act of comparing its attributes to the context of a third party.

Defining Schema The act of defining the structure of a credential.

Register Identifier The act of registering an unique identifier.

A. Physical Credentials Case

A simple case will be examined, starting with a person named Alice. Alice wants to hire a car at a car rental company. Before Alice can do this, the car rental company requires that Alice provides some evidence that she is able and permitted to drive a car. Alice does this by providing a drivers licence that she received from the Department of Motor Vehicles (DMV). A person at the car rental service, named Victor, inspects and accepts the license, and granting Alice permission to rent a car.

Next, a brief analysis of this case is provided. Both parties have an incentive to start a business transaction. Alice wants to rent a car, and the company wants to conduct business. However, both parties wish to have some assurance about the legitimacy of the other. Not only does the car rental company want the assurance that Alice is able and permitted to drive a car, but they also want to be able to identify her in case of damage claims or other situations. To this end, Victor will need a document or piece of evidence that states the permission to drive and Alice her identity information, which is, in this case, her driver's license. Victor accepts the driver's license as he trusts the authenticity of the provided license; the credential schema is known to him. Also, he trusts the third party issuing the document, in this case, the DMV that issued the document to test for the necessary qualification of her ability to drive. On the other hand, Alice accepts the legitimacy of the car rental company by judging it on certain factors, which is specific for each individual. For Alice, this may be the appearance of the company building and staff, or she may check the Chamber of Commerce for any registration on the companies name.

It is important to note that the person at the car rental company does not verify the authenticity directly with the DMV. This person assumes the authenticity of the license by looking at the attributes of the license, such as the size, the colour, the font used and the displayed photo on the card. It is then assumed that it is valid, without consulting a verifiable data register. In real life, this process is highly dependant on context because the person inspecting the license should be familiar with the format of the license. This means that there is a wide variety in the quality of inspection, as it differs per person.

There is no commercial, contractual, or technical relationship required between the issuer and verifier for the transaction to be able to take place. In the case of Alice, the DMV does not issue driving licenses for people to be able to rent a car. Nevertheless, the rental company is still able to choose to accept licenses issued from the DMV as sufficient proof that Alice is able to drive. This brings us to the next point. The verifier can make its own decision in what kind of evidence they as an entity wish to accept. The rental company may at any moment choose to no longer accept licenses of a specific type if they wish to do so. On the other hand, the holder has control over which credentials to present to the verifier. Alice could choose not to share her license. However, this would most likely result in Alice not being able permitted to rent a car. Furthermore, there is no central authority governing all the credentials within this realm. This means anyone can be an issuer, a holder, and a verifier. However, it is important to note that not every credential holds the same value towards each participant in the system.

B. Digital Credentials Case

In the previous case, the transaction was done in person. However, what if this process were to be conducted over the internet? Most assurances given to the involved parties are no longer available. Another simple case is given to be examined.

The car rental company has a website that Alice can use to rent a car. Alice visits the website and starts the procedure to rent a car. The website asks Alice to identify herself before her request can be fulfilled. This can be done by registering an account and sending in a scan of her license. After Alice does this, she needs to wait for the company to verify the submitted credentials. This can take up three working days. The company now verifies that the received credentials were issued by a trusted third-party and that Alice meets all the requirements. The company accepts the credentials, granting Alice permission to rent a car.

In this case, the company can no longer rely on the presented credentials. The company, as an example, is no longer able to judge if the photo on the license is a photo of the person communicating through the internet. Physical credentials cannot easily be replicated, and the involved parties of a transaction can immediately verify the physical attributes. In the digital world, this is not possible as anything can be copied in the digital world. Nothing is preventing a person from copying credentials many times over and sharing them with multiple people. Impersonation is harder to detect. The company, therefore, needs to verify Alice at the time of picking up the car.

A system could be created where the verifier contacts the issuer directly. However, this would be at the expense of the privacy of the holder as the issuer has the incentive to be able to know which verifier and holder are present in the transaction. In addition, the credential of Alice would be passed on to several communication points, which all pose a risk of potential data breaches. This is even worse as such credentials usually contain more information than is needed for the transaction. Name, address, state or external characteristics are at risk as well.

In the offline case, Victor could rely on attributes such as the size, colour, and quality of the presented credential. These attributes are unavailable or meaningless in the digital domain. The credential schema may not be well known. If it is, there may be no guarantee the credential was not copied or edited as well. In other words, the possibility of proper verification depends on the infrastructure of the domain in question.

C. Digital Sovrin Credentials Case

Alice has a self-sovereign identity setup on the Sovrin network with a list of credentials issued by the DMV. The car rental company has a website that Alice can use to rent a car. Alice visits the website and starts the procedure to rent

a car. The website asks Alice to identify herself using the Sovrin network. She clicks on the connection request. The mobile Sovrin agent app opens and notifies her that a party claiming to be the car rental company wants to connect with Alice. Alice accepts. The car rental company now requests the required credentials from Alice. Alice sees an overview of all the information that she will be sharing with the company and then accepts. The company now verifies that the received credentials were issued by a trusted third-party and that Alice meets all the requirements. The company accepts the credentials, granting Alice permission to rent a car. Alice can now proceed on the website.

Again, both Alice and the company want to engage in a business transaction. Just like in the other case, the company wants to rent out a car to Alice. Alice needs to identify herself and prove that she meets all the requirements. The company presents Alice the option to identify herself through Sovrin which Alice accepts. A relationship is set up between the company and Alice which creates a secure private channel using the DIDs over the Sovrin network. For every relation, a new DID and key pair are generated on both sides. This is called a pairwise-pseudonymous DID pair. It strengthens the privacy of the identity holders as the newly generated DID cannot be correlated over all existing and future relationships the holder may establish.

The company uses this new channel to request the credentials of a specific type. As Alice holds the required credentials, she is able to supply the requested credentials. The company receives the requested credentials and can prove that the presented claims are authentic and issued to Alice using cryptographically sound evidence. It does this by using the cryptographic keys corresponding to the DID that issued the credential. The company also verifies that the credentials have not been revoked, this is done using the proof of non-revocation that the holder included in the presentation of its credentials. This is a zero-knowledge proof that the verifier can confirm by contacting the Sovrin blockchain. The company can verify the presented credentials instead of only inspecting them. At no point is the issuer contacted by the verifier.

To conclude, credentials can be issued to the holder, which can be presented to the verifier without contacting the issuer. The company can validate that it is talking to someone that has the required credentials and that these have not been tampered with. The company can do this without the need for extra verification steps outside of this process. The issuer is not able to track to which verifier the holder presents its credentials to. The transaction of credentials over the Sovrin network has the same characteristics as the case in the physical world case, with improved verification, and mitigating some of the problems from the digital case.

Within the Sovrin network, DIDs are used to establish a secure private channel between two parties. This in itself does not ensure that there is trust between these two parties. This can be compared to self-signed certificates in traditional PKI. To establish trust, the exchange of verifiable credentials over this channel is required. These credentials may be used to determine if the party in question is trustworthy. Sovrin describes itself as a web of trust as trust comes from multiple issuers. There is no root of trust present as with traditional PKI with centralised CAs.

V. Web of trust

In the following section, the characteristics of the Web of Trust will be analysed. This is done to determine what the main problems are with the Web of Trust, and especially those that Sovrin tries to solve. As stated in section III, this analysis is mainly based on Philip Zimmermann and Jon Callas review on PGP's history [27]. First, a brief description of the Web of Trust as implemented by PGP is given. Then, the identified problems are analysed more closely. Finally, an overview of the problems found is presented.

PGP became an acronym for many different meanings, including organisations, companies, protocols and software. In this study, we will mainly focus on the protocol aspects of PGP. PGP functions as follows: every user in the network generates a cryptographic key pair. This key pair is used to encrypt and sign messages sent to other users. The encryption is used to obfuscate the message to other users, except for the recipient. Signing is necessary to prove that the message came from the sender. In short, to send and receive messages securely one needs to possess the private key of the sender and the public key of the receiver. To prove a certain message came from the sender, the receiver needs the public key of the sender.

The public key in PGP acts as an identifier. This means that if the corresponding private key were to be lost, the identity associated with this identifier would also be lost.

Another problem arises when a key pair needs to be revoked, for example when the private key was lost. It is possible to sign a revocation certificate, however, broadcasting the revocation is still an issue due to uncontrolled distribution. In other words, it is not guaranteed that every participant receives the revocation certificate. Due to the decentralised nature of the network, it is not desirable to distribute the certificate to every PGP user. Instead, key servers are used to distribute the certificates as well. This introduces the problem that the user is dependant on the behaviour of the key servers in order to get the revocation across to other users.

The Web of Trust may be represented as a graph. The graph contains all users of PGP including existing relationships regarding trust. For example, one may have existing trust in friends, but not in friends of a friend. In order to establish a connection with an unknown person, the judgement of known users may be used. That is, if there is a direct relationship to a given user, this user might be able to provide a path to the user in question. Such a relationship is visualised in figure 3.

Essentially, a chain of trust is established. Every node in this chain introduces new risk, degrading the certainty of the public key belonging to the person at the end of the chain. To establish such a chain, every step forward needs to be verified in person. That is, every user in the chain needs to meet the next person in the chain and verify their public key. This makes it very costly to create long chains, and often it is impossible to find a valid chain as well.

In other words, managing all relationships within large organisations or companies can quickly become very complex.

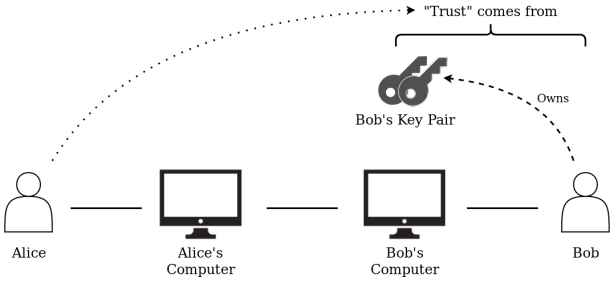

Fig. 2. The definition of trust in a Web of Trust. Depending on Bob's ability to manage his key pair.

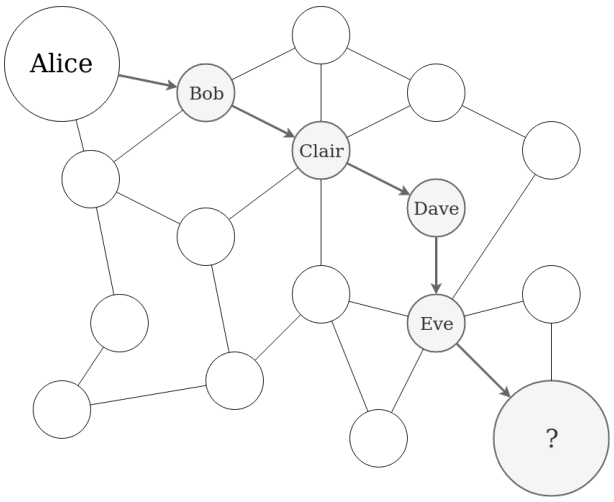

Fig. 3. The *Chain of Trust* depicting the relationship between Alice and a acquaintance of Eve, including all actors in between.

In order to solve this problem, PGP 5.0 [27] introduced the concept of *delegated trust*. This concept entails the delegation of trust from a certain *trust anchor* which actors in the Web of Trust can rely on. For example, the board of a given company may be delegated trust by the chairman of the board. From there, depending on the depth the keys were signed with, the board members themselves may delegate this trust further to other employees of the company. This is done by creating a signature of the public key of a given person, which may then be used by others to verify trust in this signed key. One of the problems with this concept is the propagation of the trust keys. Delegated trust is a hierarchical, centralised structure. Users may rely on multiple delegated trust sources, as is common in real life. Another example of a source of trust is the government, which citizens may choose to trust. In reality, there may be a multitude of trust anchors which overlap in the same Web of Trust.

Each of these trust anchors needs to propagate their signed keys to the users that wish to use them. This can be done via email, but that would cause a lot of congestion. For example, in a domain with n users, the upper bound of the amount of emails e needed to be sent throughout the domain would be equal to the following equation:

$$e = n \cdot (n-1) \cdot (n-2)$$

This upper bound may be reached if each user in the domain signed the key of each other user into a domain, and then sent those signed keys to all remaining users. In addition, it is undefined how every user should behave when they receive a key. For the key to propagate throughout the network, all users should be inclined to reach the same goal.

In order to mitigate this, the keys may be provided by *key servers*. Although trust anchors may choose to host their own key server, it is more common to have independent parties tending to this problem. Keys may then be uploaded to this server. In either case, users are dependant on the parties maintaining the key servers. Ideally, the key servers should collaborate on keeping their key stores consistent, as this would enable users to contact each other regardless of the key server used. However, this is not necessarily the case; the key server parties are free to maintain their own policy. This would thus separate the global PGP domain into multiple smaller domains, one for each (collaboration of) key server(s).

Extra problems arise when using public key servers. It is difficult to remove keys from key servers, as one's identity should not be denied. If a given key was to be deleted, collision errors might arise when the key was newly generated and uploaded for another person. Also, as any party can upload keys, bogus keys may be introduced to the keystore. For example, a key may be generated and uploaded by someone pretending to be someone else, primarily if the victim does not use PGP in the first place. Lastly, users may be subjected to a decrease in privacy. Public keys and their signed counterparts are available on the key server for anyone to see. This means that the signing relationships can be correlated over time.

Certain parties proposed solutions to some of these issues. For example, the *PGP Global Directory* introduced a global key server that verifies all keys per email for initial uploads as well as periodically. Although these initiatives have apparent drawbacks individually, a common problem that remains is that they are all centralised. Users are dependent on the individual key servers without guaranteed authenticity of the keys or collaboration between other key servers. In other words, users might not be able to establish a direct connection in the Web of Trust.

To conclude, we have identified the following issues:

- The attachment of identity to key
- The synchronisation of key states
- Graph traversal distance problem
- Centralisation of multiple components

This becomes more problematic when introducing the concept of credentials in a decentralised system. First of all the problem of authentication is still present. With credentials, this boils down to the verifier not being able to consistently verify whether credentials came from the issuer that claims to have issued them. If this is established, the verifier may also be able to validate the format of the credential, as the party where it came from is then identified. It is important to note however that this relationship between the verifier and issuer does not

necessarily need to be trusted. Verifiers may verify credentials as authentic, but they do not need to be valuable to the verifier. Lastly, credentials typically do not last forever; they have an expiry date. Besides, credentials may be revoked before this expiry date in certain cases. This is hard to verify without a centralised source.

VI. Sovrin's Web of Trust

In this section, we analyse the problems identified in section V and compare them to the solutions offered by Sovrin. We begin with the problem of centralisation and then move on to, among others, the synchronisation of state problem, the attachment of IDs to keys and the graph traversal problem.

PGP has the problem that the introduction of key servers centralises multiple parts of it's infrastructure. This makes its users dependant on a single organisation. Sovrin can communicate peer-to-peer between parties and store state on a distributed ledger. In Sovrin's case, state is 1) an identifier (DID) pointing to a DID document, 2) registries per DID that authorize other DIDs to delegate control to, 3) Credential schema's and definitions used by issuers to check validation of the credential format, and 4) registries per issuer that prove the non-revocation proofs when presenting credentials [26] [22]. All parts require that every user may access them in order to solve the associated problems. In this chapter, we will discuss these problems more closely.

The distributed ledger can function as a consistent state store for all clients while still being decentralised. The ledger acts as a single source of truth. This solves the problem in PGP with the synchronisation of key states between key servers. This way, every user can guarantee its existence with the blockchain. An example is when two users connect and roll over their key simultaneously. When this happens, the connection has to be reset in an out of band process. However, if one of the users is listed on the distributed ledger, the other user can look up the associated DID, DID document and new public key. Another example is for verification. In this case, the verifier needs to know if the issuer of a given credential is indeed the actual issuer. As all issuers are to be recorded on the ledger, this is a simple lookup.

Furthermore, the distributed ledger allows the detachment of identity and cryptographic key material. The identity of a subject within Sovrin is built up by the collection of credentials with the unique pairwise-pseudonymous DIDs. Not just the ownership over a key. Revocating an agent and its keys do not lockout a subject out of its identity domain as the subject may have multiple keys and agents. The ledgers policy registry facilitates the relationship between the different agents that are authorised to act on the identity holders behalf. This also removed the construction of trust from the traditional Web of Trust, where trust solely depended on the key pair. In this case, trust comes from the combination of keys. This derivation of trust is depicted in figure 4.

Another problem in the PGP system is that it is not always possible to create a path to any given user. Previously this was mitigated by introducing centralised key servers. In order to mitigate this problem, Sovrin introduces a solution that

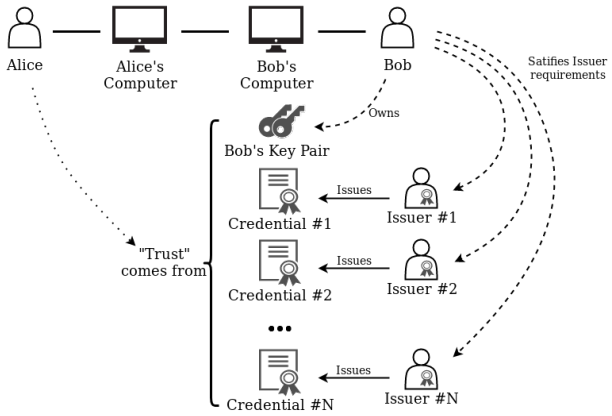

Fig. 4. The diagram displays the trust relationship within the DKMS protocol. The trust comes from the credentials that Bob presents to Alice

reduces the traversal distance needed to establish a valid path. This is done by introducing a single source of truth which is the distributed ledger. Again, the main concern here is the decentralisation of the ledger. It allows verifiers to quickly verify the authenticity of credentials to a datastore in a fixed traversal distance. In addition, credentials allow for the reduction of this path as well. Issuers now perform a similar function as trust anchors in the PGP system, which is to provide shortcuts to other users. Instead of signing keys, credentials are issued which may be verified independently. However, whether a given credential is valuable to a verifier varies. Sovrin does not solve this problem of establishing trust between the verifier and the issuer.

In conclusion, Sovrin is able to solve or mitigate most of the problems identified in section V.

VII. Discussion

First of all, it should be noted that both the standard for Verifiable Credentials by the W3C as well as the Sovrin protocols are still under daily development. Therefore, a lot of material is subject to change. In addition, this is reflected in the documentation provided by Sovrin, as some sources hosted at *Sovrin.org* seem to be outdated. The definition of SSI is also subject to change in the Sovrin project.

We find that Sovrin can supply credentials in a decentralised manner alike to what happens in the physical world. However, this was done using a single case, but the major part of Self-Sovereign Identities is that every party can dictate its own rules and weights. In other words, the received credentials should be weighed to its context. The analysis of the relationship between multiple contexts and weights are therefore left out of this research.

Sovrins distributed ledger is decentralised as multiple entities participate as validator nodes. However, this ledger is permissioned, meaning only invited parties can participate as a node. The argument that this is a degradation of decentralisation can be made. In addition, as Sovrin still resides in its bootstrapping phase, the Sovrin Foundation has privileged control over the ledger. This is a temporary case of centralisation in the network.

VIII. Conclusion

Using the model and cases from section IV, we conclude that the Sovrin network is able to implement a system that is user-centric and follows the principles proposed by Sovrin's self-sovereign identity model. The Verifiable Credentials, as implemented by Sovrin, allows users to identify themselves in a decentralised manner as well as a privacy-preserving manner. Sovrin's transaction flow of credentials incorporates many characteristics that are also present in the physical world. On top of that, Sovrin allows for better verification processes and real-time revocations compared to the physical world.

The blockchain in the Sovrin network is a manner to operate the network in a decentralised manner while providing a single source of truth. This source of truth is a datastore that facilitates, among others, the separation of identity and a single key. These characteristics of the blockchain technology allow Sovrin to implement certain features to combat the problems as identified in section V.

In conclusion, blockchain technology is required to achieve the combination of features that Sovrin implements in its network. There might be viable solutions to the problems individually, however we argue that the same principles apply. That is, The source of truth should be decentralised which means that there is consensus over this shared source of truth. This can be done in multiple ways, one of which is blockchain technology.

A. *Future work*

Only one type of case was discussed in this paper, we would like to see more cases and multiple discussions. It will be interesting to see if other unusual cases may be analysed that introduce significant problems with the current infrastructure.

Also, it will be interesting to see how the relationships between verifiers and issuers will be built up and maintained, as Sovrin does not provide a solution to this problem as of yet. The role of agent providers, or agencies, will be interesting as well. If implemented and used correctly, they act as a centralised but obfuscated agent service provider. These agencies will undoubtedly create a new market for such applications. If this will introduce the problem of centralisation is an interesting topic for further research.

What may be interesting as well is to test the resilience of the Sovrin network in terms of speed and capacity, as well as any security implications. Also, it will also be interesting to see what happens in the case of a service outage. For example, whether the verification will come to a halt when the revocation registries are unavailable. Another subject of interest is the scaling of the network in terms of State keeping. More precisely, how will the distributed ledger be kept under control without compromising people's identities?

Lastly, the rollover of control from the Sovrin Foundation to the network itself is a fascinating subject, as it is undefined when or how this will happen. Besides, it will be interesting to see how the network will function without a governing body. For example, what happens if there exists a dispute between two large parties in the network. Will the network be able to be partitioned in the future? Although, most of these last questions may be answered after substantial adoption of the network.

Acknowledgement

A special thanks to our supervisors Dr. Ir. Oskar Deventer and Rieks Joosten for guiding us during this project.

References

[1] A. Jøsang and S. Pope, "User centric identity management," in *AusCERT Asia Pacific Information Technology Security Conference*. Citeseer, 2005, p. 77.
[2] A. Christopher, "The path to self-sovereign identity," https://github.com/ChristopherA/self-sovereign-identity/blob/master/ThePathToSelf-SovereignIdentity.md, 2017.
[3] R. Joosten, "Self-sovereign identities: It is going to happen!; may 2018 — tno," 2018. [Online]. Available: https://blockchain.tno.nl/blog/self-sovereign-identities-it-is-going-to-happen/
[4] The Sovrin Foundation, "Sovrin: A protocol and token for selfsovereign identity and decentralized trust," 2018. [Online]. Available: https://sovrin.org/wp-content/uploads/2018/03/Sovrin-Protocol-and-Token-White-Paper.pdf
[5] K. Cameron, "The laws of identity," *Microsoft Corp*, 2005.
[6] ——, "A user-centric identity metasystem," *Microsoft Corp*, 2008.
[7] P. Windley, "Multi-source and self-sovereign identity," 2018. [Online]. Available: http://www.windley.com/archives/2018/09/multi-source_and_self-sovereign_identity.shtml
[8] ——, "Multi-source identity," 2018. [Online]. Available: http://www.windley.com/archives/2018/05/multi-source_identity.shtml
[9] W3C Verifiable Credentials working group, "Verifiable credentials working group faq." [Online]. Available: http://w3c.github.io/webpayments-ig/VCTF/charter/faq.html
[10] ——, "Verifiable credentials data model." [Online]. Available: https://w3c.github.io/vc-data-model
[11] ——, "Decentralized identifiers (dids)." [Online]. Available: https://w3c-ccg.github.io/did-spec
[12] The Linux Foundation, "Hyperledger indy sdk." [Online]. Available: urlhttps://github.com/hyperledger/indy-sdk/blob/677a0439487a1b7ce64c2e62671ed3e0079cc11f/doc/design/005-dkms/DKMSDesignandArchitectureV3.md
[13] C. Allen *et al.*, "Decentralized public key infrastructure," *WebOfTrustInfo*, 2015.
[14] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
[15] S. Meunier, "Blockchain 101: What is blockchain and how does this revolutionary technology work?" in *Transforming Climate Finance and Green Investment with Blockchains*. Elsevier, 2018, pp. 23–34.
[16] G. Zyskind, O. Nathan *et al.*, "Decentralizing privacy: Using blockchain to protect personal data," in *Security and Privacy Workshops (SPW), 2015 IEEE*. IEEE, 2015, pp. 180–184.
[17] N. Kshetri, "Can blockchain strengthen the internet of things?" *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.
[18] M. Conoscenti, A. Vetro, and J. C. De Martin, "Blockchain for the internet of things: A systematic literature review," in *Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of*. IEEE, 2016, pp. 1–6.
[19] R. Beck, J. S. Czepluch, N. Lollike, and S. Malone, "Blockchain-the gateway to trust-free cryptographic transactions." in *ECIS*, 2016, p. ResearchPaper153.
[20] K. Wüst and A. Gervais, "Do you need a blockchain?" in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018, pp. 45–54.
[21] S. Techonologies, "Hyperledger indy." [Online]. Available: https://www.hyperledger.org/projects/hyperledger-indy
[22] A. Tobin, "Sovrin: What goes on the ledger?" 2017.
[23] D. Hardman, "How dids, keys, credentials, and agents work in sovrin," 2018.
[24] "Your identityin your control." [Online]. Available: https://verified.me/
[25] "Irma." [Online]. Available: https://privacybydesign.foundation/irma-en/
[26] D. Hardman, *How DIDs, Keys, Credentials, and Agents Work in Sovrin*. The Sovrin Foundation, 2018. [Online]. Available: https://sovrin.org/wp-content/uploads/2019/01/How-DIDs-Keys-Credentials-and-Agents-Work-Together-in-Sovrin-131118.pdf
[27] J. Callas and P. Zimmermann, *The PGP Paradigm*. Github, 2015.