Data & Policy (2023), 5: e38
doi:10.1017/dap.2023.41

RESEARCH ARTICLE

A comparative cyber risk analysis between federated and self-sovereign identity management systems

Anhtuan Le, Gregory Epiphaniou and Carsten Maple

Warwick Manufacturing Group, University of Warwick, Coventry, UK
Corresponding author: Anhtuan Le; Email: a.le.1@warwick.ac.uk

Received: 17 December 2021; **Revised:** 18 October 2023; **Accepted:** 13 November 2023

Keywords: attack surface; FIDM; identity management; risk analysis; SSI

Abbreviations: DKMS, decentralized key management system; DLT, distributed ledger technology; FIDM, federated identity management; IDM, identity management; SLA, service level agreement; SSI, self-sovereign identity

Abstract

Self-sovereign identity (SSI) is an emerging and promising concept that enables users to control their identity while enhancing security and privacy compared to other identity management (IDM) approaches. Despite the recent advancements in SSI technologies, federated identity management (FIDM) systems continue to dominate the IDM market. Selecting an IDM to implement for a specific application is a complex task that requires a thorough understanding of the potential external cyber risks. However, existing research scarcely compares SSI and FIDM from the perspective of these external threats. In response to this gap, our article provides an attack surface analysis focused solely on external threats for both systems. This analysis can serve as a reference to compare the relevant security and privacy risks associated with these external threats. The threat landscapes of external attackers were systematically synthesized from the main components and functionalities of the common standards and designs. We further present a use case analysis that applies this attack surface analysis to compare the external cyber risks of the two systems in detail when managing cross-border identity between European countries. This work can be particularly useful for considering a more secure design for future IDM applications, taking into account the landscape of external threats.

Policy Significance Statement

The design of any identity management (IDM) system will involve considering cyber risks with each of the deployment options. However, there lacks literature comparing risks between different available IDM approaches. This article provides insights into the attack surfaces of the two most widely used IDM systems, namely federated identity management (FIDM) and self-sovereign identity (SSI), specifically by covering their reference architecture with underlying components and analyzing the respective threats. This knowledge is essential to understand the relevant risks of each approach. Using these insights, we demonstrate a use case to compare the risks of FIDM and SSI when managing cross-border European ID schemes. Our work will help policymakers to understand the risk basis to select or design the appropriate solution for their future needs.

1. Introduction

The development of new Internet technologies facilitates the expansion of information systems by reducing the barriers to linking different business applications within and across corporate boundaries.

In addition, the demands on digital identities are increasing in modern societies as people have spent a significant part of their lives participating in online services and activities. In this context, identity management (IDM) has become a strategic need for contemporary life and requires solutions that can overcome technical, political, and social barriers (Ayed, 2014).

Over the past few decades, the literature has seen IDM systems evolve in four main phases. At first, the vast majority of Internet identities (e.g., emails, accounts) are centralized, in which identities are owned and managed by a single organization, such as an e-Commerce website or a social network (Tobin and Drummond, 2016). Users who have registered with numerous online services would have to keep passwords on all servers for authentication, which means that authentication data are duplicated and hidden in many places (Lim et al., 2018). Such repetitions can easily attract threats to the authentication process, while server vulnerabilities have led to theft and privacy breaches in user identities (Lim et al., 2018). To address this, the federated identity management (FIDM) approach provides a centralized identity system with greater mobility by allowing users to use different services with the same credentials (Tobin and Drummond, 2016). FIDM can also enable various services to exchange information about user identity. Both centralized and FIDM are server-centric, in which power remains with the identity provider (IdP). The server side has access to the subscriber information for authentication purposes, but this raises many privacy concerns. The lack of transparency makes it difficult for users to ensure that the relevant service level agreement (SLA) standards are met (Lim et al., 2018). There are some significant examples of data breach by service providers, such as the recent dispute between Facebook and Cambridge Data Analytica over the alleged collection and use of personal data that could have impacted the 2016 US presidential election; and the Brexit vote on the UK referendum on leaving the European Union (Satybaldy et al., 2020). During the past decade, the IDM paradigm has changed from server-centric to user-centric, giving users control over their own identities and providing them with a clear presentation and intuitive assessment of privacy (Pöhn and Wolfgang, 2020). The latest sophisticated self-sovereign identity (SSI) management systems enable identity holders to transmit the identity and verified attributes of individuals, organizations, and objects with full authority and consent (Tobin and Drummond, 2016). The core concept of SSI is verifiable credentials (VCs), which shift the utility and portability of physical identity credentials to digital devices (Preukschat and Drummond, 2021). VCs enable tamper-proof SSI by allowing assertions to be validated by any verifier. A key distinction between FIDM and SSI revolves around the scope of assertions made about identity subjects. In FIDM, assertions are strictly confined to the user or subject in request, directly tying their identity attributes to them. Conversely, in SSI, the responsibility is on the holder who possesses the wallet. This unique configuration in SSI allows for a broader range of assertions, encompassing not only the holder's own identity attributes but also potentially extending to any other user or subject. This flexibility in SSI underscores its capacity to facilitate a more diverse set of identity assertions compared to FIDM.

Security is important in IDM as identification opens many doors to access numerous services on behalf of the owners. Private user data must be protected from theft and fraud, while security must be robust enough to ensure that only authorized access is granted. However, IDM systems are vulnerable to a wide variety of cyber-attacks and security issues have sparked a great deal of research interest. There are several works dealing with threat modeling and attack surface analysis for FIDM and SSI systems. Simpson (2016) conducted a systematic FIDM security analysis survey, which categorizes security incidents that occur in FIM protocols to specify the FIDM problem landscape. Aldosary and Norah (2021) provided a comparison between FIdM architectures such as liberty alliance, security assertion markup language SAML v2.0, WS-Federation, Shibboleth, and so forth to summarize the FIDM limitations based on how it affects the user. In Simpson (2016), the author not only reviews a comprehensive attack surface of attacks in FIDM, but also models the escalation of attacks, that is, how attacks on one stakeholder can cause possible attacks on other stakeholders. There is less significant work on SSI threat modeling, security analysis, and risk assessment. Naik et al. (2022) combine attack tree model and the risk matrix model to perform evaluations of possible attacks and their security risks targeting SSI. The research employs a systematic approach that includes specifying the system architecture and assets to assess potential attacks and their security risks. Kim et al. (2021) address the implementation aspects of SSI by conducting

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

security analysis of a blockchain-based DID services that enable SSI. The authors analyze the data flow between DID system components, as well as functional domains, to justify potential security threats. While there is extensive work on addressing attacks and risks in SSI or FIDM systems, there are no noteworthy research studies that focus on comparisons between them. To the best of our knowledge, this study is one of the first to conduct risk comparisons between the two systems with a use case analysis to bridge the gap in evaluating the risks of different IDM models to use for specific purposes. Such work can later also support the designs of new IDM systems to improve their cybersecurity and privacy. Our main contributions to this work are as follows:

1. We synthesize the essential components and functionalities in each IDM system that can be the primary targets for security attacks.
2. We review and identify the typical attacks (i.e., attack surface) toward different IDM systems based on their essential components and functionalities.
3. We compare cyber security risks between FIDM and SSI and examine the insights by analyzing a use case that focuses on the two specific IDM designs (i.e., FIDM and SSI) in cross-border IDM applications.

The structure of this article is as follows: Sections 2 and 3 specify the main components and functionalities of the FIDM and SSI systems along with their respective threat landscapes. Section 4 uses the results of Sections 2 and 3 to compare cyber risks between the two specific IDM designs for cross-border IDM, and Section 5 concludes the article.

2. Federated IDM

This section examines the general components and functionalities of the FIDM architecture before summarizing their common threats.

2.1. FIDM reference architecture

According to Aldosary and Norah (2021), the four main FIDM components include:

1. A *user* is a person who acquires a specific digital identity to interact with some services.
2. The *user agent* or *user interface* is a software application or browser that allows users to interact with the services they require.
3. The *service provider* (SP) site is an entity that offloads authentication to a third party. SP can also be called *relying party* (RP) as it relies on external identity authorization entities (i.e., IdP) to decide access to its services.
4. The *identity provider* (IdP) is an entity that identifies users that later enables SPs to authorize user accesses based on their identities.

These four components interact differently in each FIDM standard and implementation, including but not limited to Liberty Alliance, Shibboleth, WS (Web Service) Federation Architecture, SAML (Security Assertion Markup Language), OIDC (OpenID Connect), and OAuth (Open Authentication). Moreover, some standards have modules that deliver the same functionality, but have a different name. Therefore, it is more feasible to find a high-level architecture with basic functionalities rather than to capture all the details in a broad model. Such an architecture can be found in Cabarcos (2013), which is illustrated in Figure 1 with the functionalities described below:

1. *Circle of trust (CoT) configuration*: Both SPs and IdPs provide a CoT configuration component on which the services depend. This component is responsible for accessing local data stores to verify that a provider involved in a current identity-related transaction can be trusted. The selection is

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

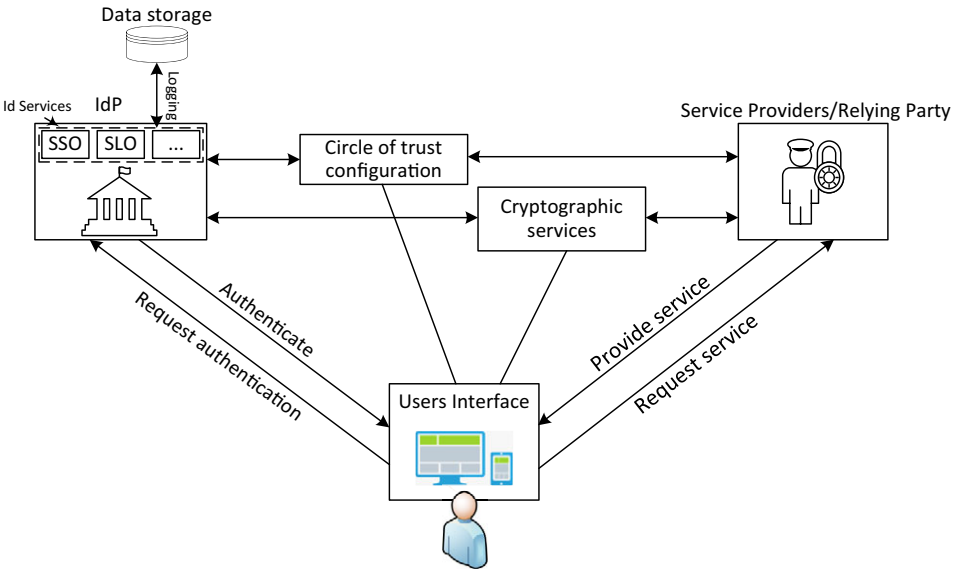

Figure 1. *A general architecture for FIDM (based on Cabarcos, 2013).*

essentially made by contacting the local data store to verify that the entity is on a list of trusted entities and, if necessary, that there is an explicit SLA between them.

2. *Identity services*: This component comprises the services offered by the identity framework, that is, Single Sign-On, Single Log Out, Authentication, Authorization. SP and IdP use this component to exchange user identity data.
3. *Cryptographic services*: This component provides cryptographic support for the required security processing such as encryption/decryption of assertions, signing/validation, and so forth.
4. *Logging*: Vendors typically implement a logging module to monitor user and service activity. Identity services use the registers, but an interface for auditors (external parties) can also be provided.
5. *Data storage*: contains information such as metadata documents, policies, SLAs, trust data, credentials, logs, session data, messages, and so forth.

FIDM standards and implementations differ from the methods or protocols used for authentication and authorization and maintaining the sessions, such as using access tokens (Liberty Alliance) or issuing access tokens which grant access to the resource (Oauth 2.0).

2.2. Federated IDM threat landscape

In general, attackers can exploit vulnerabilities in the user interface to attack users, RP, and IdP. User attacks involve breaching user access rights, manipulating user session, or information leakage. In this section, we synthesize the 23 potential threats in FIDM, which are divided into seven groups as follows.

2.2.1. Authentication/Authorization attacks

These attacks target the token or authorization code to gain access rights of users, including:

A1. Code/Token/State leak attack

When users click links or resources on the attacker's website, their browsers submit certain requests to the attackers' page, including status and code. Attackers can exploit these exposed states to launch a Cross-Site Request Forgery (CSRF) attack against the victim. For example, it can redirect the victim's

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

browser to the RP endpoints and override the previously completed authorization. The consequences of this attack are that an attacker could cause a browser to log on to an RP under the attacker's identity or force an RP to use a resource belonging to the attacker instead of a resource belonging to the user (Fett et al., 2016).

A2. Token spoofing

This threat arises from inadequate protection of honest ID and access tokens and/or from insufficient validation of ID tokens at the target RP. These issues during validation allow an adversary to access resources in the RP that represent an honest ID token (provided by the benign IdP) that may have previously expired. Navas and Marta (2019) presented five patterns of token spoofing, including stolen token, sniffed token, compromised device, phished token, and honey-RP.

A3. Replay of authorization code

The threat comes from attackers who can obtain the ID and access tokens after capturing the authorization code. This threat exists with authorization code flows due to improper validation of the token request at IdP and/or RP spoofing. There are four common patterns in this category, including stolen code, sniffed code, leaked code via headers, and compromised device (Navas and Marta, 2019).

A4. Manufacturing fake token

Incorrect or insufficient validation of ID tokens enables an attacker to access resources in the RP and supply fake ID tokens that are tailored to their goals. Fake tokens can be created by changing certain characteristics of a legal token (crafted token) issued by a legitimate IdP or they can be created from scratch by a malicious IdP (Navas and Marta, 2019).

A5. XSS (Cross-Site Scripting) attack

In typical XSS attacks, the attacker uses social engineering techniques to entice the victim to click on a malicious link. In SAML attacks, exploiting the vulnerability of incorrect implementation of the SAML framework makes it easy to systematically capture a user (Naik and Paul, 2017). XSS attacks can also happen in OIDC, for example, an attacker can exploit an automated authorization feature that automatically generates an authorization response when a user has an existing session with the provider and has previously given permission for the same client/relying party. The attacker could potentially steal a user access token by exploiting an XSS vulnerability in the client browser (Naik et al., 2017).

A6. Third-party resource attacks (malware)

RPs and IdPs that contain third-party resources could expose their users to token theft and other attacks from malicious programs in those resources (such as tracking or promotional scripts). RPs should avoid adding third-party resources to web resources served from the same origins as the OIDC endpoints. In newer browsers, the integrity of subresources can help minimize the dangers associated with embedding such resources (e.g., rejecting third-party content if it does not match a particular hash).

2.2.2. Session attacks

As there can be many active sessions in a federated identification system, an attacker can access all authorized services from multiple SPs by hijacking just one current session. The attacks include:

A7. Redirect attack (307 redirect)

Attackers can run a malicious RP to obtain users' credentials when they sign in to an IdP that is using the wrong HTTP redirect status code (Fett et al., 2016). This can happen if the IdP used to log in dials HTTP status code 307 when redirecting users' browser back to the RP, and the IdP redirects users immediately after they enter their credentials (i.e., in the response to the HTTP POST request with the form sent by users' browser). The status code 307 causes the users' browser to send a POST request to the RP that contains all the form data from the previous request, including users' credentials. Since attackers own the malicious RP, they can use these credentials to impersonate victims.

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

A8. Naïve RP session integrity attack

When an RP uses naive user intention tracking, attackers could initiate a session using an honest IdP to acquire an authorization code or access token for their own account (Fett et al., 2016). The next time a user tries to log into an RP with an attacker-controlled IdP, that IdP will redirect the user back to the redirect URI of the honest IdP. The attacked IdP adds to this redirect URI the status specified by the RP and the code or token that the honest IdP received. Since the RP is now conducting naive monitoring of user intent, it assumes that the user has logged in to the honest IdP. Therefore, the user logs into the RP with the identity of the attacker in the honest IdP, while believing that these resources are controlled by the user.

A9. Flow interception

This threat is created by the adversary's ability to intercept some of the IAAA (identification, authentication, authorization, and accounting) flows masquerading as the real RP (e.g., changing the redirect URI) or the valid IdP. It can work with both implicit and Authorization Code flows (Navas and Marta, 2019), such as follows:

1. *Network access*: Attackers use Man-in-the-Middle (MitM) attack to redirect the IdP's URI responses sent through the end user's browser. If parameters in this redirect are changed at a certain point in a flow but are not properly processed by the IdP, attackers can obtain information such as authorization codes or ID tokens.
2. *Web access*: Initially, attackers trick the end user into initiating an IAAA flow through a malicious website and submitting a modified authentication request. If the IdP does not properly verify the redirection, the real ID token or authorization code will be transmitted to the attacker.
3. *IdP hot swapping*: The adversary controls a malicious IdP and the end user initiates an IAAA flow following a malicious link. If the targeted RP has the same client ID at the genuine IdP, during an Authorization Code flow, the adversary can intercept the legal Authorization Code. More details of this attack can be found in Mainka et al. (2017).

A10. Session handling

Sessions are usually identified by a nonce that is stored as a cookie in the user's browser. Cookies should use the secure attribute (i.e., the cookie is only ever used over HTTPS connections) and the HttpOnly. If RP does not update the user's session ID with a newly selected nonce after logging in, attackers could create a login session cookie that is linked to a known state value in the user's browser and trick the user into logging in the associated RP (Fett et al., 2017). Attackers could then use the session cookie to access the user's data at the RP.

A11. Injection attack

Cross-Site Scripting (XSS) and SQL injection attacks against RPs or IdPs can lead to the theft of access tokens, ID tokens, and authorization codes (Fett et al., 2017). For example, XSS attacks could give an attacker access to session IDs. A malicious IdP can attempt to inject user attributes containing harmful JavaScript into the RP. If the RP displays these data without performing the appropriate escape, the JavaScript is run. Attackers can also attempt to inject new parameters into URIs by adding them to existing parameter values.

A12. Man in the middle attack

MitM attacks are feasible in many FIDM standards, such as SAML, OIDC, or Shibboleth. For SAML, MitM attackers can replay the RelayState token to pass information about user actions at the SP to the IdP. Attackers can also use Domain Name System (DNS) spoofing attacks to pretend to be one party to deliver MitM attacks (Groß, 2003). Another method is to rewrite the HTTP response that starts the redirect and change the destination URL (Groß, 2003). For OIDC, attackers could mislead a RP into choosing an appropriate IdP at the beginning of the login or authorization process to acquire an authentication code (Naik and Paul, 2017). For Shibboleth, if the assertion is transmitted with the HTTP artifact binding and the HTTP redirect binding is used to transport the artifact, the artifact could leak to the attacker if he gains

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

access to the browser after it is closed because the artifact has been saved in browsing history (Ghasemisharif et al., 2018).

2.2.3. IdP attacks

These attacks specifically target IdP, including

A13. IdP mix-up attack

An honest RP is confused about which IdP is used in a login process. The honest RP thinks that the login uses the attacker's IdP and communicates with this IdP, while the user's browser interacts with an honest IdP and sends the data received from this IdP to the RP (Fett et al., 2017). As a result, the attacker learns information such as authorization codes and access tokens that he should not know and that enable him to violate the authentication and authorization features.

A14. CSRF attacks and third-party login initiation

In the OIDC core standard, a so-called login initiation endpoint is defined, which enables a third party to initiate a login by sending a user to this endpoint. This endpoint is essentially a deliberate bypass of the CSRF protection. Therefore, in addition to the protection offered by the state parameter, further protection against CSRF is required at the endpoint.

A15. Server-side request forgery (SSRF)

SSRF attacks can occur when attackers instruct a server to send requests to other servers, causing undesirable side effects or disclosing information (Fett et al., 2017). OIDC defines a method to indirectly provide parameters for the authorization request. SSRF attacks can be used by attackers to contact services or scan the internal network, which means that attackers can simply launch an SSRF attack against IdP even without OIDC extensions. For example, he can include any URI in an authorization request that requests the IdP to contact this URI. Hence, both RPs and IdPs can be susceptible to SSRF.

A16. IdP account compromise

If an IdP account is compromised, the attackers can pretend to be this IdP to compromise all RPs that support it (Mainka et al., 2017). This attack can be done by:

1. *Compromised IdP password*: Attackers can use various methods (e.g., compromising physical access, network access and/or online access, social engineering, phishing, malware) to obtain the victim's password from the IdP. As soon as the attackers learn the victim's password in the IdP, they can communicate with any RP who supports this IdP as a victim. When being asked to authenticate/consent, the attackers will forge the victim's identity with their password and continue the IAAA flow by impersonating them.
2. *Session hijacking*: Instead of using the victim's password, attackers can hijack their IdP session. If the RP only checks whether the end user is already logged in via the session cookie, the attackers can pretend to be victims by simply loading the acquired session cookie into their browser.

2.2.4. Information disclosure

This attack can be used to expose both user private information and security-related information such as passwords or keys. It can affect a wide range of components such as user access rights, session, user information, logging, or storage.

A17. Snooping

Attackers can collect sensitive user information through advanced monitoring methods such as keystroke monitoring (Malik et al., 2015).

A18. Network eavesdropping attack

If identity data are transmitted over insecure channels, attackers can eavesdrop on identification information by intercepting data packets over the targeted network connections.

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

2.2.5. DoS attack

A19. DoS attack

Attackers can flood IdP or RP with requests from compromised users to cause significant processing overhead on the server side or disrupt the services to the users. Some examples of DoS attacks in FIDM are exploiting the SP-Initiated SSO (Redirect/POST Bindings) message flow in SAML or exploiting dynamic discovery queries in OIDC (Naik et al., 2017).

2.2.6. Privilege elevation attack

A20. Elevation of privileges attack

Attackers can illegally elevate their access rights by mimicking more privileged users to access illegal services (Mohamed et al., 2019).

2.2.7. Privacy threats

IdP can collect end-user attributes during the initial OpenID Connect registration phase, while RP can use IdP access tokens to get user data. However, end users have no control over their PII, that is, they have no method of controlling the use and storage of their attributes which were requested by IdP or RP. Therefore, the following privacy threats are possible (Mainka et al., 2017):

A21. PII leakage

If sensitive information is sent over insecure communication channels (e.g., without proper encryption), an attacker with network access can use a sniffer tool to extract identity information (name, surname, email, phone number, credit card details, etc.). In the case of opponents with web-only access, PII approved by an IdP or RP can be obtained or purchased via the Internet or the Dark Web. This threat can take effect with both privileged and nonprivileged access.

A22. User profiling

This happens when attackers can monitor the behavior of end users and combine this activity across different apps, services, or resources to create user profiles (e.g., behavior, habits, hobbies, schedule). Dynamic profiling is also possible if attackers combine existing data with other sources of information (e.g., social networks), investigate data from other similar users, or use advanced prediction methods.

A23. Location tracking

This is caused when agents within the IAAA flows can monitor the end-user locations over time through their computers and mobile devices. Location information can also be viewed from the logs of various applications, such as Wi-Fi history, IP addresses, and more. Location tracking can expose not only where an end user lives, works, or travels, but also their habits, hobbies, connections with others, and so forth.

A summary of attacks on FIDM components is given in Table 1.

3. Self-sovereign IDM

This section describes different SSI architectures and components before synthesizing the common threats toward these systems.

3.1. SSI reference architecture

SSI management allows users to manage and distribute their digital identities in a more decentralized manner. Instead of central storage, ad hoc communication, and peer-to-peer protocols are used to store and exchange identification data. The data itself is self-claimed by the user or asserted by sovereign bodies such as governments or corporations via an out-of-band trust formation (Schanzenbach, 2020).

There are many proposals for SSI; however, such systems share some similarities, such as the following (van Wingerde, 2017):

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

Table 1. *A summary of attacks[a] toward FIDM components, attacks will be marked ✗ if included*

Attack categories	C1						C2						C3				C4		C5	C6	C7		
Attacks	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23
User access rights	✗	✗	✗	✗	✗	✗											✗	✗	✗	✗			
User session							✗	✗	✗	✗	✗	✗					✗	✗	✗				
User information																	✗	✗			✗	✗	✗
IdP				✗		✗							✗	✗	✗	✗			✗				
Logging																	✗	✗			✗	✗	✗
Storage	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗					✗	✗			✗	✗	✗

[a]Classification of attacks: C1, Authentication/Authorization; C2, Session; C3, IDP; C4, Information disclosure; C5, DoS; C6, Privilege elevation; C7, Privacy. List of attacks: A1, Code/Token/State leak attack; A2, Token spoofing; A3, Replay of authorization code; A4, Manufacturing fake token; A5, XSS attack; A6, Third-party resource attacks (malware); A7, Redirect attack (307 redirect); A8, Naïve RP session integrity attack; A9, Flow interception; A10, Session handling; A11, Injection attack; A12, Man in the middle attack; A13, IdP mix-up attack; A14, CSRF attacks and third-party login initiation; A15, server-side request forgery (SSRF); A16, IdP account compromise; A17, Snooping; A18, Network eavesdropping attack; A19, DoS attack; A20, Elevation of privileges attack; A21, PII leakage; A22, User profiling; A23, Location tracking.

1. Users (or subjects) have an independent existence that relies on decentralized identifiers
2. IdPs issue verified claims that are linked to a user ID
3. Users can save self-asserted or verified claims on a personal repository
4. Users can give their consent after being informed who they wish to exchange certain parts of a claim
5. RPs can review attestations of a claim

A general SSI architecture was synthesized from such similarities, as can be seen in Figure 2. SSI has some components that are similar to FIDM, which are:

1. *Subject* or *holder*: subject is similar to *user* in FIDM, but this concept is extended to not only a person but can be a group, organization, or even a physical item that requires a unique identity without relying on any central authority (Kim et al., 2021).
2. *Issuers*: similar to IdP, which is an entity that can hold and state information about the subject. Once receiving a request from the subject, the issuer can check its proprietary data and issue a *verifiable credential* (VC) if the subject is satisfied.
3. *Verifiers*: similar to RP, which is a service provider that needs to verify whether users satisfy its requirement. The verifier can check the authority of the issuer by verifying its signature in the VC. If the VC is valid (i.e., issued by a legitimate issuer), the verifier can obtain the subject claim from that VC.

A key distinction between the SSI and FIDM architectures is the use of a *Verifiable Data Registry* (VDR). The VDR is a conceptual component that can be either internet-accessible or manually configured in each end system, storing all the necessary data and metadata such as issuer public keys, credential schema, credential definition, and a revocation registry (Preukschat and Drummond, 2021; Dixit et al., 2022). It serves a similar function as the Circle of Trust (CoT) in the FIDM model. Blockchain can be a viable choice for a VDR as it offers a highly tamper-resistant distributed database that is not controlled by any single entity (Bai et al., 2022). Other alternatives include centralized or distributed VDRs (Preukschat

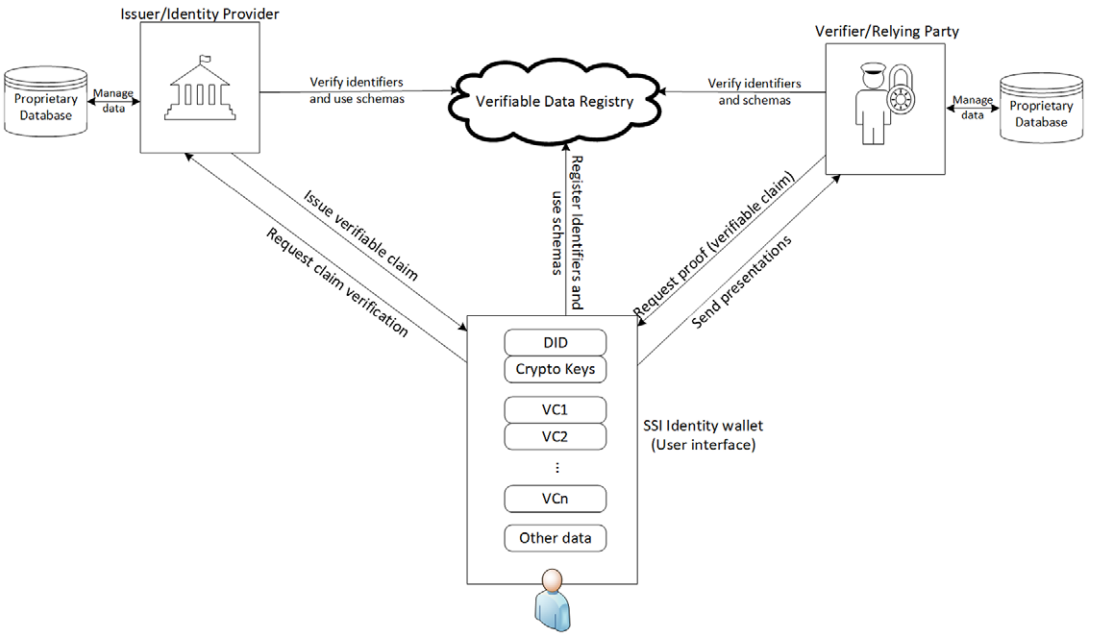

Figure 2. *A general architecture for SSI (based on* Preukschat and Drummond, 2021).

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

and Drummond, 2021), while the technology used to implement the VDR may continue to evolve over time as standards develop.

SSI subjects use the *Identity Wallet*, which is a software on a phone or computer, as the user interface. Identity Waller contains the credentials, including VCs, DID signature and verification keys, link secret, policy keys, and secret value commitments (Shuaib et al., 2021). It provides each edge agent (i.e., the device and software used to store, manage, and share SSI digital credentials such as mobile, laptop) to handle each different relationship and also contains an interface for a tamper-proof secure element, with which each agent communicates via a cryptographic secret key management system. A digital wallet stores identity verification information, including passwords, keys, and personal information; therefore, it is a lucrative target for attackers. If the private keys in a digital wallet are exposed, the attackers can access all services provided via this wallet under the owner's identity.

In addressing the cryptographic services needed to enable multi-device support and wallet synchronization, various key management solutions have emerged in the SSI landscape. One such solution is the *decentralized key management system (DKMS)*, which avoids the dependency on a single organization to secure and distribute keys on behalf of the identity holder. In DKMS, link secrets are used in each identity holder's credentials in blind form, allowing for the association of multiple credentials to a single logical identity holder, even when using various wallet software on different devices (Kim et al., 2021). DKMS also supports key revocation, agent revocation, key rotation, and recovery schemes. However, other key management solutions have gained prominence in SSI implementations. Hardware-based keys stored in the device or its peripherals, such as secure elements or trusted platform modules (TPM), offer enhanced security by isolating cryptographic operations from the main device environment (Edlira et al., 2022). Software-based key management solutions, on the other hand, can store keys on the device or in remote key stores, providing more flexibility and easier deployment (Edlira et al., 2022). These alternative key management approaches do not require the use of link secrets and may be more prevalent in certain SSI implementations.

To resolve the DID, SSI can use the *Universal Resolver* (UR) which enables the uniform search and resolution of DIDs using different DID methods (Soltani et al., 2021). The UR contains a variety of DID method drivers that interact with various providers. As UR is similar to the DNS resolver in the resolution method, it is susceptible to the same cache poisoning and pollution attacks that run on the DNS resolver.

3.2. SSI threat landscape

For comparison, we categorize the 20 potential SSI threats in a structure similar to the FIDM section. Attack classification involves seven groups, including Authentication/Authorization; Session Attacks; Issuer Attacks (similar to IdP attacks in FIDM); Information Disclosure; Denial of Service; Privilege Elevation; and Privacy. These attacks are presented in the following.

3.2.1. Authentication/Authorization attacks

B1. Key exposure attacks

Keys are the most critical part to ensure the security of wallets; therefore, there are many attempts to expose keys. For air-gap wallets, attackers can use some data exfiltration measures to extract keys. For example, Davenport and Shetty established a covert channel in a running blockchain-based wallet to infiltrate its private keys (Davenport and Shetty, 2019). Guri (2018) reviewed several techniques that can extract keys from Bitcoin wallets, including installing malware and worms into a physical wallet (e.g., USB flash drive or wallet that needs to connect to a host computer); reading binary information from electromagnetic emissions emitted from communications (e.g., PC display cables or radio signals); reading data on top of changes in current flow measured by electrical power consumption; or using magnetic, optical, acoustic, or thermal measurement devices to decode the data transferred between the air gap wallets and the host system. Although these techniques are reported to be successful in the lab testing environment, they require attackers to be skillful, having bespoke devices, and full access to the physical wallets. Therefore, it is unlikely that these attacks can happen in real-life or large-scale scenarios. With

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

software wallets, attackers can trick users to install malware by phishing websites. Malware (e.g., keylogger) can be used to capture passwords or sensitive information that can lead to key exposures.

B2. Reverse engineering

Attackers can overcome authentication by using reverse engineering and altering the digital wallet to bypass the authentication test results. It is possible for an attacker to use different tools to reverse engineer SW binaries of the DID wallet. Firstly, Code Analysis is performed to assess the overall security of the applications by searching for typical reverse engineering countermeasures. Haigh et al. (2018) identified ways to overcome the three most basic protection against reverse engineering in Android applications, including code obfuscation, signature verification, and installation verification. Attackers can circumvent protective measures with the right tools. For example, the code obfuscation may be decompiled using JEB or baksMALI, while signature and installation verification can be bypassed by recompiling with Keytool or Jarsigner to install the DID wallet on other devices. In case the keys are stored in the phone hardware, reverse engineering attacks are much more difficult to conduct as it requires physical access to the wallet and other advanced invasive techniques to detach the keys from the firmware or hardware memory.

B3. FIDM-inherent authorization/authentication attacks

SSI can inherit FIDM technology such as OpenID Connect to take advantage of its user-centric design while maintaining interoperability with current web technologies (Yasuda et al., 2022). Built on the OAuth 2.0 protocol, OpenID Connect ensures secure authorization flows, user authentication, and consent-based identity provision. However, adopting this also means that SSI could inherit some vulnerabilities, including potential attacks such as A1–A4 associated with the OpenID Connect protocol. Successful attacks could lead to unauthorized access or user impersonation, posing significant risks to privacy and data security within an SSI framework. In response, OpenID for SSI has implemented specific measures to mitigate these threats (Yasuda et al., 2022). The authorization code flow and the preauthorized code flow secure the authorization process, reducing risks such as client impersonation and authorization code interception. Measures include sender-constrained tokens that are cryptographically bound to key material, which must be authenticated by the client during token usage to prevent token leakage. Additionally, a pre-authorized code can require the entry of an end-user PIN, adding an additional security layer and further mitigating attack risks (Yasuda et al., 2022). Despite these safeguards, the risks associated with online identity systems mean that these attacks cannot be completely eliminated. However, OpenID for SSI continually strives to minimize these vulnerabilities with customized authorization flows designed for the SSI context.

3.2.2. Session attacks

B4. Man-in-the-middle (MitM) attack

When establishing communication between the identity holder and verifier, some plain text messages must be transferred before creating a trusted communication. These situations are known as trust on first use or blind trust before verification. Attackers can exploit this time to launch MitM attacks to expose critical information, depending on the protocols used. Conti et al. (2016) examined a wide range of network protocols that are susceptible to MitM attacks. In DID communication systems that use UR, SSL/TLS, and Boarder Gateway Protocol (BGP), the MitM attack has been shown to be possible. Some of the potential scenarios include DID Spoofing-based MitM attack through UR; DID SSL/TLS/MitM attack via two distinct SSL connections maintained by the attackers; and DID BGP MitM attack by traffic tunneling via attacker's Autonomous Station.

In addition to that, if the wallet keys are exposed, attackers can also set up a backdoor channel and act as a MitM to monitor, intercept, or alter agent-to-agent message traffic.

B5. Phishing/Impersonation attack

Phishing is the act of impersonating another party to target a victim by stealing information or money (Steinebach et al., 2021). These attacks may be conducted through different impersonation forms, such as

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

legitimately looking websites, emails, or a UI of a third-party app. Altering the website may typically be done automatically by changing specific patterns such as URLs and cryptocurrency addresses. Attackers can also conduct a MitM attack to view the latest version of the targeted website. The phishing website then makes a request for each request from the user, while attackers can examine the information in real-time and modify relevant information such as the bitcoin addresses before delivering the page.

3.2.3. Issuer attacks

B6. Fraudulent issuance

This type of attack occurs when an illegitimate issuer creates and distributes counterfeit credentials. The impact of such an attack could be high, leading to identity theft or unauthorized access to services. The likelihood is relatively low in an SSI ecosystem due to the cryptographic verification of issuers and credentials, but it is not zero, particularly if an attacker can compromise an issuer's private keys.

B7. Issuer impersonation

This attack occurs when an attacker poses as a legitimate issuer, tricking users into accepting false credentials. If successful, the impacts are similar to fraudulent issuance, including identity theft and unauthorized access to services. The likelihood is low due to cryptographic verification and DIDs which uniquely identify issuers.

B8. Disclosure of confidential information

In scenarios where issuers hold sensitive user data, a breach could lead to unauthorized disclosure. This could result in privacy violations and identity theft. The likelihood of this risk can vary depending on the security measures in place, but with the SSI paradigm in which users primarily control their data, the overall risk is typically reduced.

3.2.4. Information disclosure

B9. Snooping

Snooping attack poses a real threat if care is not taken to protect systems. An unsecured device could potentially have snooping malware installed to steal sensitive information. For SSI, risks are high if monitoring occurred, since digital IDs could be compromised. However, SSI does encourage security best practices like local key control, encryption, and secure channels. Such measures, if implemented properly, may help reduce exposure to monitoring attempts.

B10. Network eavesdropping

The risk of network eavesdropping, where attackers intercept data packets over network connections, depends greatly on the security of communication channels. SSI typically uses strong encryption for both data at rest and in transit, making eavesdropping significantly more challenging. Additionally, SSI often uses DIDs which can be resolved without exposing sensitive information, which minimize the impacts of network eavesdropping attacks.

B11. Wallet query language (WQL) injection attacks

When the smartphone-based wallets use an external SQL database, attackers can use the WQL injection attacks employing code injection techniques, which is comparable to SQL injection. For example, attackers may inject dangerous codes into WQL strings to enable data exfiltration from the DID wallet. This is possible because the inputted codes are then transmitted to the DID wallet SQL Server for processing and execution. Attackers can modify the query to obtain the information they desire. Two common WQL injection attacks are the *direct insertion* of malicious codes into user input variables that are concatenated with WQL instructions, and the *indirect attack* injects malicious code into strings included in the table or metadata of the DID wallet database. In the latter scenario, if the strings stored in the DID wallet are concatenated into a dynamic WQL command, malicious code can be executed. WQL injection attacks are more likely to be found in systems with loss of access control, misuse of privilege accounts, or unprotected input validation in the wallet database (Kim et al., 2021).

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

B12. DID wallet database information disclosure attack

Some sensitive information such as Personally Identifiable Information (PII) or High Business Impact (HBI), can be stored in plaintext in some kinds of wallets for richer and faster searchability. Plaintext can be susceptible to information disclosure attacks through untrusted SD cards or local storage. Moreover, extra information may be tagged to improve accessibility. Tag names, record IDs, and record values are always encrypted, except when a particular prefix is appended to the name of the tag value. This occurs when users wish to perform certain complex searches, such as comparison queries or predicates values like $gt (greater than), $lt (less than), or $like. Attackers can use this functionality to expose sensitive information (Kim et al., 2021).

B13. Jailbreak/Rooting attack against DID identity wallet

Reverse engineering techniques and elevated privilege can also be combined to carry out a jailbreak or rooting assault on the DID identity wallet (Kim et al., 2021). Attackers may arbitrarily alter the mobile system (e.g., rooted the Android phone, jailbreaking the iOS device, debugging applications) and thus influence the execution and data of the merchant app and embedded TP-SDKs (Yang et al., 2019). Client applications are typically considered untrustful because all the data handled by applications may be modified by an attacker using a rooted Android phone or a jailbroken iOS device. Jailbreaking significantly simplifies the installation of third-party applications, and many users who jailbreak their phones do not know how to alter the default user and password or take the necessary preventive measures (Talal et al., 2019). As a result, the DID wallet installed on the Jailbreak/Rooting devices can be exploited to extract sensitive information or gain unauthorized access.

3.2.5. DoS attack

B14. Cache poisoning/Pollution attack

In this threat, attackers damage the DNS resolver cache and cause the server to deliver an incorrect result that renders a particular location on the network inaccessible (Davenport et al., 2018). DNS amplification attack can lead to a DDoS attack. Attackers can forge search queries into DNS servers to hide the origins of the vulnerability and send the response to the target network. Attackers can also convert the basic DNS query into a larger payload to launch DDoS attacks or to change and/or steal access and permission in blockchain certificates.

B15. VDR partitioning

This attack can occur in the VDR implementation that uses the blockchain. In detail, when the hashing power is unevenly distributed in the blockchain, that is, excessive aggregation of maintenance nodes such as full nodes taking part in the mining process of the Bitcoin network. If the mining pool uses the stratum overlay log server with a public IP address, there is a possibility of routing and flood attacks that can lower the hash rate by up to 50% while increasing the latency to as high as 20 minutes (Kim et al., 2021). These circumstances will disrupt the services offered by VDR.

B16. Social recovery attack

Key recovery is a desirable feature in many applications, particularly when users forget their keys or lose their devices. The methods for key backup and recovery can differ based on the specific implementation of the SSI system. For example, some implementations, such as the Type 1 European Digital Identity Wallet that relies on hardware-based keys, do not support key backup and recovery due to the nature of secure storage. Conversely, in other situations, users may regularly backup their wallets and store encrypted backups on cloud agents or other secure digital storage platforms (Schäffner, 2020). In systems that employ the DKMS, during the recovery setup phase, the edge agent generates a recovery file. This file contains the user link secret, the decryption key for the DID wallet backup, and the Certify Authority recovery endpoint (Kim et al., 2021). This recovery file is then split into multiple shares that are distributed to various trustees. If users need to recover the key, the edge agent sends the key recovery request to the trustees and collects the shares to regenerate the backup keys. However, attackers can

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

exploit the Tompa-Woll attack, colluding with trustees to interfere with the key recovery process in the DKMS (Kim et al., 2021).

3.2.6. Privilege elevation attack

B17. Elevation of privileges attack

Attackers can compromise a system to obtain sensitive information such as the ID of the wallet, the type of storage, and the storage settings, including the location of the wallet files and key generation methods, which are then used to infiltrate target devices. Attackers can then alter their database or files to obtain higher privileges (Hoang et al., 2019). Various threats to devices in smart transactions may represent a danger to the usage of Blockchain Ethereum, which is an essential factor in the management of privilege and personal information (Min, 2019). Elevation of privileges attacks can lead to the exploitation of transaction data, falsify, or tamper with personal information enquiry. Other targets that can be abused are root access permission to data or actions, root account, domain admin account, or other accounts that can access specific components in the DID wallet system (Kim et al., 2021).

3.2.7. Privacy threats

B18. Personal identifiable information (PII) leakage

In the context of SSI, the threat of PII leakage remains relevant. This could occur if sensitive data, potentially stored in a user's wallet, are transmitted over insecure communication channels without proper encryption. Attackers with network or web access could intercept or illicitly acquire these data, exploiting them for malicious purposes. However, SSI architectures adopt stringent security measures to counteract this threat. Data are typically encrypted at rest and in transit, and users have more control over their data, reducing the chance of leakage. Additionally, the use of zero-knowledge proofs allows users to verify claims about themselves without revealing the actual data, further reducing the risk.

B19. User profiling

This threat involves attackers who monitor user behavior on various platforms to create detailed profiles. Within an SSI framework, this could involve tracking user interactions with their digital wallets or other SSI services. The decentralized model and the use of pseudonymous identifiers can reduce the ease and effectiveness of such profiling. Additionally, consent-based sharing means that users have control over which data they share and with whom, limiting the amount of data available for profiling. However, a critical subtlety arises when users present the same selectively disclosed VC (such as through SD-JWT) to one or multiple RPs. In such cases, each presentation can be interconnected via the DID, resulting in a more extensive user profile. Even if a user utilizes different DIDs for varied RPs, presenting successive selectively disclosed data to an identical RP can yield expansive user profiles. To mitigate this threat, it is imperative that users employ a distinct DID for every RP and for each presentation.

B20. Location tracking

In SSI systems, location tracking could potentially occur if agents within the system monitor end-user locations over time using various applications and devices. However, SSI has measures to reduce the risk of location tracking. First, information sharing in SSI is typically consent-based, meaning users have control over what data they share, including location data. Second, by employing decentralized identifiers and various privacy-preserving technologies, the ability for agents within the system to track a user's location can be significantly limited.

A summary of attacks on SSI components is given in Table 2.

4. Use case analysis: cross-border IDM for European countries

This section considers a security analysis use case for the two IDM approaches (i.e., FIDM and SSI) to manage the identity between the European countries. One of the main goals of such IDM systems is to enable people and businesses to use their own national electronic identification schemes (eIDs) to access

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

Table 2. *A summary of attacks[a] toward FIDM components, attacks will be marked ✗ if included*

Attack categories	C1			C2		C3			C4					C5			C6	C7		
Attacks	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	B18	B19	B20
DID		✗	✗	✗			✗			✗	✗	✗	✗			✗	✗			
DID subjects/Users		✗	✗	✗	✗		✗				✗	✗	✗			✗	✗	✗	✗	✗
Issuer						✗	✗	✗										✗		
Identity wallet	✗	✗			✗						✗	✗	✗				✗	✗		
Universal resolver				✗																
Verifiable data registry														✗	✗			✗		

[a]Classification of attacks: C1, Authentication/Authorization; C2, Session; C3, Issuer attack; C4, Information disclosure; C5, DoS; C6, Privilege elevation; C7, Privacy. List of attacks: B1, Key exposure attacks; B2, Reverse engineering; B3, FIDM-inherent authorization/authentication attacks; B4, Man-in-the-middle (MitM) attack; B5, Phishing/Impersonation attack; B6, Fraudulent issuance; B7, Issuer impersonation; B8, Disclosure of confidential information; B9, Snooping; B10, Network eavesdropping; B11, Wallet query language (WQL) injection attacks; B12, DID wallet database information disclosure attack; B13, Jailbreak/Rooting attack against DID identity wallet; B14, Cache poisoning/Pollution attack; B15, VDR partitioning; B16, Social recovery attack; B17, Elevation of privileges attack; B18, Personal identifiable information (PII) leakage; B19, User profiling; B20, Location tracking.

public services in other EU countries where eIDs are available. The system will create a trust network at European level for electronic services by ensuring that they will work across borders and have the same legal status as traditional paper-based procedures (Cuijpers and Jessica, 2014). This will lead to a reduction in bureaucracy for citizens, more savings for companies, greater security, and superior convenience. Some examples of practical activities that can benefit from this system are tax declarations, enroling in a foreign university, opening a remote bank account, starting a business in another Member State, authenticating for Internet payments or participating in an online tender (European-Commission, 2022). Implementing such a system is very complex due to its very large-scale eID system (involving nearly 500 million citizens). In addition to that, there are security and user privacy issues that restrict the widespread adoption of the system in different institutions and nations.

There are various proposals for European IDM systems. One of the best-known solutions that uses the FIDM approach is the European Identity Federation Initiative (eIDAS), which was established in 2014 (Carretero et al., 2018. Since eIDAS was developed mainly to improve the convenience of identity transactions between European countries, it does not consider providing identity control to users as in the SSI concepts. There have been several attempts to bridge eIDAS to SSI. For example, Preukschat and Drummond (2021) considered two scenarios, including SSI implementation behind current existing eID connection nodes (the proxy on the border between the two nations) or the use of a middleware model to provide the SSI operational protocols and artifacts for replacing the eIDAS proxy nodes. Some researchers have also tried to redesign the system by considering the self-sovereign requirements from the very beginning. For instance, the Aries project (Bernabe et al., 2020) has developed and implemented a privacy-preserving and user-centric IDM framework and associated management practices that ensure usability and flexibility for IDM processes. Despite improvements in privacy-preserving and more usage options, a feasible SSI solution is not yet available due to the differences in regulations between countries, making it difficult to issue and distribute other VCs or presentations (Preukschat and Drummond, 2021). Since there is no practical SSI solution for managing eID on a large scale, this article selects a theoretical SSI model using blockchain, which was presented in Bernabe et al. (2019), for comparison with the eIDAS system. The components and functionalities of these two systems are described below.

4.1. Federated IDM: eIDAS

The eIDAS system consists of a network of Member States, each of which subscribed to a federated operator, namely the eIDAS Node. Each Member State has to provide an eIDAS Node that acts as an IdP for the national eID of other countries. All SPs participating in a national network must be subscribed to the eIDAS Node of this country. Every citizen recognized by a Member State should be recognized within the trust network at the European level, enabling the use of services in other Member States that were previously not allowed or whose concession was laborious (Carretero et al., 2018. Note that there is a trade-off between convenience and security, as the eIDAS system skips the standardization of the authentication method (eID Scheme) of the Member States for cross-border authentication, although each country has different security level of the eID scheme.

The provision of the nodes to establish this trust network is the responsibility of a governmental institution of the Member State (e.g., a ministry), as it is linked to the national public eID scheme.

The workflow of eIDAS is illustrated in Figure 3. The details of this process are as follows (Carretero et al., 2018:

1. The citizen applies for access to a SP in her host country with the home eID (Company A).
2. The SP for Company A sends the request to its own eIDAS-Node Connector using a HTTP Redirect Binding (or HTTP POST Binding) that contains a SAML AuthnRequest.
3. The eIDAS-Node Connector in Company A forwards the SAML request in Step 2 to the eIDAS-Node Proxy Service of the citizen's Member State (Company B).

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

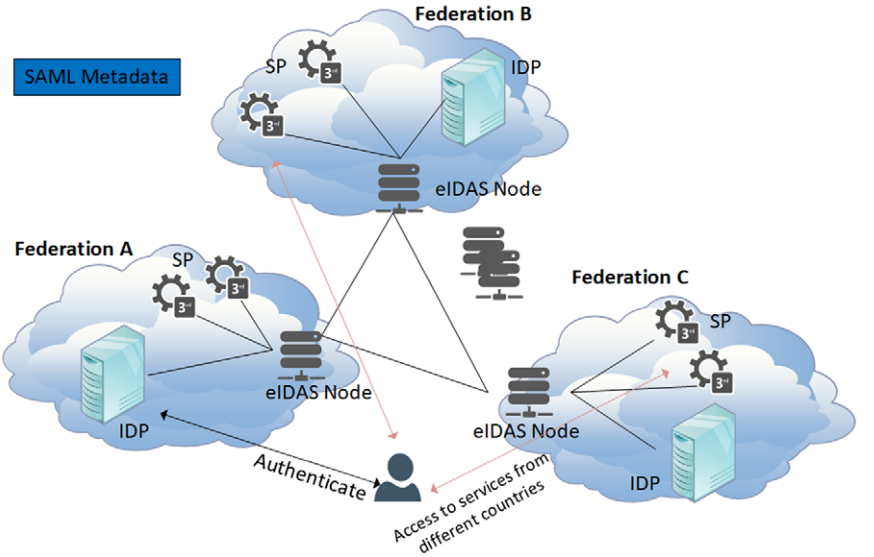

Figure 3. *High-level architecture of eIDAS (based on Carretero et al., 2018).*

4. The citizen authenticates himself at the IdP of his country with his eID and the confirmation is forwarded via the IdP with an HTTP POST Binding to the eIDAS-Node Proxy Service. This step can have two additional steps depending on the implementation:
   - for the citizen to choose the attributes to be provided (therefore give consent);
   - for the citizen to agree on the values of the attributes to be assigned.
5. The eIDAS-Node Proxy Service sends a SAML Response with an encrypted SAML Assertion to the requesting eIDAS-Node Connector, which forwards the response to the SP.
6. The SP grants access to the citizen.

4.2. SSI approach: theoretical model

The main components and activities of the model can be seen in Figure 4. Users (identity owner) can have DIDs and obtain verifiable claims and credentials from the Issuer authority using his smartphone. Users' private keys are securely stored in their digital wallets. To increase privacy-preserving capabilities, users can be equipped with means to present Zero-Knowledge Proofs (ZKP) against a SP who acts as a verifier that verifies the attestations and signatures on the blockchain. Blockchain enables sovereignty, as users can be equipped with means to transfer digital assets, including DID, DID documents, identity attributes, verifiable claims, and proofs of identity (including ZKPs), to anyone privately and without rules (Bernabe et al., 2019. Blockchain can also act as a distributed and reliable identity verifier that provides the origin and verfiability of identities.

4.3. Cyber risk comparisons between the two systems

According to Engelbertz et al. (2019), the eID attackers mainly aim at:

1. Disrupt the system services by creating internal system errors.
2. Reducing the availability of the targeted service, for example, by consuming a large amount of computing resources.
3. Accessing services on the internal network, such as cloud instance metadata, internal databases, or local file system.

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

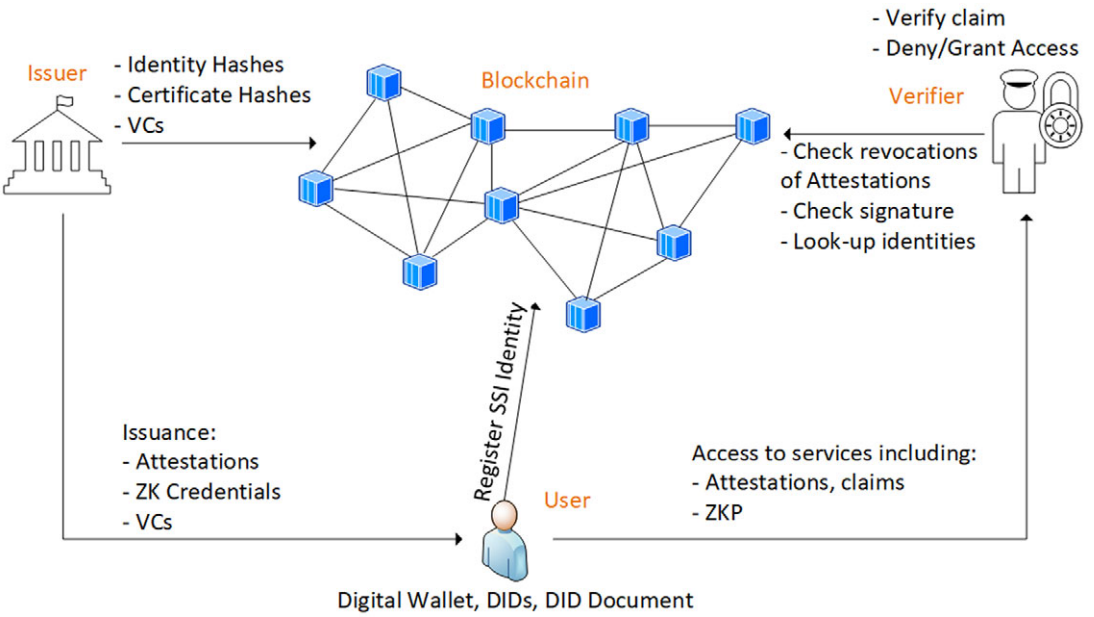

Figure 4. *Self-sovereign IDM using blockchain (based on* Bernabe et al., 2019*).*

4. Accessing to identity data, for example, exfiltration if there is a direct return channel at application level. Attackers can also try to expose users' privacy via data leakage.
5. Bypassing signature protection and injecting arbitrary XML elements, for example, by manipulating the exchanged data or smuggling harmful XML content. Consequently, attackers can trick the server logic into processing newly inserted elements while the signature validation logic is still confirming a successful signature verification.

We evaluate the risks coming from these attackers via the following security and privacy criteria, which were influenced from Zhang et al. (2021):

1. *Consistency*: Identity data stored on a node (or IdP) should remain consistent for verification. For SSI, it means that all copies of the public identity data (which can be stored on the blockchain) maintain the same state at the same time even in the cases of partial failures and attacks.
2. *Integrity*: Identity data packaged in a transaction cannot be changed with others during the entire communication and storage process.
3. *Authenticity*: Users must confirm that the transaction is authenticated, that is, that it was provided by a legitimate owner with genuine content.
4. *Availability of system and transactions*: Availability implies that users can access transactional data anytime, anywhere, including system-level and transaction-level availability. The former relates to the requirement that the system function reliably even in the event of large-scale network attacks. The latter implies that the recorded transactions are always available to users without being destroyed, damaged, or disturbed.
5. *Confidentiality*: The confidentiality means that (1) an unauthorized user cannot successfully read or infer any private information from the stored data and (2) the confidentiality of identity data must be ensured even in the event of an unexpected failure or malicious network attack.
6. *Anonymity of users*: Users can ask for anonymity when authenticating and transmitting identity data. For example, users can request that their sensitive personal data to be unlinked from the information used for authentication.

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

7. *Identity-control*: Identity control includes (1) the users should control (or consent to) who can access which part of their identity data and (2) the system administrator or other users cannot disclose the identity data to third parties without the consent of the users.
8. *Fine-grained access control of transactions*: When exchanging identity data with others, users do not have to reveal all of the attributes (claims) contained in a credential, but they can disclose only the minimal information as required from the verifiers.

It should be noted that the first five criteria focus on security requirements, while the last three criteria focus on privacy needs for the operations of the IDM systems.

In the following, we will refer to the FIDM system in Section 4.1 as A and the theoretical SSI system in Section 4.2 as B. A summary of the comparisons is given in Table 3. The details for each criterion are as follows.

Consistency: In system A, the main consistency risks arise from IdP attacks (A13–A16) that could lead to unauthorized modification of user identity records stored on eIDAS nodes, with the most severe impact potentially coming from compromise of an eIDAS node account itself (A16). Others are related to code and token theft (A1–A4) or injection attacks (A11), which could allow unauthorized data modification. Despite having robust server-side protections, system A faces more significant consistency risks due to potential high-impact attacks targeting eIDAS nodes. On the contrary, the main consistency risks that system B faces include man-in-the-middle attacks (B4) and issuer impersonation attacks (B7), which could introduce inconsistencies between nodes; and some attacks toward users' identity wallet (B1–B2, B5, B11, B13, B17). Due to its decentralized nature and the use of a distributed ledger to store identity records, system B can offer more robust consistency protection. Comparatively, system B appears to offer a significantly lower risk profile with respect to consistency.

Integrity: In system A, the main integrity risks arise from attacks aimed at unauthorized data modification, particularly those that target eIDAS nodes, such as IdP account compromise (A16) and server-side request forgery (A15). A successful attack could lead to large-scale data manipulation, hence posing a significant threat to data integrity. Other integrity risks come from attacks such as token theft (A1–A4) or session attacks (A7–A10), which do not directly alter data, but could conceal data corruption. In contrast, for system B, the main integrity risks originate from MitM attacks (B4) and cache poisoning attacks (B14), both of which could allow undetected tampering of identity data before it reaches the blockchain. Other risks come from issuer impersonation (B7), which could directly alter immutable user identity claims, and information disclosure attacks on the wallet database (B12), which could expose data

Table 3. *A comparison of cyber risks between the two IDM systems - The red, orange, yellow colour indicates high, medium, and low risk respectively.*

	System A: eIDAS			System B: SSI			
Criteria	Likelihood	Impact	Risk	Likelihood	Impact	Risk	Comments
Consistency	L	H	M	L	L	L	The security threats on the SSI systems tend to have lower likelihood and impacts compared to those of FIDM, but the FIDM vulnerabilities can be fixed faster
Integrity	M	H	M	L	H	M	
Authenticity	H	M	M	L	M	L	
Availability	H	H	H	L	M	L	
Confidentiality	H	M	M	M	M	M	
Anonymity	H	M	M	L	L	L	SSI systems have much lower risks of privacy leakages compared with the FIDM systems
Identity-control	M	M	M	L	L	L	
Fine-grained	H	M	M	M	L	L	

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

to improper modification. When comparing these two systems, both face high-impact integrity threats that could result in large-scale alterations of identity data. However, system B's inherent blockchain architecture adds an additional layer of security, as manipulations require higher levels of sophistication from attackers. Considering other controls in system B such as tamper-evident protocols and multiparty authorization schemas, it can be concluded that system B offers a lower risk of integrity violation than system A.

Authenticity: System A faces authenticity risks primarily from attacks that facilitate user impersonation, such as token theft (A1–A4), session hijacking (A7–A10), and particularly IdP account compromise (A16), which could significantly aid large-scale impersonation. Other authenticity risks relate to attacks such as XSS (A5) or injection (A11), which also have the potential for identity spoofing, but require more technical expertise. In system B, the main authenticity risks arise from issuer impersonation (B7) and key exposure (B1), both of which could lead to false credentials or forged transactions. Other risk arises from phishing/impersonation (B5), with the protection coming from cryptographic verifiability. When comparing the two systems, system A, despite the high level of security offered by eIDAS, appears to be more vulnerable to authenticity attacks, especially those targeting user access and session control. System B, through the use of decentralized identifiers, credential verification schemas, and blockchain-based consensus, offers strong defenses against inauthentic transactions, regardless of whether VDR uses blockchain technology or not. However, it is important to note that system B must still manage the risks of compromised issuer keys or sophisticated phishing attempts. In summary, system B appears to exhibit a lower risk profile in relation to authenticity compared to system A.

Availability of system and transactions: System A, which is centralized, faces a high level of risk from distributed denial-of-service (DDoS) attacks (A19). These attacks could inundate eIDAS nodes and disrupt the availability of system and transaction transactions across EU borders. Other availability risks pertain to attacks like injection (A11) and SSRF (A15), which could potentially crash local nodes, and thus degrade the system's availability. However, these threats require a higher level of technical expertise and impact fewer nodes than a DDoS attack. In contrast, system B, being decentralized, is designed to offer greater resilience to availability disruptions. However, it faces significant risks from VDR partitioning attacks (B15) that could disrupt system availability by increasing transaction latency or completely preventing transactions. Other relevant attacks include cache poisoning (B14), which could make network locations inaccessible. Transaction-level availability risks arise from potential wallet database information disclosure attacks (B12) and social recovery attacks (B16), which could disrupt the key access necessary for transactions. In comparing the two systems, system A appears to be more vulnerable to availability attacks due to its centralized nature and the high likelihood of DDoS attacks. System B, on the other hand, enjoys inherent resilience due to its decentralized architecture but remains susceptible to targeted attacks on network infrastructure and data access. However, it can maintain a high level of availability for identity transactions and data with proper safeguards, such as redundancy, integrity checks, and multi-party authorizations. Therefore, with the right precautions, system B potentially offers a lower risk profile for availability compared to system A.

Confidentiality: System A faces main confidentiality risks from attacks such as network eavesdropping (A18) which could potentially expose sensitive user data if communications between eIDAS nodes are unencrypted. Other threats like snooping (A17) and SSRF (A15) can acquire user data through side channels, although they need technical expertise and have less impact than wholesale data intercept. On the other hand, system B, due to its decentralized nature, presents a more complex set of confidentiality risks. Notable risks include information disclosure attacks to the wallet database (B12), which could expose sensitive user information, and network eavesdropping (B10) that could lead to data packet interception, for example, in communication between users and RP when the users send VCs to RP. Other attacks such as phishing (B5) and code injection (B11) can also trick users into revealing private information or extracting it through malicious code. Comparatively, system A seems more susceptible to confidentiality breaches due to its higher-likelihood attacks, particularly network eavesdropping. On the contrary, system B, with its user-managed access controls and minimum disclosure mechanisms,

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

presents a lower risk profile despite its possible attack vectors. Therefore, system B seems to offer a slightly more secure environment regarding confidentiality compared to system A.

Anonymity of users: For system A, user anonymity is significantly threatened by potential tracking and profiling attacks such as network eavesdropping (A18) and user profiling (A22). Large-scale intercepting of unencrypted traffic could reveal identifying information, facilitating profiling, while targeted monitoring could identify usage patterns and undermine user anonymity. Other threats like snooping (A17) or SSRF (A15) can potentially deanonymise users over time and across multiple contexts, although these require more technical proficiency. On the contrary, system B operates under the anonymity criterion that requires users to authenticate and share data without revealing sensitive personal information. The main risks include user profiling attacks (B19) and location tracking (B20), which could expose identifiers and compromise anonymity. Other risks such as phishing (B5) and network eavesdropping (B10) pose threats to anonymity through the potential disclosure of private data and interception of anonymized transactions. However, system B's architecture, incorporating decentralized identifiers, zero-knowledge proofs, and consent-based data sharing, offers robust protections to preserve user anonymity. When comparing the two systems, it appears that system A is much more susceptible to anonymity breaches due to its higher probability of threats that can reveal user data. In contrast, system B, with its use of system identifiers and privacy-preserving technologies, presents a much lower risk profile for anonymity attacks.

Identity-Control: The main identity-control risks that system A faces aris from network eavesdropping (A18) and Personal Identifiable Information (PII) leakage (A21). Unencrypted traffic interception or data breaches could result in nonconsensual exposure of user information, leading to significant violations of identity control. Other risks come from threats such as snooping (A17) or profiling or tracking (A22–A23) can lead to unauthorized acquisition of user data through side channels, subtly undermining user control. On the other hand, system B operates under the identity-control criterion, stipulating that users should command control over their identity data access and prevent unauthorized disclosure. However, certain attacks can potentially subvert this control. The most relevant threats include user profiling (B19) and network eavesdropping (B10), which could lead to nonconsensual data gathering and identity transaction interception, respectively. Other threats such as phishing (B5) and code injection (B11) could also trick users into revealing private data or extract uncontrolled data. However, system B's defenses against these threats, which include consent-based data sharing, encryption, and decentralized storage, provide strong identity control protections. Comparatively, system A is subject to greater risks regarding identity control due to the relevant privacy attacks having greater likelihood and impacts. In contrast, system B, where user consent is mandatory before entities can access their Verifiable Credentials (VCs), presents lower risks regarding identity control. Therefore, system B's user-centric approach and consent-based architecture provide a more secure environment for maintaining identity control compared to system A.

Fine-grained access control: System A poses significant risks to this criterion, particularly from attacks like network eavesdropping (A18) and PII leakage (A21) that could nonconsensually expose user attributes, thus undermining fine-grained control. Server account compromises (A16), while less probable, could have severe consequences, leading to full dataset exposures. Other relevant threats include injection (A11) or SSRF attacks (A15) which could manipulate backend queries to access unauthorized data. On the other hand, system B operates under the fine-grained access control criterion, which advocates selective disclosure of identity attributes. However, it still faces some relevant threats include user profiling (B19) and network eavesdropping (B10), which could lead to uncontrolled data aggregation and complete transaction interceptions. Other threats include reverse engineering (B2) and code injection (B11) which could also extract unauthorized data. Despite these threats, system B provides substantial protection mechanisms such as zero-knowledge proofs, encryption, and consent-driven sharing to ensure fine-grained control. Comparatively, System A presents a higher risk, as it lacks user-centric fine-grained access control, making it susceptible to privacy attacks such as A21–A23. In contrast, system B, through the use of advanced protocols like ZKPs and selective disclosure non-ZKP schemes such as atomic credentials (W3C, 2019), enables fine-grained access control by requiring minimal identity information disclosure. Therefore, system B provides a more secure and privacy-preserving environment for identity data management than system A.

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

Overall, SSI design seems to have lower risk profile across most dimensions, provided that certain safeguards, such as multi-signature schemas, multiparty authorization schemas, user-managed access controls, minimum disclosure mechanisms, decentralized identifiers, zero-knowledge proofs, consent-based data sharing, and so forth are proper implemented. Another important point is the "decentralized" characteristic that shifts the risks from the server side more to the client side (e.g., user wallet) when switching between FIDM and SSI. Consequently, the impacts of cyber-attacks on SSI are lower as they often influence a smaller number of users and do not scale as quickly as that of the FIDM systems. However, FIDM vulnerabilities can be identified and fixed faster than those of the SSI system, since SSI users may not be aware that there are security vulnerabilities in their systems, while with FIDM the server takes care of security. As a result, the impacts of attacks on SSI can be longer-lasting than those on the FIDM system. SSI systems also offer much better privacy protections compared to FIDM systems, mainly because the FIDM lacks consideration of privacy issues in its design.

5. Conclusion

In this article, we have systemically examined the architectures and threat landscapes of the federated and self-sovereign IDM systems. First, we looked at a general architecture that contains the main components and functionalities of each system. We carefully analyzed the potential attacks to create extensive attack surfaces for both systems. In detail, we identified and categorized 23 common threats to FIDM systems into seven main targets. On the other hand, 20 common SSI threats were also classified similarly to threats in FIDM for easing the comparisons. We applied this knowledge in threat modeling to a cross-border European IDM use case to compare the cyber risks of the two approaches in eight criteria. We found that the SSI design generally offers lower attack impacts, less risk, and better privacy than the FIDM system. However, SSI designs should enhance the security robustness of the user's system (e.g., stronger security for digital wallets and proactive self-monitoring of user identity transactions), as well as study user behaviors to understand potential weaknesses and reduce the impacts of cyber threats in the long term. The limitations of this research are that it is conducted based on assumptions and theoretical simplifications rather than trying to capture all the diversity and complexity of different IDM concepts and implementations. As the IDM approaches, SSI-related solutions in particular may evolve in the future, the relevant architecture, attack surfaces, and risks may change, which requires revisions and updates. In the future, we plan to expand this research to design an IDM that addresses and minimizes security risks in specific domains such as healthcare.

Acknowledgments. The authors would like to thank Mark Hooper for reviews and comments on this article. We also thank the three anonymous reviewers whose comments/suggestions helped improve and clarify this manuscript.

Author contribution. Conceptualization: G.E., C.M.; Formal analysis: A.L.; Funding acquisition: C.M.; Methodology: A.L., G.E., C.M.; Project administration: G.E.; Supervision: G.E., C.M.; Writing—Original draft preparation: A.L.; Writing—Review, editing, and approved the final version of the manuscript: G.E., C.M.

Funding statement. This work was supported, in whole or in part, by the Bill & Melinda Gates Foundation (INV-001309). Under the grant conditions of the Foundation, a Creative Commons Attribution 4.0 Generic License has already been assigned to the Author Accepted Manuscript version that might arise from this submission.

Competing interest. The authors declare none.

References

Aldosary M and Norah A (2021) A survey on federated identity management systems limitation and solutions. *International Journal of Network Security & Its Applications 13*, 43–59.

Ayed GB (2014) *Architecting User-Centric Privacy-as-a-Set-of-Services: Digital Identity-Related Privacy Framework*. New York: Springer.

Bai P, **Kumar S**, **Aggarwal G**, **Mahmud M**, **Kaiwartya O and Lloret JS-S** (2022) Identity management model for smart healthcare system. *Sensors 22*(13), 4714.

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

Bernabe JB, **Luis CJ**, **Hernandez-Ramos Jose L**, **Torres MR and Antonio S** (2019) Privacy-preserving solutions for blockchain: Review and challenges. *IEEE Access 7*, 164908–164940.

Bernabe JB, **Martin D**, **Torres MR**, **Presa CJ**, **Sébastien B and Antonio S** (2020) ARIES: Evaluation of a reliable and privacy-preserving European identity management framework. *Future Generation Computer Systems 102*, 409–425.

Cabarcos PA (2013) *Dynamic Infrastructure for Federated Identity Management in Open Environments. Doctoral Dissertation, Universidad Carlos III de Madrid.*

Carretero J, **Guillermo I-M**, **Mario V-C and Javier G-B** (2018) Federated identity architecture of the European eID system. *IEEE Access 6*, 75302–75326.

Conti M, **Dragoni N and Lesyk V** (2016) A survey of man in The middle attacks. *IEEE Communications Surveys & Tutorials 18* (3), 2027–2051. http://doi.org/10.1109/COMST.2016.2548426.

Cuijpers C and Jessica S (2014) eIDAS as guideline for the development of a pan European eID framework in futureID. In *Presented at the Open Identity Summit 2014, Gesellschaft für Informatik* (Vol. *237*). Berlin: Gesellschaft für Informatik e.V., pp. 23–38.

Davenport A, **Sachin S and Xueping L** (2018) Attack surface analysis of permissioned blockchain platforms for smart cities. In *IEEE International Smart Cities Conference*. Kansas City, MO: IEEE, pp. 1–6.

Davenport A and Shetty S (2019) Air gapped wallet schemes and private key leakage in permissioned blockchain platforms. In *IEEE International Conference on Blockchain (Blockchain)*. Atlanta, GA: IEEE, pp. 541–545.

Dixit A, **Smith-Creasey M** and **Rajarajan** (2022) A decentralized IIoT identity framework based on self-sovereign identity using blockchain. In *2022 IEEE 47th Conference on Local Computer Networks (LCN)*. Edmonton, AB: IEEE, pp. 335–338.

Edlira D, **Larsen B**, **Chen L**, **Kassem NE**, **Liang K and Fu S** (2022) **D4.5. Assured TC-based functionalities. Project deliverable 2022**.

Engelbertz N, **Vladislav M**, **Juraj S**, **David H**, **Nurullah E and Schwenk J** (2019) Security analysis of XAdES validation in the CEF digital signature services (DSS). *Open Identity Summit* 2019.

European-Commission (2022), Revision of the eIDAS Regulation – European Digital Identity (EUid). *2022, Available at*: https://www.europarl.europa.eu/RegData/etudes/BRIE/2022/699491/EPRS_BRI(2022)699491_EN.pdf (accessed on 4 Dec 2023).

Fett D, **Ralf K and Guido S** (2016) A comprehensive formal security analysis of OAuth 2.0. In *2016 ACM SIGSAC Conference on Computer and Communications Security*. New York: ACM, pp. 1204–1215.

Fett D, **Ralf K and Guido S** (2017) The web SSO standard openid connect: In-depth formal security analysis and security guidelines. In *IEEE 30th Computer Security Foundations Symposium (CSF)*. Santa Barbara, CA: IEEE, pp. 189–202.

Ghasemisharif M, **Amrutha R**, **Stephen C**, **Chris K and Jason P** (2018) O single sign-off, where art thou? an empirical analysis of single sign-on account hijacking and session management on the web. In *USENIX Security Symposium (USENIX Security 18)*. Berkeley, CA: USENIX, pp. 1475–1492.

Groß T (2003) Security analysis of the SAML single sign-on browser/artifact profile. In *19th Annual Computer Security Applications Conference*. Washington, DC: IEEE, pp. 298–307.

Guri M (2018) BeatCoin: Leaking private keys from air-gapped cryptocurrency wallets. In *IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. Halifax, NS, Canada, pp. 1308–1316 https://doi.org/10.1109/Cybermatics_2018.2018.00227.

Haigh T, **Frank B and Ibrahim B** (2018) If i had a million cryptos: Cryptowallet application analysis and a trojan proof-of-concept. In *International Conference on Digital Forensics and Cyber Crime*. New Orleans, LA: Springer, pp. 45–65.

Hoang HD, **The DP and Van-Hau P** (2019) A security-enhanced monitoring system for Northbound interface in SDN using blockchain. In *Proceedings of the 10th International Symposium on Information and Communication Technology* (SoICT '19). New York, NY, USA: Association for Computing Machinery, pp. 197–204 https://doi.org/10.1145/3368926.3369709.

Kim BG, **Young-Seob C**, **Seok-Hyun K**, **Hyoungshick K and Woo Simon S** (2021) A security analysis of blockchain-based did services. *IEEE Access 9*, 22894–22913.

Lim SY, **Tankam FP**, **Abdullah A**, **Omar M**, **Mat KML**, **Fong AT and Reza I** (2018) Blockchain technology the identity management and authentication service disruptor: A survey. *International Journal on Advanced Science, Engineering and Information Technology 8*, 1735–1745.

Mainka C, **Vladislav M**, **Jörg S and Tobias W** (2017) SoK: Single sign-on security—An evaluation of OpenID connect. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. Paris, France: IEEE, pp. 251–266.

Malik AA, **Anwar H and Shibli MA** (2015) Federated identity management (FIM): Challenges and opportunities. In *2015 Conference on Information Assurance and Cyber Security (CIACS)*. Rawalpindi, Pakistan: IEEE.

Min Y-A (2019) A study on privilege elevation attack management for smart transaction security on blockchain etherium based system. *Journal of The Korea Society of Computer and Information 24*(4), 65–71.

Mohamed MIB, **Fadzil HM**, **Safdar S and Saleem MQ** (2019) Adaptive security architectural model for protecting identity federation in service oriented computing. *Journal of King Saud University-Computer and Information Sciences 33*, 580.

Naik N and Paul J (2017) Securing digital identities in the cloud by selecting an apposite Federated Identity Management from SAML, OAuth and OpenID Connect. In *11th International Conference on Research Challenges in Information Science (RCIS)*. Brighton: IEEE, pp. 163–174.

Naik N, **Paul G**, **Paul J**, **Kshirasagar N and Jingping S** (2022) An evaluation of potential attack surfaces based on attack tree modelling and risk matrix applied to self-sovereign identity. *Computers & Security 120*, 102808.

https://doi.org/10.1017/dap.2023.41 Published online by Cambridge University Press

Navas J and Marta B (2019) Understanding and mitigating OpenID connect threats. *Computers & Security 84*, 1–16.

Pöhn D and Wolfgang H (2020) An overview of limitations and approaches in identity management. In *Proceedings of the 15th International Conference on Availability, Reliability and Security* (ARES '20). New York, NY, USA: Association for Computing Machinery, pp. 1–10 https://doi.org/10.1145/3407023.3407026.

Preukschat A and Drummond R (2021) *Self-Sovereign Identity: Decentralized Digital Identity and Verifiable Credentials*. New York: Simon and Schuster.

Satybaldy A, **Mariusz N and Jørgen E** (2020) *Self-sovereign identity systems evaluation framework.*

Schäffner M (2020) *Analysis and Evaluation of Blockchain-Based Self-Sovereign Identity Systems*. Munich: Technical University of Munich.

Schanzenbach M (2020) *Towards Self-Sovereign, Decentralized Personal Data Sharing and Identity Managementl*. Germany: Technical University of Munich.

Shuaib M, **Mohd DS and Shadab A** (2021) Self-sovereign identity framework development in compliance with self sovereign identity principles using components. *International Journal of Modern Agriculture 10*(2), 3277–3296.

Simpson (2016) A survey of security analysis in federated identity management. In: *IFIP International Summer School on Privacy and Identity Management*. Berlin: Springer, pp. 231–247.

Soltani R, **Trang NU and Aijun A** (2021) A survey of self-sovereign identity ecosystem. *Security and Communication Networks, 1939-0114*.

Steinebach M, **Sascha Z and Katharina B** (2021) Phishing detection on tor hidden services. *Forensic Science International: Digital Investigation 36*, 2666–2817.

Talal M, **Zaidan AA**, **Zaidan BB**, **Shihab AO**, **Alsalem MA**, **Shihab AA**, **Alamoodi AH**, **Mat KML**, **Jumaah FM and Mussab A** (2019) Comprehensive review and analysis of anti-malware apps for smartphones. *Telecommunication Systems 72*(2), 1572–9451.

Tobin A and Drummond R (2016) The inevitable rise of self-sovereign identity. *The Sovrin Foundation 29*, 1–24.

van Wingerde M (2017) *Blockchain-Enabled Self-Sovereign Identity.* Master's Thesis, Tilburg University, School of Economics and Management.

W3C (2019) Verifiable credentials implementation guidelines 1.0. Available at https://www.w3.org/TR/vc-imp-guide/ (accessed on 4 Dec 2023).

Yang W, **Juanru L**, **Yuanyuan Z and Dawu G** (2019) Security analysis of third-party in-app payment in mobile applications. *Journal of Information Security and Applications 48*, 102358.

Yasuda K, **Lodderstedt T**, **Chadwick D**, **Nakamura K and Vercammen J** (2022) OpenID for Verifiable Credentials Editor's Draft available at https://ec.europa.eu/digital-building-blocks/wikis/download/attachments/600343491/Chapter_06-Open_ID_Connect.pdf?api=v2 (accessed on 4 Dec 2023).

Zhang R, **Rui X and Ling L** (2021) Security and privacy for healthcare blockchains. *IEEE Transactions on Services Computing 15*, 3668.

Cite this article: Le A, Epiphaniou G and Maple C (2023). A comparative cyber risk analysis between federated and self-sovereign identity management systems. *Data & Policy*, 5: e38. doi:10.1017/dap.2023.41