International Journal of Information Security (2023) 22:1231–1248
https://doi.org/10.1007/s10207-023-00688-w

REGULAR CONTRIBUTION

Analyzing and comparing the security of self-sovereign identity management systems through threat modeling

Andreas Grüner[1] · Alexander Mühle[1] · Niko Lockenvitz[1] · Christoph Meinel[1]

Published online: 8 April 2023

Abstract
The concept of Self-Sovereign Identity (SSI) promises to strengthen the security and user-centricity of identity management. Since any secure online service relies on secure identity management, we comparatively analyze the intrinsic security of SSI. Thus, we adopt a hybrid threat modeling approach comprising STRIDE, attack trees, and ratings towards this unique context. Data flow diagrams of the isolated, centralized and the SSI model serve as the foundation for the assessment. The evolution of the paradigms shows an increasing complexity in security zones and communication paths between the components. We identified 35 threats to all SSI components and 15 protection measures that reduce the threats' criticality. As a result, our research shows that the SSI paradigm's threat surface is significantly higher compared to the traditional models. Besides the threat assessment on model level, the adapted methodology can evaluate a specific implementation. We analyzed uPort with a restricted scope to its user agent. Thus, 2 out of 10 threats were not properly addressed, leading to potential spoofing, denial, or repudiation of identity actions.

1 Introduction

Current Identity Management Systems (IdMS) suffer from security and privacy issues [1] despite being a fundamental component of every application's security mechanisms. In particular, large centralized Identity Providers (IdP), e.g. social logins like Facebook Login,[1] accumulate an increasing amount of user data and operational statistics. Moreover, the user as the identity's embodied entity is trapped in significant trust dependencies towards the IdP [2]. To overcome these challenges, Allen [3] proposed new guiding principles for Identity Management (IdM) under the theme of self-sovereignty in 2016. The Self-Sovereign Identity (SSI) paradigm promises to undeniably bring the user back in control of its identity and remediate privacy issues [1]. The invention of general-purpose blockchains [4] delivers an implementation approach for the SSI concept and the related decentralized IdP. Based on this advancement, just as many SSI IdMS as distinct blockchain proposals have been created [5].

Threat modeling methodologies belong to the security analyst's standard repertoire [6] to ensure sufficient protection for exposed software components. An applied threat model provides structured insights into the attack surface and allows the evaluation of countermeasures. Threat modeling can significantly increase the security posture of a system [7]. Instead of having a single comprehensive methodology, a wide variety of approaches has been developed over time. They provide generic techniques to evaluate attack vectors comprehensively. However, security researchers adapt them to the specific system for evaluation.

In this paper, we develop a systematic approach to evaluate the security of a blockchain-based SSI IdMS on the implementation and model level. Data Flow Diagrams (DFD) [8] of the isolated and centralized IdM scheme and the SSI pattern based on blockchain serve as starting point. We apply to the DFDs a version of Potteiger's [9] hybrid threat modeling methodology that combines STRIDE [6], attack trees [10], and the Common Vulnerability Scoring System version 3 (CVSSv3) ratings [11]. STRIDE is a systematic approach

[1] https://developers.facebook.com/docs/facebook-login/

✉ Andreas Grüner
andreas.gruener@hpi.uni-potsdam.de

Alexander Mühle
alexander.muehle@hpi.uni-potsdam.de

Niko Lockenvitz
niko.lockenvitz@hpi.uni-potsdam.de

Christoph Meinel
christoph.meinel@hpi.uni-potsdam.de

[1] Hasso Plattner Institute (HPI), University of Potsdam, 14482 Potsdam, Germany

to identify threats to IT systems developed. Together, all elements enable a holistic security review whilst providing a quantitative threat priority score to address risk-driven defense measures. We investigate the threat surface for the isolated, centralized, and SSI IdM scheme and compare the schemes' threat exposure. The result indicates differences on IdM model level.

Moreover, we practically apply the threat methodology to uPort [12] and evaluate its security posture. Thus, we show the benefit of our developed approach for increasing the security of a specific SSI IdMS. In our contribution, we address in particular the following research questions:

1. **RQ1:** Does SSI achieve higher security based on the threat surface than the isolated or centralized IdM model?
2. **RQ2:** Which threats exist for a blockchain-based SSI IdMS and how can they systematically be evaluated?

Related work has only a specific focus on particular security attacks for dedicated implementations. Our contribution enables the comparison on model level and the comprehensive evaluation of dedicated systems.

The remainder of the paper is organized as follows. In Sect. 2, we present related work to our contribution. Subsequently, in Sect. 3, we outline background knowledge regarding threat modeling and SSI. We propose our threat analysis in Sect. 4 and conduct the security analysis of uPort in Sect. 6. Additionally, we discuss our results in Sect. 7, and provide insights into future research directions in Sect. 8. Finally, we conclude in Sect. 9.

2 Related work

Related research work addresses solely partially security considerations for dedicated components of SSI IdMS. There exist extensive surveys about blockchain security. Conti et al. [13] elaborate in detail about attacks and countermeasures on Bitcoin [14] and its Proof-of-Work consensus protocol. The researchers describe double-spending and wallet, network, and mining attacks. Li et al. [15] published a broader analysis by considering the security of blockchain in general. The authors investigate the 51% vulnerability, private key security, criminal activity, double spending, transaction privacy leakage, criminal and smart contract vulnerabilities.

Additionally, the researchers outline real cases, e.g. the Decentralized Autonomous Organization (DAO) attack [16]. Shaharir et al. [17] describe comprehensively the flow of attacks against blockchain systems by using Petri Nets [18] following the STRIDE methodology.

Besides these publications, various authors examined specific attacks on SSI IdMS. Dingle et al. [19] explore techniques of a malicious verifiable credential holder. Allen et al. [20] investigate the security of the distributed ledger, data access, and private key management. Stöcker et al. [21] consider the impact of quantum computing on the security of SSI.

Moreover, Stokkink et al. [22] present and evaluate their own SSI IdMS regarding denial of service and Sybil [23] attacks. Additionally, Alexopolous et al. [24] analyze the benefits of SSI IdMS towards traditional IdM models. The authors concluded that the use of blockchain could prevent stealthy target, double registration, stale information, denial of service, and censorship attack.

Kim et al. [25] examine the security of SSI IdMS with a specific focus on the implementation of Hyperledger Indy [26]. Within the analysis, the SSI IdMS is clustered in different compartments that are analyzed for its threat surface using DFDs. Furthermore, researchers conducted threat modeling for traditional IdMS. Ahmad et al. [27] and Khattak et al. [28] examine threats to federated identity schemes. Additionally, Dominicini et al. [29] investigate identity threats focusing on the mobile internet.

In contrast, our research systematically investigates the security of SSI IdMS and its single components based on a hybrid threat modeling approach. Moreover, we show practicality by analyzing a component of a specific SSI IdMS and comparing the IdM models.

3 Background

In this section, we briefly introduce threat modeling techniques (Sect. 3.1), traditional IdM patterns (Sect. 3.2), and blockchain-based SSI (Sect. 3.3). Additionally, we describe DFDs of the IdM schemes as a foundation for the threat analysis.

3.1 Threat modeling

Attacks represent the intentional exploitation of a weakness by an adversary. Moreover, threats additionally encompass the unintentional use of a flaw leading to a more comprehensive consideration. Thus, researchers developed various threat modeling techniques [30].

Among the methodologies, STRIDE is widely used [17]. Furthermore, attack trees allow a simple yet powerful way to structurally model threats. Additionally, we see the advantages of CVSSv3 for having a criticality score of a threat despite criticism of its subjectivity [31]. Potteiger et al. [9] build a hybrid threat modeling approach by combining these methodologies.

We apply this technique to SSI IdMS. Thus, we concisely introduce the approach. The evaluated application is depicted in the system model. The system model comprises components, communication flows, and an overall graph for each

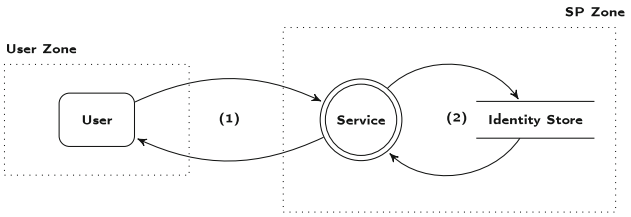

Fig. 1 Isolated IdM data flow diagram

STRIDE category. STRIDE abbreviates Spoofing (*S*), Tampering (*T*), Repudiation (*R*), Information Conflict of interest (*I*), Denial of Service (*D*), Elevation of Privilege (*E*). Per system component, a separate model encompasses attribute templates and STRIDE attack trees.

The component attack trees form the basis for the system attack graph. The attribute templates are properties of the component. An attack tree node reflects a threat or a mitigation measure. Each threat node is annotated with an attack score based on CVSSv3. A mitigation node is extended by a risk reduction value. Threat probabilities propagate from leaf nodes to the root via intermediary vertices.

The CVSSv3 score combines the rating of a base, temporal and environmental score on a scale between 0 and 10, whereof we solely apply the base value to reduce complexity. We use the NIST Calculator[2] to determine the rating.[3]

3.2 Traditional identity management models

Traditional IdM models differentiate the user, the IdP, and the Service Provider (SP) as actors. We assume a password-based authentication method because it is still most prevalent on the Internet [32].

Figure 1 depicts the DFD of the isolated IdM setting. In this scheme, the IdP is part of the SP's service and it is no distinct entity. Thus, the DFD considers a user and a SP zone. The user interacts with a service that requires authentication (1). Verification information for user credentials is stored in an identity store that is accessible by the service (2).

In the centralized IdM model, the IdP is an independent entity that provides IdM services to the user and the SP. Thus, the IdP spans a separate zone in the DFD (see Fig. 2). The IdP owns the identity store. Upon authentication request of a user at the SP (0), the user is redirected to the IdP. The IdP verifies the presented authentication credential (1) against der identity store (2) and returns the result to the SP (3). The SP grants access to the user if the process is successful. Otherwise, the user's admission is denied.

[2] https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator

[3] We outline additional details on the components of the CVSSv3 metric composition in the Appendix 1.

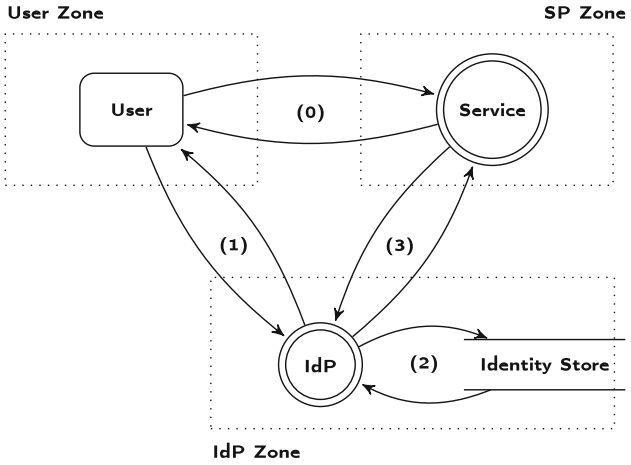

Fig. 2 Centralized IdM data flow diagram

3.3 Self-sovereign identity

The traditional actors changed in the SSI paradigm. We differentiate the identity holder, the verifier, and the issuer. Furthermore, distinct implementation variants realize the SSI scheme. The predominant solution applies a blockchain. Therefore, a decentralized IdP is implemented on a blockchain network. We focus on this SSI setting. Figure 3 outlines the corresponding DFD.

The identity holder represents the user and spans a separate zone. The user manages its identity with the user agent. The user agent has access to a Verifiable Claim (VC) [33] store. A VC represents an issuer-attested attribute of the user. Moreover, a lightweight node facilitates the interaction with the decentralized IdP. This node stores only a minimal subset of the blockchain data to verify transactions. Additionally, it enables the communication to the decentralized IdP, for instance, to register an identifier. A mobile app is the primary implementation variant of a user agent.

The verifier personifies the SP. The identity holder presents its identifier and VCs to the verifier upon intended service consumption. The SP verifies the VC's signature and validity. Thereby, the verifier uses an organizational (org) agent. The org agent mediates the communication between the service and the user agent. Furthermore, it interacts with the decentralized IdP via a node. Additionally, the org agent communicates with a trust store containing trusted issuers. A VC originating from a trusted issuer is accepted as an attribute of the identity.

The third party is the issuer. In the centralized model, the IdP or AP can be compared to it. The issuer attests properties of the user. Thus, it provides the VCs. The issuing process interacts with an issuer-owned instance of the org agent. The org agent accesses various data stores to retrieve and verify VC values. Moreover, the interaction path with the node allows the rooting of the VC on the decentralized IdP.

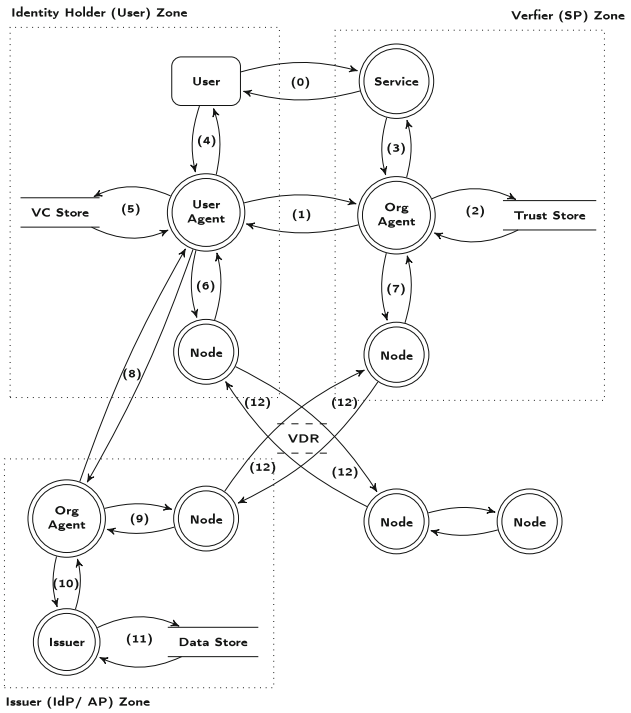

Fig. 3 Blockchain-based SSI data flow diagram

The decentralized IdP is implemented on a blockchain. The blockchain is a consecutive sequence of blocks [34]. Each block contains transactions and a cryptographic hash of its predecessor. The hash of the predecessor securely links the blocks to a chain. A transaction may include a token transfer, smart contract creation or execution, or further computations. A peer-to-peer network propagates the blocks and transactions to each other. The network nodes compete with each other to build the next block. Within the Proof-of-Work scheme, a node tries to solve a computationally difficult puzzle. The result is part of the new block and defines the competition's winner.

The blockchain serves as the execution platform for the decentralized IdP. This IdP is comprised of a Verifiable Data Registry (VDR). The VDR provides an identifier registry, or additionally a verifiable claim registry [35]. We assume the latter one because it provides for each VC a verifiable timestamp of existence and revocation. Here the existing identifiers and VCs are marked.

The VDR can either be implemented as smart contracts or as a blockchain itself. Nodes in the blockchain network process changes to the VDR as transactions. A transaction can carry, for instance, the registration or revocation of identifiers. Furthermore, the modification of VCs are conducted by transactions. The transactions are compiled to blocks by the nodes according to their consensus protocol. The processing of these connected blocks lead to the series of changes and the current state of the VDR.

The decentralized IdP does not have a separate security zone. However, each node of the blockchain network executes the IdP routines and, therefore, supports the VDR. Each entity that hosts a node is part of the blockchain network. The decentralized IdP and the network is bound to the security guarantees of the blockchain consensus protocol.

Furthermore, blockchains intrinsically apply public-key cryptography. Thus, it determines the authentication method. Before the user can start an authentication process, it registers an identifier on the VDR and interacts with the issuer's org agent via the user agent (8). The issuer queries its data store for the VC value (11) and issues the attestation (9, 10). Furthermore, the issuer anchors the VC in the VDR (9).

Subsequently to this preparation, communication with the SP can commence. The user opens the respective service (0) that requires authentication and selects the SSI-based method. Afterward, the user runs its user agent (4) to interact with SP's org agent (1). In detail, the user selects required VCs and provides them to the SP. The SP verifies the issuer against the trust store (2) and checks the signature of the VC to detect manipulations. In particular, the control of the VC's associated identifier must be verified. Furthermore, the validity is affirmed with the VDR via the SP's node (7). In case sufficient VCs from trusted issuers are received, service consumption of the user initiates.

4 Threats on self-sovereign identity

In this section, we first elaborate on security objectives in relation to the STRIDE categories (Sect. 4.1) and describe the adversary model (Sect. 4.2). Moreover, we outline the component attack models for each identified module in the SSI DFD (Sects. 4.3–4.8). Within the description, we concentrate on the threats and their impact on dedicated SSI factors. For each threat, we present the impact rating and list countermeasures with a mitigation score. We concentrate on technical aspects and do not consider adversarial actions against actors (e.g. blackmailing).

4.1 Security objectives

The CIA triad, comprising Confidentiality, Integrity and Availability, defines well-known security objectives. The STRIDE categories group threats against these security goals. Confidentiality is endangered by spoofing, tampering, elevation of privileges and unwanted information disclosure. Integrity is vulnerable by tampering, repudiation and elevation of privilege threats. A denial of service attack targets availability.

4.2 Adversary model

We assume a computationally bounded adversary [24] that tries to attack the SSI IdMS for violating any of the security objectives. Thereby, our analysis is independent of the actual position of the adversary. We can assume an internal adversary that is a participant of an interaction and can be either the user, the SP or the issuer. Despite that, external attackers outside an interaction have an interest in the exploitation of a threat.

4.3 User agent

We start with the descriptions of the user agent's threats. Thereby we assign a unique number to each threat and countermeasure that is referenced by re-occurrence at another STRIDE category or component. The user agent enables the identity holder to interact with its identity.

4.3.1 (S) Spoof identity actions

At the user agent, the category spoofing refers to the illegitimate execution of identity actions. This mainly references authentication, disclosure of credentials and authorization when using the identity. The following threats enable the take-over of the identity.

- Acquire Private Key (T1): Access to the identity is granted by a self-authenticating scheme based on a private key. To obtain control, the adversary acquires the private key. The impact has a severe score of 8.7 based on the high confidentiality impact. As countermeasure, the private key should be stored in a secure enclave where it is non-extractable (C1). This raises the attack complexity and required privileges. The defence measure reduces the rating by 1.3.
- Steal or Covertly Access User Agent Device (T2): A device hosts the user agent and stores the private key. A smartphone reflects such a device with a user agent. The rating is comparable to T1 with a score of 8.7. Countermeasures comprise access protection of the device (C2) and a remote revocation process (C3) to remediate the impact and to make the device/ identity unusable. Defence measure C2 preventively reduces the score by 1.3 based on an increased complexity and demanded access level. The use of C3 may limit the damage after the exploitation of the threat has been detected.
- Exploit Recovery Mechanism of Identity (T3): A recovery mechanism allows the legitimate identity holder to restore the identity. For instance, it is necessary in case of losing the device. If the recovery mechanism is flawed, an adversary may exploit the weakness and take-over the identity illegitimately. The rating is 9.9 based on the network attack vector. Common platform security measures form the protection.

4.3.2 (T) Tampering with the user agent's data

The user agent's data comprises the private key that controls the identifier. We considered the disclosure of the key already (cf. 4.3.1). Additionally, the VC store covers the VCs. Thus, eventually remaining uncritical data poses no further threats.

4.3.3 (R) Repudiate identity actions

The category refers to the legitimate execution of actions by a user. Afterwards, the user deliberately denies these actions to gain an advantage. Thus, the user repudiate actions that she/ he has done.

- Deliberately Disclose Private Key (T4): The private key authenticates the identity. The user could deliberately disclose this private key to deny any actions. The rating is 8.7 due to the high impact on integrity. As countermeasure, the private key should be stored in a secure enclave (C1). This prevents the extraction of the key and reduces the score by 1.3.
- Revoke Identity (T5): The identity holder executes actions and later on revokes the identity. In case no timestamped order exists, the user can repudiate the actions. The rating is 9.1 due to the high impact on integrity. Timestamping (C4) reduces the score by 1.2 based on increased complexity and privileges.
- Deliberately Loose User Agent Device (T6): The identity holder accesses a service. Subsequently, the user pretends to have lost the device and strives for reimbursement. The impact rating is 9.1 due to lost integrity. As countermeasure, access protection of the device (C2) reduces the rating by 1.3. Additionally, the remote revocation procedure (C3) should be used in a timely manner.

4.3.4 (I) Reveal confidential identity information

Comparable to data tampering, the user agent protects solely the private key. Furthermore, the VC store captures the confidential VC data (cf. 4.4). Thus, breach of confidentiality is not applicable for this component.

4.3.5 (D) Deny identity actions

The class encompasses threats to deny the usage of the identity. As a result, the identity holder is prevented from accessing services.

- Steal or Break User Agent Device (T7): The device is fundamental to operate the user agent and the identity. In

case the device is not available to the user, no interaction can commence. The impact is 5.2 due to a physical attack vector and high availability impact. A recovery mechanism (C5) serves as countermeasure to regain access to the identity after threat exploitation.
- Delete Private Key (T8): The private key enables the user to access and interact with its identity. The deletion of the private key leads to denial of service. The rating is comparable to T8 and reflects 5.2. Device access protection (C2) and a recovery procedure (C5) are countermeasures. C2 decreases the rating by 1.3 due to increased complexity and privileges.
- Exploit Identity Revocation (T9): An authorized user can initiate the revocation procedure to disable the identifier. This may be the case to securely abandon an old identity. Exploiting the procedure lead to denial of service. The impact rating is 7.7 due the network attack vector. As a countermeasure, the revocation procedure must properly authenticate entitled users (C6). This reduces the score by 1.9.

Additionally, the exploitation of the recovery mechanism (T4) are applicable to the denial of service category. General platform defence measures (cf. 4.3.1) apply.

4.3.6 (E) Elevate privileges on the user agent

The user agent manages the identity of a single user. Therefore, no distinct privilege levels exists to differentiate access. Privilege elevation attacks are not applicable.

4.4 VC store

The VC store comprises all VCs of the identity holder to be available for a disclosure request.

4.4.1 (S) Spoofing VCs

The category encompasses the illegitimate creation of a VC. The threat can originate within the VC Store or in the security zone of the issuer.

- Create Self-attested Claim (T10): The user creates a self-attested claim comprising a wrong value. In case the SP relies on the claim value, it may lead to an illegitimate service consumption. The impact is 9.6 due to the network attack vector. The verifier must check the issuer of the claim (C7). This defence measure reduces the rating by 1.4 due to increased attack complexity.

4.4.2 (T) Tampering with the VCs

The class comprises attacks to manipulate the VC. The adversary adjusts an already issued claim in the VC Store to gain a benefit.

- Change VC Value (T11): The user manipulates the value of an issued claim. The new value enables unjustified service consumption. The rating is 8.4 with local attack vicinity and low complexity. The verification of cryptographic signatures of the VC (C8) serves as defence measure. It reduces the score by 0.9 based on increased attack complexity.

4.4.3 (R) Repudiate VC issuance

The category contains threats for the repudiation of claims. A negative claim for the user might be repudiated by the user itself. Additionally, the issuer may deny a positive attribute despite its validity.

- Delete VC (T12): The identity holder can delete a VC which is in its possession to repudiate its issuance. Then, the identity holder can credibly deny the publishing. The threat score is 6.5 due to the high impact on integrity and local attack vector. The VC registry model approach (C9) is a defence measure. It reduces the rating by 1.2 based on increased attack complexity and required privileges.

4.4.4 (I) Reveal confidential VC information

The category comprises threats to derive illegitimately confidential information from VCs.

- Gain Unauthorized Access (T13): An adversary circumvents access controls on the VC Store and obtains confidential VC data. The threat score is 7.7 based on the network attack vector and the high impact on confidentiality. General platform security measures, including tested access controls, form the protection.
- Request Unnecessary Data (T14): A verifier may request extensive or not required attributes during a VC disclosure request. This behaviour reveals confidential information. The impact rating is 7.7 due to the network attack vector and the high impact on confidentiality. VC disclosure based on zero knowledge proofs limit the revealed information but does not protect against superfluously requested data.

4.4.5 (D) Deny VC store serviceability

The threats in this category target the availability of the VC store for requests to retrieve and store newly issued claims.

Attacks in this category refer to claim deletion (T12). They apply with the corresponding countermeasures.

4.4.6 (E) Elevate privileges on the VC store

The VC store can be implemented through distinct options. A local storage within the user agent may not differentiate privilege levels for various users. In contrast, a cloud or decentralized storage requires different access levels. However, threats and countermeasures are not specific to SSI.

4.5 Organizational agent

The org agent's functionality is comparable to the user agent. In contrast, it is not hosted on a single user device. Additionally, the org agent serves an organization and, therefore, several persons use it. However, the identity holder remains the organization.

4.5.1 (S) Spoofing identity actions

The category comprises threats to spoof identity actions. These behaviours are not in the interest of the identity holder and may result in liabilities for it.

– Misuse identity (T15): In case a single person controls a corporate identity, executed actions might not be in the most interest of the organization. The actions can be more beneficial for the controlling entity. 9.6 is the threat rating based on the network attack vector as well as the high impact on confidentiality and integrity. A split control scheme (C10) distributes the responsibility and serves as defence measures. It reduces the rating by 1.9 due to increased attack complexity and required privileges.

Additionally, the following threats for the user agent apply likewise (cf. 4.3): acquire private key (T1) and exploit recovery mechanism (T3).

4.5.2 (T) Tampering with the org agent's data

The organizational agent comprises configuration data, e.g. approval schemes. Tampering with this data might impose security threats.

– Manipulate Configuration (T16): An adversary within the organization manipulates the configuration data of the org agent. For instance, security measures as a split control scheme might be deactivated. In consequence, the threat surface increases. The impact score is 6.5 based on a medium impact on the security objectives and the adjacent network attack vector. The latter reason assumes an insider with access to the corporation. General access controls and platform security measures increase protection. Furthermore, an audit trail (C11) enables a compliant post-mortem analysis.

4.5.3 (R) Repudiate identity actions

Users rely on actions of the issuer and the verifier. An illicit repudiation of conducted actions undermines trustful communications.

– Illegitimate VC Revocation (T17): Issued VCs are the foundation for service consumption by the user. The user trusts the issuer that the VC remains valid based on agreed terms. A single person of the issuer might illegitimately revoke the VC. The rating is 6.5 due to the medium impact on the security objectives. A split control scheme (C10) reduces the score by 1.4 due to the increased attack complexity and privilege level.

Furthermore, the deliberate disclosure of the private key (T4) and identity revocation (T5) including associated protection measures are applicable.

4.5.4 (I) Reveal confidential identity information

Comparable to the user agent, the org agent protects solely the private key. Furthermore, the trust and data store captures the confidential data (cf. 4.6). Thus, this threat category is not applicable for the component.

4.5.5 (D) Deny identity actions

A non-usable identity prevents service provisioning and affects all actors. In this category, comparable threats as for the user agent exists (cf. 4.3). These threats encompass the deletion of the private key (T8) and the illicit use of identity revocations (T5) as well as their associated countermeasures.

4.5.6 (E) Elevate privileges on the org agent

The org agent differentiates privilege levels to allow different roles. For instance, a split control scheme requires different entitlements to collaborate. If an individual might obtain a higher privilege level, it may illicitly execute actions.

– Take-over Role (T18): An individual takes over a higher privileged role and executes actions. The impact is 7.1 based on the high impact on confidentiality and integrity. The assignment of the roles should follow a split control scheme (C10) as countermeasure. Due to increased complexity and demanded access rights, the score is reduced by 1.4.

4.6 Trust and data store

The trust store contains the verifier's trusted issuers. This list determines the VC issuer that are accepted during a disclosure process. The data store comprises locations that comprise the base data for the issued VCs.

4.6.1 (S) Spoofing trusted issuers or VC data

This category does not encompass any threats. We map potential attacks to the tampering category because they are not discriminable.

4.6.2 (T) Tampering with trusted issuers or VC data

The category comprises any threats to tamper with the trusted issuers or underlying VC data. A manipulated list or changed base data for the VC allows fraud at the side of the SP.

- Circumvent VC Verification (T19): An adversary may circumvent the VC verification procedure by exploiting the process or manipulating verification data. Thus, the adversary holds an illegitimate claim. The VC enables the adversary to consume services. The impact is 7.1 due to the breach of integrity and availability. General platform security measures protect from this threat.
- Manipulate Trusted Issuers (T20): An adversary may manipulate the verifier's list of trusted issuers. As consequence, untrusted and potential malicious issuers become trusted. Thus, the adversary might consume a service with a forged VC. The rating is 5.5 based on the high impact on integrity. Common platform security measures apply.

4.6.3 (R) Repudiate trusted issuers or VC data

Repudiation threats are inapplicable because the store components do not provide repudiable actions.

4.6.4 (I) Reveal confidential issuer/ verifier information

The disclosure of information of the trust store or data store enables the adversary to gain an advantage for further attacks.

- Disclose Trusted Issuers (T21): The disclosure of issuers lead to an information advantage. It enables the adversary to specifically attack a weakness to consume the service under false pretences. The rating is 3.3 due to the low confidentiality impact. Common security measures form the protection.

4.6.5 (D) Deny store serviceability

The threats in this category target the availability of the store itself to prevent serving the stored data. Threats and protection measures are non-SSI-specific.

4.6.6 (E) Elevate privileges on the store

Access to the trust or data store require different privilege levels. Comparable to the denial of service category, threats and defence approaches are not specific to SSI.

4.7 Identity holder/ verifier/ issuer node

The nodes build the blockchain network to form the decentralized IdP with the VDR. We assume that each actor has a node in its security zone according to the DFD (cf. 3.3). Another approach is the use of an external node. However, an additional threat zone would be introduced. The nodes communicate with messages to identify peers as well as receive and propagate transactions and blocks.

4.7.1 (S) Spoofing node messages

The category comprises the spoofing of messages to mislead the node of the identity holder, verifier or issuer. As the blockchain network consists of peers, there is no central verification authority for the data.

- Propagate Forged Message (T22): An adversary may isolate the node by manipulating the known neighbour nodes. Additionally, forged transactions or complete blocks might be propagated to the attacked node. This may result in a different processing state and leads to disparate content of the VDR. For instance, identifiers or VCs might be presented as valid or revoked on the discretion of the adversary. The threats rating is 6.5 due to the network attack vector and the high impact on integrity. Independent blockchain network monitoring (C11) protects from this threat and reduces the score by 1.2 due to increased attack complexity.

4.7.2 (T) Tampering with the node

The node preserves the state of the blockchain network to support the VDR. Requests of the issuer, service or identity holder processes are responded against the actual state of the node. In case the node is manipulated, the responses are disguised.

- Manipulate State (T23): An adversary directly manipulates the internal state of the node to gain benefits

comparable to the spoofing category (cf. 4.7.1). The rating is 5.5 due to the local attack vector and the high impact on integrity. Separate blockchain network monitoring (C11) uncovers state manipulations and reduces the rating by 1.2 due to increased complexity. Furthermore, common platform security measures protect the node's state.
– Manipulate Configuration (T24): An adversary may manipulate the configuration of the node to enable other attacks. The impact is 5.5 due the local attack vector and the high impact on integrity. Common platform security measures form the protection.

4.7.3 (R) Repudiate node messages

The peers apply a low level communication protocol to exchange messages. A repudiation of messages are commonly not part of a gossip protocol for a peer-to-peer network.

4.7.4 (I) Reveal confidential node information

The node stores the public state data of the blockchain network. There is no additional confidential information to be disclosed. Thus, no threats exist in this category.

4.7.5 (D) Deny node serviceability

The category comprises threats to deny availability of the node. In case the peer is unavailable, proper verification of the identifier and VCs are not possible.

– Reset or Close Connections (T25): An adversary may send forged messages the target node to reset or close established connections to neighbour peers. Thus, the state is preserved to the current version and does not receive updated information. For instance, new revocations or added VCs are not known to the node. The impact is 6.5 due to the network attack vector and high impact on availability. Message sender verification (C12) protects against this threat. The defence measure reduces the score by 1.2 due to increased attack complexity.
– Flood Connections (T26): An adversary floods all connection slots of a node. As a consequence, the peer is not able to establish new connections and is restricted in its communication. The impact is 6.5 due to the network attack vector and high impact on availability. Comparable to the previous threat, message sender verification (C12) serves as protection and reduces the score by 1.2.

4.7.6 (E) Elevate privileges on the node

A node does not differentiate privilege levels. Furthermore, general threats and security measures are not specific to SSI.

4.8 Verifiable data registry

The VDR serves as identifier and claim registry. It is a decentralized single point for verification of their validity.

4.8.1 (S) Spoofing VDR entries/ (T) tampering with the VDR/ (R) repudiate VDR entries

The threat categories spoofing, tampering and repudiation comprise a similar attack vector. The attack enables the manipulation of the VDR and impacts the integrity of the stored data.

– Exploit Smart Contract Vulnerabilities (T27): An adversary exploits vulnerabilities in the smart contract or the blockchain of the VDR. Such a vulnerability may allow the attacker to register or revoke identifiers and VCs. Thus, the adversary interferes with identity communication processes. The DAO attack [16] is an example for generally exploiting smart contract vulnerabilities. The impact is 6.5 due to the network attack vector and the high impact on integrity. As countermeasure, the smart contract or blockchain code must be scanned for vulnerabilities (C13). This defence measure reduces the score by 1.3 due to the increased attack complexity.

4.8.2 (I) Reveal confidential VDR information

Comparable to the threat analysis of the node in this category (cf. 4.7.4), the VDR stored data is public available. Therefore, threats to disclose confidential information are not applicable.

4.8.3 (D) Deny VDR serviceability

This category encompasses threats that render the VDR unusable. A non-available VDR leads to a non-functioning of the decentralized IdP. Thus, validity of objects and revocations cannot be verified.

– Deactivate VDR Smart Contract (T28): A smart contract can be deactivated by the owner. In case an adversary illegitimately disables the VDR, the complete SSI solution is rendered unusable. The score is 6.5 due to network attack vector and high impact on the availability. As countermeasure serves a vulnerability scan (C13) to prevent flaws leading to unauthorized deactivations. It decreases the rating by 1.2 due to increased attack complexity.

– Manipulate Blockchain Configuration (T29): In case a permissioned blockchain realizes the VDR, the privileged peers are able to change the configuration. An adversary may change the configuration to deny service. The rating is 9.6 due to the network attack vector and an high impact on availability and integrity. A split control scheme (C10) serves as protection measure.

Furthermore, the exploitation of vulnerabilities (T27) apply similarly with the according countermeasures.

4.8.4 (E) Elevate privileges on the VDR

The VDR realized by a smart contract or independent blockchain must support different privileges to distinguish entities The take-over of a privilege that is not intended for an individual poses a threat.

– Take-over VDR Owner Role (T30): The VDR owner role enables a user to change the VDR smart contract or the underlying blockchain network. In case an adversary takes-over the role, illegitimate changes can be the consequence. The rating is 7.7 due to the network attack vector and the high impact on integrity. Countermeasures are the vulnerability scan (C13) and the split control scheme (C10). They reduce the rating by 1.9 due to increased complexity and demanded privileges.
– Take-over Identity Holder Role (T31): The identity holder has the privilege to register new identifiers or to revoke existing identifiers. An adversary that takes-over the role may illegitimately revoke an identifier and cause unavailability. The impact is 7.7 due the network attack vector and high impact on availability. A vulnerability scan (C13) mitigates potential flaws. C13 reduces the rating by 1.4 due to increased attack complexity.
– Take-over Issuer Role (T32): An issuer can add new VCs or revoke existing VCs. An adversary that takes over a certain issue role may misuse these privileges leading to spoofing, repudiation or denial of service. The impact is 9.6 due to high impact on integrity and availability. Comparable to the previous elevation threat, a vulnerability scan (C13) reduces the rating by 1.4 due to increased attack complexity.

4.9 Communication channels

The various components interact with each other across the security zones and require communication channels. These paths are partially covered in the analysis of the node (cf. 4.7). Further communication paths apart from the node interaction is covered in this section.

4.9.1 (S) Spoofing/ (T) tampering with the communication

These categories comprise threats that spoof or tamper with the communication between the components. Thus, messages are illegitimately altered or blocked between the interaction endpoints to gain an advantage for the attacker.

– Spoof Communication Partner (T33): An adversary may act as a legitimate communication endpoint. Thus, another entity may falsely communicate with the faked endpoint. The result can lead to unwanted disclosure of information or the obtainance of wrong VCs. The rating is 9.6 due to the high impact on integrity and confidentiality via the network. As countermeasure serves communication partner verification. This reduces the score by 1.4 due to increased complexity.

4.9.2 (R) Repudiate communication

The category encompasses threats where the sender is able to repudiate sent communication. For instance, the sender can deny that it has send a message. An adversary may sent messages and later on dispute the communication.

– Dispute Message (T34): The adversary may communicate with an communication partner, for instance, to order a good during service provisioning. Later on, the adversary may dispute to conducted the communication. The score is 7.7. due to the high impact on integrity. As countermeasure serves message verification. This reduces the rating by 1.4 due to increased attack complexity.

4.9.3 (I) Reveal confidential information

The category comprises threats to disclose confidential data that is transported during the communication. Within the communication personal and further information is transported between the various components.

– Traffic Interception (T35): The adversary listens to the communication channels between the entities. Based on the communication, the adversary learns confidential attributes of the user and further data. The impact is 7.7 due to the high confidential impact and the network attack vector. As countermeasure serves communication encryption (C15). It reduces the rating by 1.4.

4.9.4 (D) Deny communication

The denial of communication leads to a denial of service. Comparable to the node components (cf. 4.7), the threats of resetting (T25) or flooding (T26) connections and their associated countermeasure (C12) apply.

4.9.5 (E) Elevate privileges in the communication

The category of elevating privileges is not applicable for communication channels due to non-existence in the communication process.

4.10 Summary

Tables 1 and 2 provide an overview of the described threats for the SSI model. We could identify overall 35 threats with 15 associated countermeasures. The listed protection measures mitigate the threats. In the tables, the—represents the non-existence of SSI-specific threats or countermeasures.

Nonetheless, general IT threats and security measures still apply for the components. Furthermore, n/a marks not applicable threat categories for a certain component and, therefore, a non-existent threat surface.

The user agent, org agent, and the VDR exhibit the highest quantity of threats. In particular, the user and the org agent are vulnerable regarding the spoofing, denial and repudiation of identity actions. For spoofing identity actions, the adversary takes over the control of the identifier. To achieve this objective, the attacker can acquire the private key (T1), steal the device (T2), misuse the identity (T15) or exploit the identity recovery mechanism (T3). The deliberate disclosure of the private key (T4) or device loss (T6), identity revocation (T5), and illegitimate VC revocation (T17) represent the repudiation threats. A denial of identity actions is predominantly caused by deleting the private key (T8), exploiting identity revocation (T9) and exploiting the recovery mechanism (T3). As security measures, the usage of a secure enclave (C1) for the private key, device access protection (C2), and a recovery procedure (C3) for the identity are essential.

Furthermore, there are threats that cannot be mitigated with SSI-specific countermeasures. Exploit recovery mechanism (T3), gain unauthorized access (T13), circumvent VC verification (T19), manipulate trusted issuers (T20), disclose trusted issuers (T21) and manipulate configuration (T24) belong to these threats. However, general platform security measures apply.

5 Comparison of identity management models

In Sect. 3, we presented the DFDs of the isolated (see Fig. 1), the centralized (see Fig. 2), and the SSI model (see Fig. 3). The number of security zones between the models increases from 2 in the isolated to 3 in the centralized and the SSI paradigm. Furthermore, the quantity of components significantly elevates from 3 in the isolated setting to 4 in the centralized model and to 12 in the SSI scheme. Likewise, the number of communication channels raises as well.

Table 3 presents the relevance of the analyzed threats to the defined IdM models. A • indicates an applicable threat. The ○ reflects a threat that is analogously suitable depending on the actual implementation. The isolated and the centralized model might use a password-based authentication, but can also integrate another authentication scheme. In contrast, the implementations of the SSI paradigm use private key cryptography. The represents no applicability of the threat.

The 35 previously described threats affect the SSI model and their implementations. Thus, a security review of an SSI IdMS demands mitigation of them. Moreover, a sub set of 28 threats are applicable to the centralized model and 17 threats are relevant in the isolated setting. For both last schemes, 7 threats depend on the actual implementation of the paradigm. These threats encompass acquire (T1) or deliberately disclose (T4) the private key, steal or covertly access the user agent device (T2), deliberately loose or break the user agent device (T7) and delete the private key (T8). These attack vectors can be transferred to a password-based authentication system. Moreover, the change of VC values (T11) belongs to this category.

Overall considering the analyzed attack vectors, the threat surface is significant higher in the SSI paradigm compared to the centralized and isolated scheme. Nonetheless, certain threats are common attack vectors for all schemes. In particular the used authentication method determines potential attacks. In this regard, the SSI scheme generally uses a self-authenticating method based on public key cryptography. This originates from its decentralized nature and the use of blockchain. In contrast, isolated and centralized model implementations tend to use password-based schemes which are widely considered as less secure. Thus, regarding the used authentication scheme the SSI model is advantageous.

6 Security analysis of uPort

uPort [12] is one of the earliest developed SSI IdMS based on Ethereum [4]. uPort comprises a user agent with integrated VC store, libraries to build an org agent or to integrate directly into applications and a set of smart contracts that form the VDR. The smart contracts encompasses the controller and the proxy contract.

The controller contract establishes the self-authenticating scheme. The proxy contract abstracts the control from the identifier. Furthermore an application contract ensures the decentralized nature of the provided service. We concentrate our security assessment on the user agent as it is the most critical component encompassing the highest number of threats.

The uPort user agent is available in the regular app store on the iOS platform. The user is able to generate a new identifier after the installation. According to the white paper [12], the private key of the identifier is stored in the secure enclave

Table 1 SSI component threats, impact and countermeasures (1/2)

Component	STRIDE Category	Threat	Rating	Countermeasures	Mitigation
User agent	Spoof identity actions	Acquire private key (T1)	8.7	Secure enclave (C1)	1.3
		Steal or covertly access user agent device (T2)	8.7	Device access protection (C2)	1.3
				Remote revocation process (C3)	–
		Exploit recovery mechanism (T3)	9.9	–	–
	Tampering with user agent's data	n/a	n/a	n/a	n/a
	Repudiate identity actions	Deliberately disclose private key (T4)	8.7	Secure enclave (C1)	1.3
		Revoke identity (T5)	9.1	Action timestamping (C4)	1.2
		Deliberately loose user agent device (T6)	9.1	Device access protection (C2)	1.3
				Remote revocation process (C3)	–
	Reveal confidential identity inf	n/a	n/a	n/a	n/a
	Deny identity actions	Steal or break user agent device (T7)	5.2	Recovery mechanism (C5)	–
		Delete private key (T8)	5.2	Device access protection (C2)	1.3
				Recovery procedure (C3)	–
		Exploit identity revocation (T9)	7.7	Authenticate entitled users (C6)	1.9
		Exploit recovery mechanism (T3)	9.9	–	–
	Elevate privileges on user agent	n/a	n/a	n/a	n/a
VC store	Spoofing VCs	Create self-attested claim (T10)	9.6	Issuer verification (C7)	1.4
	Tampering with the VCs	Change VC value (T11)	8.4	Signature verification (C8)	0.9
	Repudiate VC issuance	Delete VC (T12)	6.5	VC registry model (C9)	1.2
	Reveal confidential VC info.	Gain unauthorized access (T13)	7.7	–	–
		Request unnecessary data (T14)	7.7	n/a	n/a
	Deny VC store serviceability	Delete VC (T12)	6.5	VC registry model (C9)	1.2
	Elevate privileges on the VC store	–	–	–	–

Table 1 continued

Component	STRIDE Category	Threat	Rating	Countermeasures	Mitigation
Org agent	Spoofing identity actions	Misuse identity (T15)	9.6	Split control scheme (C10)	1.9
		Acquire private key (T1)	8.7	Secure enclave (C1)	1.3
				Remote revocation process (C3)	–
		Exploit recovery mechanism (T3)	9.9	–	–
	Tampering with the org agent's data	Manipulate configuration (T16)	6.5	Audit trail (C11)	–
	Repudiate identity actions	Illegitimate VC revocation (T17)	6.5	Split control scheme (C10)	1.4
		Deliberately disclose private key (T4)	8.7	Secure enclave (C1)	1.3
		Revoke identity (T5)	9.1	Action timestamping (C4)	1.2
	Reveal confidential identity info	n/a	n/a	n/a	n/a
	Deny identity actions	Revoke identity (T5)	9.1	Action timestamping (C4)	1.2
		Delete private key (T8)	5.2	Device access protection (C2)	1.3
				Recovery procedure (C3)	–
	Elevate privileges on the org agent	Take-over role (T18)	7.1	Split control scheme (C10)	1.4
Trust & data store	Spoofing trusted issuers or VC data	n/a	n/a	n/a	n/a
	Tampering with trusted issuers or VC data	Circumvent VC verification (T19)	7.1	–	–
		Manipulate trusted issuers (T20)	5.5	–	–
	Repudiate trusted issuers or VC data	n/a	n/a	n/a	n/a
	Reveal confidential iss./ ver. info	Disclose trusted issuers (T21)	3.3	–	–
	Deny store serviceability	–	–	–	–
	Elevate privileges on the store	–	–	–	–

Table 2 SSI component threats, impact and countermeasures (2/2)

Component	STRIDE category	Threat	Rating	Countermeasures	Mitigation
Identity Holder/Verifier/Issuer Node	Spoofing node messages	Propagate forged message (T22)	6.5	Blockchain network mon. (C11)	1.2
	Tampering with the node	Manipulate state (T23)	5.5	Blockchain network mon. (C11)	1.2
		Manipulate configuration (T24)	5.5	–	–
	Repudiate node message	n/a	n/a	n/a	n/a
	Reveal confidential node info	n/a	n/a	n/a	n/a
	Deny node serviceability	Reset or close connections (T25)	6.5	Message sender verification (C12)	1.2
		Flood connections (T26)	6.5	Message sender verification (C12)	1.2
	Elevate privileges on the node	n/a	n/a	n/a	n/a
VDR	Spoofing VDR entries	Exploit smart contract vuln. (T27)	8.1	Vulnerability scan (C13)	1.3
	Tampering with the VDR	Exploit smart contract vuln. (T27)	8.1	Vulnerability scan (C13)	1.3
	Repudiate VDR entries	Exploit smart contract vuln. (T27)	8.1	Vulnerability scan (C13)	1.3
	Reveal confidential VDR info	n/a	n/a	n/a	n/a
	Deny VDR serviceability	Deactivate VDR smart contract (T28)	6.5	Vulnerability scan (C13)	1.2
		Manipulate blockchain conf. (T29)	9.6	Split control scheme (C10)	1.9
		Exploit smart contract vuln. (T27)	8.1	Vulnerability scan (C13)	1.3
	Elevate privileges on the VDR	Take-over VDR owner role (T30)	7.7	Vulnerability scan (C13)	1.9
				Split control scheme (C11)	1.9
		Take-over identity holder role (T31)	7.7	Vulnerability scan (C13)	1.4
		Take-over issuer role (T32)	9.6	Vulnerability scan (C13)	1.4
Com. Channels	Spoofing communication	Spoof communication partner (T33)	9.6	Comm. partner verification (C14)	1.4
	Tampering with the communication	Spoof communication partner (T33)	9.6	Comm. partner verification (C14)	1.4
	Repudiate communication	Dispute message (T34)	7.7	Message sender verification (C12)	1.2
	Reveal confidential information	Traffic interception (T35)	7.7	Communication encryption (C15)	1.4
	Deny communication	Reset or close connections (T25)	6.5	Message sender verification (C12)	1.2
		Flood connections (T26)	6.5	Message sender verification (C12)	1.2
	Elevate privileges in the comm	n/a	n/a	n/a	n/a

Table 3 Threat comparison of IdM models

Threat	Iso	Centr	SSI
Acquire Private Key (T1)	○	○	●
Steal or Covertly Access User Agent Device (T2)	○	○	●
Exploit Recovery Mechanism of Identity (T3)	●	●	●
Deliberately Disclose Private Key (T4)	○	○	●
Revoke Identity (T5)	●	●	●
Deliberately Loose User Agent Device (T6)	○	○	●
Steal or Break User Agent Device (T7)	○	○	●
Delete Private Key (T8)	○	○	●
Exploit Identity Revocation (T9)	●	●	●
Create Self-attested Claim (T10)	–	–	●
Change VC Value (T11)	○	○	●
Delete VC (T12)	●	●	●
Gain Unauthorized Access (T13)	●	●	●
Request Unnecessary Data (T14)	–	●	●
Misuse Identity (T15)	●	●	●
Manipulate Configuration (T16)	–	●	●
Illegitimate VC Revocation (T17)	–	●	●
Take-over Role (T18)	–	●	●
Circumvent VC Verification (T19)	●	●	●
Manipulate Trusted Issuers (T20)	–	●	●
Disclose Trusted Issuers (T21)	–	●	●
Propagate Forged Message (T22)	–	●	●
Manipulate State (T23)	–	●	●
Manipulate Configuration (T24)	–	–	●
Reset or Close Connections (T25)	–	–	●
Flood Connections (T26)	–	–	●
Exploit Smart Contract Vulnerabilities (T27)	–	–	●
Deactivate VDR Smart Contract (T28)	–	–	●
Manipulate Blockchain Configuration (T29)	–	–	●
Take-over VDR Owner Role (T30)	●	●	●
Take-over Identity Holder Role (T31)	●	●	●
Take-over Issuer Role (T32)	●	●	●
Spoof Communication Partner (T33)	–	●	●
Dispute Message (T34)	–	●	●
Traffic Interception (T35)	–	●	●

(C1) of the iPhone. Therefore, a protection measure against the acquiring of the private key (T1) exists. Considering the threat of stealing or covertly accessing the smartphone (T2), uPort does not enforce access protection, e.g. face id or additional PINs. Thus, uPort is vulnerable to this threat. Furthermore, there is no defined remote revocation process (C3) for the identifier. Moreover, uPort offers an identity recovery (C5) and backup process. For the recovery, the user must securely store a passphrase that consists of several words. The screen showing the passphrase is not protected from screenshots. This may lead to an additional threat surface. However, no external parties are involved in the recovery mechanism.

Assessing the category of the repudiation of identity actions, the deliberate disclosure of the private key (T4) is protected by the use of a secure enclave (C1). The use of action time stamping (C4) could not be verified to prevent illegitimate revocation (T5). However, there is no official revocation process (T9) and, therefore no threat to exploit it. Furthermore, uPort is vulnerable to the deliberate loss of the user agent's device (T6) because no device access protection (C2) and remote revocation (C3) process is offered. Analyzing the denial of identity actions, stealing or breaking the user agent device (T7) is countered by a recovery mechanism (C3). As described, uPort offers a passphrase-based recovery mecha-

nism. The deletion of the private key (T8) is possible by the deletion of the uPort app. The user agent does not enforce any access protection (C2) on the device. However, the recovery procedure (C5) enables a restore.

Overall, 8 threats for the uPort user agent are addressed. Nonetheless, no protection measures against stealing or covertly access the user agent device (T2) and the deliberate loss of the device (T6) are implemented.

7 Discussion

We analyzed on a conceptual level the threat surface of the SSI paradigm and identified high-level threats including associated countermeasures. The basis of the threat analysis for SSI forms the components and their communication paths of the specified DFD. We created additional DFDs for the comparison towards the traditional models. In case the structure of the DFDs misses a significant component or communication flow, the respective threats are not captured in our analysis. The DFD's level of detail reflects the threats and protection measures.

Furthermore, the listed threats can only serve as a starting point for a detailed security analysis of a specific SSI IdMS. In particular, general platform security measures, that are independent from the SSI context, are relevant for security. It encompasses the used encryption and signature algorithms in the various libraries. Furthermore, the actual application integration is subject for evaluation. If a specific SSI IdMS has additional components, that are not captured in the DFD, the analysis must consider them.

8 Future work

Future work of our research can encompass several directions. On the side, the implementation level threat analysis is an highly relevant field of research. The investigated conceptual level already provides insights to the threat surface. However, the implementation level of a specific SSI IdMS comprises an extended level of threats. Additionally, the security examination of a wider range of popular SSI IdMS and the comparison of their security posture is an interesting field of research. Results in this area can be leveraged to increase the overall security level of SSI IdMS.

9 Conclusion

The new SSI paradigm addresses inherent security and privacy issues of the traditional IdM models. To analyze the security of SSI, we adopted a hybrid threat modeling approach that combines STRIDE, attack trees, and CVSSv3 ratings. DFDs of the isolated, centralized and SSI paradigm serve as the basis for the assessment and to draw comparative conclusions along the model development. Within our examination, we identified 35 SSI-specific threats and 15 protection measures. (answering RQ2). Comparing the traditional models to the SSI paradigm, the number of security zones, components, and communication channels increased significantly. Along the same lines, the threat surface expanded towards the SSI model (answering RQ1). Moreover, we practically applied the developed threat analysis methodology to uPort and evaluated its user agent. We found that threats regarding the loss, theft and covert access of the user agent device are not adequately addressed by protection measures. Overall, the SSI paradigm manifests an increased threat surface and requires additional security measures.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability The authors did not use any specific research data for quantitative evaluations.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethical approval The authors did not receive support from any organization for the submitted work. The authors strongly comply with ethical standards.

Appendix A: CVSSv3 score metrics components

We use the CVSSv3 rating methodology to score the identified threats. The base score rating encompasses exploitability and impact metrics. The exploitability characteristics comprises the attack vector, attack complexity, required privileges, user interaction and scope. The attack vector differentiates levels of local or network-based attack vicinity. We evaluate the value according to the proximity that is required by the adversary. Attack complexity and required privileges provide subjective levels to rate. We evaluated

it to the lowest level with an increased attack complexity and required privileges if countermeasures are applied. User interaction refers to any required action by the entities.

Moreover, the scope is assessed to either changed or unchanged. A changed scope might affect several authorities. This is the case for distributed IdM like the SSI model. The impact is qualified for the security objectives confidentiality, integrity and availability with the levels none, medium or high. We evaluate the impact according to the threat's impact on the respective security objective.

Appendix B: threat heatmaps

The description of the threats in Sect. 4 and the overview in Tables 1 and 2 might no guide the reader directly to the most vulnerable component or the most severe threat. Therefore, we created a heatmap for each component to lead the analyst visually to the right direction. Figures 4, 5, 6, 7, 8, 9 and 10 show the heatmaps. On each heatmap the applicable threats represent the columns and the rows reflect the countermeasures. The cells of the heatmap show the CVSSv3 scoring. The row NC expresses the CVSSv3 scoring of a threat without any countermeasure.

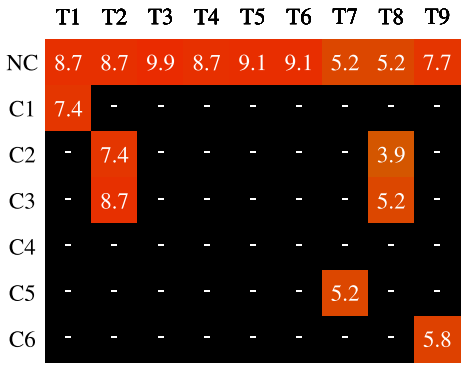

Fig. 4 Heatmap user agent

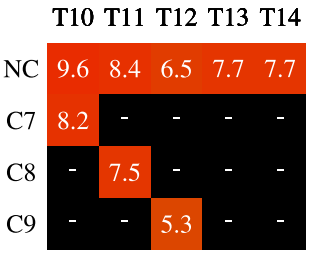

Fig. 5 Heatmap VC store

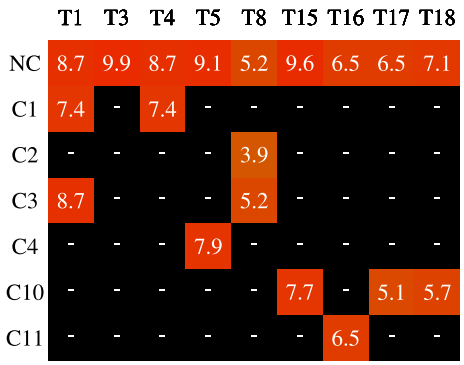

Fig. 6 Heatmap org agent

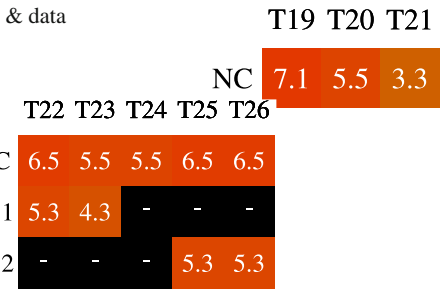

Fig. 7 Heatmap trust & data store

Fig. 8 Heatmap nodes

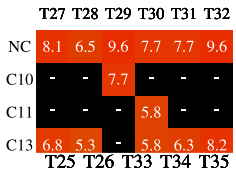

Fig. 9 Heatmap VDR

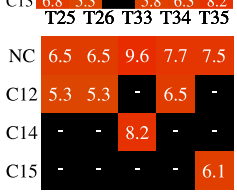

Fig. 10 Heatmap communication channels

References

1. Tobin, A., Reed, D.: The inevitable rise of self-sovereign identity: a white paper from the sovrin foundation (2017). Accessed on 2022-03-04. [Online]. Available: https://sovrin.org/wp-content/uploads/2017/06/The-Inevitable-Rise-of-Self-Sovereign-Identity.pdf
2. Jøsang, A., Fabre, J., Hay, B., Dalziel, J., Pope, S.: Trust requirements in identity management. In: Proceeding of the 2005 Australasian Workshop on Grid Computing and e-Research (AusGrid), pp. 99–108 (2005)
3. Allen, C.: The path to self-sovereign identity (2016). Accessed on 2022-03-04. [Online]. Available: http://www.lifewithalacrity.com/previous/2016/04/the-path-to-self-sovereign-identity.html
4. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Accessed on 2022-03-04. [Online]. Available: https://gavwood.com/paper.pdf
5. Kuperberg, M.: Blockchain-based identity management: a survey from the enterprise and ecosystem perspective. IEEE Trans. Eng. Manag. 1008–1027 (2019)
6. Shostack, A.: Threat Modeling: Designing for Security. Wiley, Hoboken (2014)
7. Deborah, D.B.F., Bodeau, J., McCollum, Catherine D.: Cyber threat modeling: survey, assessment, and representative framework (2018). Accessed on 2022-03-04. [Online]. Available: https://apps.dtic.mil/sti/pdfs/AD1108051.pdf
8. Li, Q., Chen, Y.-L.: Data Flow Diagram 85–97 (2009)

9. Potteiger, B., Martins, G., Koutsoukos, X.: Software and attack centric integrated threat modeling for quantitative risk assessment. In: Proceeding of the 2016 Symposium and Bootcamp on the Science of Security (HotSos), pp. 99-108 (2016)
10. Schneier, B.: Attack Trees (1999). Accessed on 2021-01-17. [Online]. Available: https://www.schneier.com/academic/archives/1999/12/attack_trees.html
11. First. Common vulnerability scoring system version 3.1: specification document. Accessed on 2022-03-04. [Online]. Available: https://www.first.org/cvss/specification-document
12. Lundkvist, C., Heck, R., Torstensson, J., Mitton, Z., Sena, M.: uport: a platform for self-sovereign identity (2016). Accessed on 2022-03-04. [Online]. Available: http://blockchainlab.com/pdf/uPort_whitepaper_DRAFT20161020.pdf
13. Conti, M., Sandeep Kumar, E., Lal, C., Ruj, S.: A survey on security and privacy issues of bitcoin. IEEE Commun. Surv. Tutor. **20**(4), 3416–3452 (2018)
14. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008). Accessed on 2022-03-04. [Online]. Available: https://bitcoin.org/bitcoin.pdf
15. Li, X., Jiang, P., Chen, T., Luo, X., Wen, Q.: A survey on the security of blockchain systems. Futur. Gener. Comput. Syst. **107**(C), 841–853 (2020)
16. Mehar, M.I., Shier, C.L., Giambattista, A., Gong, E., Fletcher, G., Sanayhie, R., Kim, H.M., Laskowski, M.: Understanding a revolutionary and flawed grand experiment in blockchain: the dao attack. J. Cases Inf. Tech. **21**(1), 19–32 (2019)
17. Shahriar, M.A., Bappy, F.H., Hossain, M.A., Saikat, D.D., Ferdous, M.S., Chowdhury, M., Bhuiyan, M.Z.A.: Modelling attacks in blockchain systems using petri nets. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1069–1078 (2020)
18. Pinna, A., Tonelli, R., Orru, M., Marchesi, M.: A petri nets model for blockchain analysis. Comp. Jour. **61**, 374–1388 (2018)
19. Dingle, P., Hammann, S., Hardman, D., Winczewski, C., Smith, S.: Alice attempts to abuse a verifiable credential. rebooting the web of trust IX: Prague (2019). Accessed on 2020-12-12. [Online]. Available: https://github.com/WebOfTrustInfo/rwot9-prague/blob/master/final-documents/alice-attempts-abuse-verifiable-credential.pdf
20. Allen, C., Brock, A., Buterin, V., Callas, J., Dorje, D., Lundkvist, C., Kravchenko, P., Nelson, J., Reed, D., Sabadello, M., Slepak, G., Thorp, N., Wood, H.T.: Decentralized Public Key Infrastructure. Rebooting the Web of Trust I: San Francisco (2015). Accessed on 2020-12-12. [Online]. Available: https://github.com/WebOfTrustInfo/rwot1-sf/blob/master/final-documents/dpki.pdf
21. Stöcker, C., Smith, S.M., Cabellero, J.: Quantum Secure DIDs. Rebooting the Web of Trust X: Virtual Papers (2020). Accessed on 2020-12-12. [Online]. Available: https://github.com/WebOfTrustInfo/rwot10-buenosaires/blob/master/final-documents/quantum-secure-dids.pdf
22. Stokkink, Q., Epema, D., Pouwelse, J.: A Truly Self-Sovereign Identity System (2020). arXiv preprint arXiv:2007.00415
23. Douceur, J.R.: The sybil attack. In: Proceeding of the 2002 International Workshop on Peer-to-Peer Systems (IPTPS), vol. 2429, pp. 251–260 (2002)
24. Alexopoulos, N., Daubert, J., Mühlhäuser, M., Habib, S.M.: Beyond the hype: on using blockchains in trust management for authentication. IEEE Trustcom BigDataSE ICESS **2017**, 546–553 (2017)
25. Kim, B.G., Cho, Y.-S., Kim, S.-H., Kim, H., Woo, S.S.: A security analysis of blockchain-based did services. IEEE Access **9**, 22894–22913 (2021)
26. The Linux Foundation. Hyperledger Indy. Accessed on 2022-03-04. [Online]. Available: https://www.hyperledger.org/use/hyperledger-indy
27. Ahmad, Z., Ab Manan, J.-L., Sulaiman, S.: User requirement model for federated identities threats. In: Proceeding of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), vol. 6, pp. 317 – 321 (2010)
28. Khattak, Z.A., Sulaiman, S., Manan, J.-L.A.: A study on threat model for federated identities in federated identity management system. In: Proceeding of the 2010 International Symposium on Information Technology, vol. 2, pp. 618–623 (2010)
29. Dominicini, C.K., Simplício, M.A., Sakuragui, R.R.M., Carvalho, T.C.M.B., Näslund, M., Pourzandi, M.: Threat modeling an identity management system for mobile internet. In: Proc. of the 9th International Information and Telecommunication Tech. Symposium (I2TS) (2010)
30. Shevchenko, N., Chick, T.A., O'Riordan, P., Scanlon, T.P., Woody, C.: Threat modeling: a summary of available methods (2018). Accessed on 2022-03-04. [Online]. Available: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=524448
31. Wang, R., Gao, L., Sun, Q., Sun, D.: An improved cvss-based vulnerability scoring mechanism. In: 2011 Third Int. Conf. on Multimedia Information Networking and Security (MINES), pp. 352–355 (2011)
32. Frederiksen, T., Hesse, J., Lehmann, A., Torres Moreno, R.: Id. Management: state of the art, Chall. and Persp., pp. 45–62 (2020)
33. Sporny, M., Longley, D., Chadwick, D.: Verifiable credentials data model 1.0 (2019). Accessed on 2022-03-04. [Online]. Available: https://www.w3.org/TR/vc-data-model/
34. Narayanan, A., Clark, J.: Bitcoin's academic pedigree. Commun. ACM **60**(4), 36–45 (2017)
35. Mühle, A., Grüner, A., Gayvoronskaya, T., Meinel, C.: A survey on essential components of a self-sovereign identity. Comput. Sci. Rev. 80–86 (2018)