The Power of Many: Securing Organisational Identity Through Distributed Key Management

Mariia Bakhtina (✉)[1][0000-0002-0940-9713], Jan Kvapil[2][0009-0005-3303-1481], Petr Švenda[2][0000-0002-9784-7624], and Raimundas Matulevičius[1][0000-0002-1829-4794]

[1] University of Tartu, Tartu, Estonia {bakhtina, rma}@ut.ee
[2] Masaryk University, Brno, Czech Republic
kvapil@mail.muni.cz, svenda@fi.muni.cz

Abstract. Organisational Digital Identity (ODI) often relies on the credentials and keys being controlled by a single person-representative. Moreover, some Information Systems (IS) outsource the key management to a third-party controller. Both the centralisation and outsourcing of the keys threaten data integrity within the IS, allegedly provided by a trusted organisation. Also, outsourcing the control prevents an organisation from cryptographically enforcing custom policies, e.g. time-based, regarding the data originating from it. To address this, we propose a Distributed Key Management System (DKMS) that eliminates the risks associated with centralised control over an organisation's identity and allows organisation-enforceable policies. The DKMS employs threshold signatures to directly involve multiple organisation's representatives (e.g. employees, IS components, and external custodians) in data signing on its behalf. The threshold signature creation and, therefore, the custom signing policy inclusion, is fully backwards compatible with commonly used signing schemes, such as RSA or ECDSA. The feasibility of the proposed system is shown in an example data exchange system, X-Road. The implementation confirms the ability of the design to achieve distributed control over the ODI during the operational key phase. Excluding a network delay, the implementation introduces less than 200ms overhead compared to the built-in signing solution.

Keywords: organisational digital identity · key management · security · zero trust · distributed control · threshold signatures

1 Introduction

Organisational Digital Identity (ODI) [3] defines an organisation and its attributes for other entities through credentials. It enables trust between business partners and ensures the authenticity and confidentiality of cross-organisational data exchanges [31]. ODI commonly relies on centrally managed credentials and keys used in Public Key Infrastructure (PKI). A centralised management introduces a

[3] If not specified otherwise, we also use *identity* to refer to ODI.

single point of failure [7] in the system. As an alternative, decentralised Identity Management (IdM) based on distributed ledgers has been proposed [7]. Decentralised ODI enhances security and identity control. However, this approach requires a new governance framework and a shift of the participants to a new infrastructure. This process is time-consuming and may require legislative updates, especially in information systems used in e-governance like X-Road [22] and Gaia-X [1]. Moreover, the decentralised trust and IdM address external threats, while internal actors contribute significantly to data breaches through privilege misuse [29]. Thus, centralised control by internal actors is an open issue which threatens the authenticity of messages sent on behalf of ODI.

Regardless of the identity and trust model, organisations may use digital wallets or hardware security modules as a part of their Information System (IS) to store their identity's certificates and key material, with policies defining access rights [12]. Proprietary solutions like OpenID [2] control internal authorisation, determining who can initiate cross-organisational data exchange on behalf of ODI. Meanwhile, some organisations prefer external trusted partners to manage their identity and key materials [12,19,28], removing access control from their IS, but this raises concerns about potential compromise or misuse. Based on [7,12], we recognise the need for a more secure yet backwards compatible identity model, which would allow enforcing security through custom access policies (under what conditions are ODI-related cryptographic keys accessible/used) for information systems that rely on a centrally issued ODI.

We see a lack of attention from the Information Systems Engineering (ISE) community to the cryptographic measures used in IdM, which aims to protect ODI from insider threats. Thus, state-of-the-art cryptographic measures are researched mainly by cryptographers and the formal security analysis community. Meanwhile, the IdM's characteristics are primarily driven by the business needs and the goals of IS users. In this paper, we bridge the gap between formal security research and research of ISE, showing how the former addresses the latter's challenges. Assuming centrally issued PKI-based credentials, we aim to eliminate centralised usage of identity by embracing the zero trust paradigm [9]. While key management enables digital identity per se, this paper considers the following research question: *how to secure a centrally issued organisational digital identity through key management mechanisms for achieving zero trust*?

In this paper, we follow the design science research method [13]. We review the knowledge base of cryptographic and business mechanisms which help secure ODI with a focus on key management. As a result, we design an artefact of Distributed Key Management System (DKMS) that secures a centrally issued ODI. The DKMS uses partial custody and threshold signatures to secure ODI in cross-organisational data exchange through zero trust principles. The proposed system ensures that messages signed on behalf of ODI originate from the organisation and are not created by a single trusted ODI custodian or a company representative. The DKMS design application and utility are evaluated through implementation for the X-Road data exchange system.

2 Background

To provide new services, multiple organisations may connect their systems to form a new information system. Securing such a system requires identifying the individual organisations to allow authenticity and integrity of the shared data. Data exchange systems enable secure interoperable cross-organisation data exchange between separately built information systems in e-governance and other industries [2,17,18]. The examples of commercial, worldwide used systems are X-Road [22], UXP [10], and Dawex Data Exchange Platform, European Gaia-X [1]; while Dutch NLX [30] and Australian Secure Data Exchange application are national. Yet, they operate securely under the assumption of already established trust between collaborating entities and their digital identities.

Digital identity enables participants of the data exchange systems to confirm the authenticity of involved entities. Credentials associated with identities are the documents enabling the verification, with issuance procedures determined by a trust model. While the procedures of issuing credentials in decentralised and centralised identity models vary [7], PKI stays essentially the same. Thus, the key management is equally relevant regardless of the trust and identity model. Both the internal and external trusted representatives that manage the keys may become malicious. To mitigate this threat, the organisation should control every single data exchange attempt made on its behalf. Such a mitigation strategy [9] of avoiding the need for trust is the key component of the Zero Trust (ZT) architecture [25]. Among the key components of ZT architecture is the policy decision point that enforces control over access to resources based on the defined policies and is supported by PKI and identity management, on which this work is focused.

2.1 Targeted System Characteristics

As zero trust is a maturing strategy, little research has been done to map which of the existing security mechanisms can help secure ODI with respect to it. To study the security of an ODI and Identity Management (IdM) system, we consider characteristics targeted by identity and key management mechanisms. The need for the characteristics depends on the scenario. Through the literature review, we gathered a set of non-functional characteristics that affect ODI's security. In [5], we describe the literature procedure and provide a mapping of system characteristics with reviewed in Sec. 2.2 mechanisms.

The following business-oriented characteristics may be targeted, primarily reflecting the trust among the individual business entities. *Trustlessness* is a part of the zero trust paradigm and refers to the ability of the system to operate without relying on the honest behaviour of internal or external entities [19,25]. To react to the dishonest behaviour, the IdM system may target to deliver *traceability* that refers to the ability to know who did what, when, and how [11,12]. The more proactive approach to secure the system against internal attacks is *privilege escalation prevention*, which aims to restrict users from gaining unauthorised access. In this research, we consider privilege escalation as a result of both privilege escalation attacks and privilege misuse. Another way of enabling zero trust is *decentralisation*, which refers to distributing the responsibility for managing ODI and its keys across multiple entities to enhance security, resilience, and user control. As a result, the organisation can differentiate between *multiple users* of the ODI. Another business-demanded aspect of its system's security is *availability*, which is the ability to deliver its value continuously. System availability [12] is crucial both in the everyday data exchange and in the case of extraordinary trigger events – e.g. a loss of access to the keys or if an employee that is involved in the ODI management leaves or becomes malicious. Finally, the *usability* refers to the feasibility and convenience of the identity and key management. Usability encompasses the backwards compatibility of the proposed design with the existing infrastructure. Besides, usability defines whether the end-users will use the ODI following the defined interaction set-up with ODI [12,19,28].

2.2 Review of Key Management Mechanisms

Assuming ODI relies on PKI, key management is an enabler of digital identity per se and ensures the security of IdM systems. The key management includes pre-operational, operational, post-operational and destroyed phases [8]. IdM is involved in the four stages: (*i*) generation and distribution of asymmetric key pairs (i.e. private and public keys) for the key generation during the pre-operational phase; (*ii*) the key registration and the public key certification during the pre-operational phase; and (*iii*) storage, usage and backup of keys during the operational phase. The certification is crucial for establishing trust within the system as it enables verification of the exchanged data integrity. Our research aims to comply with the existing verification process to remain IdM backwards compatible. Thus, we exclude the second stage from our analysis. Fig. 1 depicts artefacts used for key management that are discussed in this section.

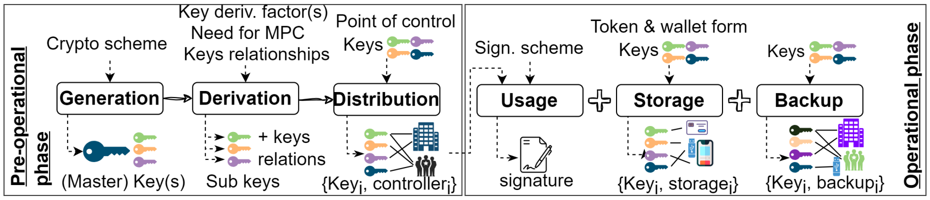

Fig. 1. Key management phases: stages, parameters and artefacts.

Here, we overview the cryptographic and business mechanisms found through the literature review, details of which can be found in [5]. The paper focuses on asymmetric cryptographic schemes that provide digital signatures.

Pre-operational Key Phase

Key generation. To use any of the existing signature schemes, a (master) key pair needs to be generated. The signature schemes differ by key and signature size and mathematical structure, computational efficiency and security assumptions. Regardless of the scheme, the key pair creation process involves a cryptographically secure pseudorandom number generator that needs to be seeded. At the expense of security, the seed can also be derived from a user-provided input, such as a password (using a password-based key derivation function). To improve trustlessness and resilience and not rely on a single factor, the multi-factor key derivation function [20] can be utilised. Also, there are protocols for Distributed Key Generation (DKG), such as FROST [16], where multiple parties collaboratively generate secret shares of a key in a distributed manner. DKGs have a major benefit over the previous methods – the key never exists in a complete form at a single location. Finally, once generated, the master (public) key can have a certificate issued to bind it to a particular entity.

Sub-keys derivation. Data can be signed either directly by the master private key or by sub-keys derived from it. Bitcoin Improvement Proposal No. 32 (BIP32) [23] defines deterministic key derivation, where sub-keys are obtained by a *path* from the master key. The path is also needed for the signature verification algorithm. The deterministic derivation is specified only for certain schemes based on elliptic curves (as in Bitcoin) or lattices [4]. Another option is threshold signature, where at least a threshold of K signers out of all N need to collaborate to form a valid signature. As a result, an attacker would need to compromise at least K signers to forge a valid signature. As mentioned previously, key shares can be derived using DKGs (e.g. FROST [16]) or generated and distributed by a trusted dealer (e.g. as in RSA-based threshold signature schemes [27]).

Key distribution. The key distribution phase defines a point of control over the ODI's keys and their delivery to controllers. The point of control is driven by the level of custody over the keys. Non-custodial (i.e. self-custodial) approach [15] enables more control but more overhead, as the ODI owner is also the controller responsible for securing the keys. An opposite approach is full outsourcing of the control (i.e. custody) [19], which removes overhead from the ODI owner but disable control. Custodial ODI directly conflicts with the ZT strategy as it puts complete trust in a third-party custodian. The custodian manages the keys and can conduct any operations on behalf of the organisation. Thus, none of the approaches address trustlessness, resilience, or protection from privilege escalation and enable multiple conditional users using internal organisational access control.

Operational Key Phase

Key storage. The keys are stored on tokens that generate, protect and manage them. The token can be a software application installed on a general device, e.g. the one that requests the signature, or a full-fledged external device such as a Hardware Security Module (HSM). The dedicated device can be a USB token, a smart card (e.g. JavaCard) or a trusted platform module built into an end-consumer device. Another device relevant for storing keys is a hardware wallet. Finally, deterministically derived keys do not require storage, as they can be generated on the spot – needing at least the threshold number of factors in the case of DKGs. The form of a token affects ODI's portability and usability.

Key usage. The key usage is split into data signing and signature verification algorithms. For a single signer and a single pre-generated key, the key should be available either explicitly (e.g. for software tokens) or through a signing interface as with dedicated signing devices. If the master key is derived, the appropriate number of valid factors must be provided to derive the signing key. If BIP32 deterministic derivation is used, the path from the master key to the child key is also needed. Threshold schemes involving multiple signers improve recoverability and, thus, resilience – any quorum of more than the threshold of signers can create a signature using there key key shares. Replacing a single ODI controller with a threshold group of K-out-of-N protects against privilege escalation. Most threshold signatures do not disclose which signers have participated in the signing. Thus, threshold signatures may help hide signers' identities from the verifier. However, all signers are accountable for N-out-of-N threshold groups. Also, elaborate policies, like signing only during a specific time range (such as working hours), can be enforced through automated signer applications. Threshold groups can also be nested – e.g. top-level 3-out-of-3 group with two shares residing with individual representatives and the last share being an employee auto-signer that signs if another K-out-of-N employee group also provides a signature. Thereby, the ODI owner can trace back to who of the key shareholders participated in the signature creation or mandate policies.

Key backup. To enable recovery of lost keys, an identity holder should back up the secret key elsewhere than the storage used during regular key operations. Such backup storage can be an additional HSM, a trusted execution environment for key backup [28] or a third-party custody of the keys. The latter carries the risks of identity theft through privilege escalation [28]. For enhanced security, an identity holder may employ multiparty computation and Shamir's Secret Sharing, distributing key shares among multiple custodians [28]. For multiparty computation, keys should be generated with DKG, while secret sharing requires the distribution of later-derived parts of a single key. A defined minimum threshold of custodians must provide their key shares to the identity holder to recover the backed-up key [28].

To sum up, we identified mechanisms that help secure keys and signatures and, thus, ODI itself. The commonly used approaches of key management can address the problem of centralisation or bring zero trust in a targeted manner, while none of them addresses the problem on their own. In this paper, we address this research gap by proposing a distributed key management system that can bring ZT and decentralisation through the combination of reviewed mechanisms.

3 Design of a Distributed Key Management System

In this section, we present a design of a Distributed Key Management System (DKMS) for ODI as depicted in Fig. 2. It enables zero trust within the organisation's information system and is compatible with any trust model. The design assumes that the identity's credentials rely on PKI and, thus, prescribes a selection of cryptographic and business mechanisms for each stage of the key lifecycle. The design aims to eliminate the centralisation of control over ODI throughout the lifetime of keys. It increases the set of representatives or custodians to enable partial custody of ODI.

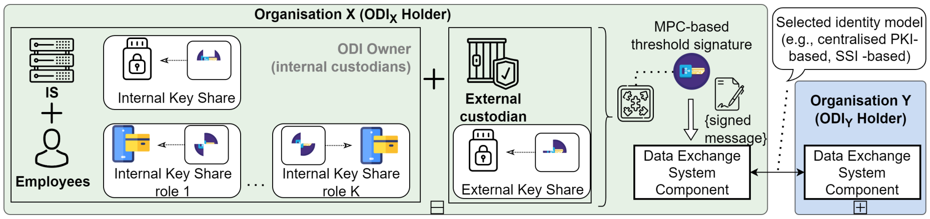

Fig. 2. DKMS for organisational digital identity in cross-organisational data exchange.

To bring zero trust into ODI management, we propose a combination of the reviewed self-custodian and custodian control, namely, *partial custody*. An IS with centralised control over ODI is prone to a single point of failure [24] due to a strong assumption on a single fully trusted entity for signing messages on behalf of an organisation. If the entity is compromised or its administrator becomes malicious, the IS cannot assure the authenticity and, thus, integrity, of the provided information. The partial custody brings decentralisation of control and storage of keys for signing or recovery. Ergo, the system can operate with reduced trust in each single entity. The distribution of the keys eliminates the single point of failure since multiple key controllers would need to cooperate to compromise the system's security. By that, partial custody boosts the overall security posture of an IS and mitigates the risks associated with centralised control of keys. DKMS allows trustlessness, improves recoverability and resilience, supports traceability, and may enable the prevention of privilege misuse.

To achieve zero trust through partial custody in DKMS, the decentralisation of control over keys may be introduced in one of two steps – during key generation or distribution. First, the keys can be initially generated in a decentralised manner. It refers to a decentralised ODI through self-sovereign identity principles usage and decentralised key generation for all the entities involved in the ODI management and data exchange (e.g. like it was proposed in [7]). Second, if there should be one centrally generated master key, decentralisation can be introduced in the step of distributing the keys (e.g. through distributed key generation or key shares derivation) so that the derived sub-keys are used for the operational phase.

Regardless of the phase during which key shares are generated and distributed to semi-trusted custodians, an organisation can derive multiple keys to enable distributed key storage and threshold signature creation. Semi-trusted custodians can be internal entities (IS components or employees) and external service providers (i.e. external custodians). When new employees come, and old ones leave, the distributed key generation re-sharing can be performed while keeping the (certified) public key the same.

For the point of control during keys distribution, we use *partial custody* achieved through *threshold signatures*, where internal entities (e.g. employees' roles and an information system component) and an external custodial are the contributors to the signing of messages on behalf of an ODI. Users' keys are the key shares for the threshold signature. Access to these shares can be based on the user's identity following internal organisation policy. Each entity can use *different token forms* to store its share. Thereby, the proposed DKMS has threshold signatures as a policy decision point component in respect of zero trust architecture [25].

In this paper, we show that threshold signature allows organisations to distribute control over the identity. First, key shares allow the distribution of trust among the organisational parties – employees and IS components – which are primary representatives of the organisation and the initiators of messages. Second, it helps maintain access control in case of employee turnover in an organisation. Finally, a threshold of K shares protects the ODI and organisation against a compromise of up to $\mathrm{K}-1$ signing parties.

Additionally, DKMS brings access policies enforcement. In a centralised trust model, the external key controllers need to be trusted with complying to the access policies. With threshold signatures, the organisation can implement these access policies by assigning shares to individual internal IS components that are then responsible for enforcing the policies. For example, an IS can include a component that signs only at a specified time range and another that verifies the requester's IP address. Compliance with the access policy is enforceable by the organisation and does not require any changes on the verifying side.

For zero trust secure ODI, we propose distributing control over ODI between internal and external custodians so that multiple parties compose a group for signing messages. For example, an organisation can generate N key shares based on a certified ODI master key, where one share is given to an IS (i.e. automatic or policy-based signer), one – to an external custodian who represents the data exchange system, and others are distributed among employees of different roles who can initiate or validate the messages, e.g. semi-automatically. Signing a message requires the participation of K-out-of-N parties, where $\mathrm{K} \leq \mathrm{N}$, (e.g. an IS, external custodian and any employee). Additionally, the second layer of threshold signature by employees of different roles or physical identities may be set up if needed to form a tree-like signing structure. In the end, the derived key shares and threshold signature on behalf of an organisation ensure that the valid signature is created based on the definition by an organisation group of ODI representatives. For internal entities, either software or hardware modules for storing key shares may be selected.

The proposed system should be applied when (i) organisational identity relies on public key infrastructure; (ii) organisation uses digital identity for cross-organisational IS (e.g. through data exchange systems for connecting standalone ISs); (*iii*) organisation want to remove a single point of control over its identity; (*iv*) multiple organisation representatives can contribute to the signing.

4 DKMS Design Evaluation

4.1 X-Road Use Case

X-Road is a distributed data exchange system between Information Systems (ISs) within the trusted network [21]. X-Road relies on PKI to ensure trust between members of the network. Members are organisations with their ISs used for operations by internal users (e.g. employees) and external users (e.g. customers). For cross-organisational data exchange, a component called *Security Server* (SS) serves as an intermediary [21]. Let us consider Member-Client and Member-Provider as X-Road Members with set-up SSs. The Client requests data from the Provider. The SSs are the components which are responsible for the PKI-verified message exchange. The Client's SS signs the data requests on behalf of the Client. The Provider's SS verifies the Client's signature in the received request and signs the response on behalf of the Provider. Finally, the response signature is verified by the Client's SS. Additionally, SS logs the data exchange so that a third party can check the Member's request or response signature if a proof is needed (e.g. in a court) [21]. In the current set-up, the X-Road is oriented toward protecting the confidentiality and integrity of the data exchange messages delivered via the public Internet. Meanwhile, all the entities inside the network are fully trusted (including SSs) based on the PKI certificates, and SS is not necessarily managed by the organisation that uses it.

Thus, we aim to eliminate the delegation of an ODI to a single SS component and enable the zero trust principle to manage identity and improve security. The following goals should be achieved: $\mathbf{G}_1$. Member's ODI cryptomaterial is not in the sole control of one entity. $\mathbf{G}_2$. Member can trace back the internal initiator of the message sent on Member's behalf through X-Road. $\mathbf{G}_3$. Member can define the access of internal entities to operations on its behalf. $\mathbf{G}_4$. The system is backwards compatible. $\mathbf{G}_5$. No single entity can create a valid signature.

Table 1 maps X-Road characteristics (Sec. 2.1), which can be achieved using the reviewed key management mechanisms (Sec. 2.2). The mapping highlights the key phase and the mechanisms, the implementation of which primarily affects the respective system characteristics. An empty cell in the table refers to no direct effect or the lack of evidence which would confirm the impact.

To achieve our goals, we propose targeting specific system characteristics. For $\mathbf{G}_1$, we aim for decentralisation, trustlessness, portability, and support for multiple users. We focus on multiple-user support and traceability for $\mathbf{G}_2$ and $\mathbf{G}_5$. For $\mathbf{G}_3$, we emphasise preventing privilege escalation. Altogether, the stated goals aim to protect the integrity of the data exchanges within the trusted network by assuring message authenticity and proof of origin.

Table 1. Mapping of system characteristics and mechanisms targeting them.

System characteristic	X-Road	Pre operational phase				Operational phase			
	Related targeted goals	Key gener.	Key derivation		Key distribution	Key storage			Key usage
		Crypto scheme	Factors for key deriv.	Enabled MPC sign.	Point of control	Token form	Wallet form	Keys relationship	Sign.& verif. algorithm
Decentralisation	G1				+				
Trustlessness	G1				+			+	
Portability	G1					+	+		
Multiple users	G1,G2,G5		+	+					
Traceability	G2			+					+
Usability	G4	+			+	+			+
Prevent. priv. misuse	G3			+	+			+	

4.2 System Design

We set up the distributed key management system in X-Road, as depicted in Fig. 3, to evaluate its feasibility. White classes depict current X-Road entities, green classes represent the added DKMS components, and dark grey represents services used to implement the DKMS for X-Road. Currently, each Member has a fully trusted custodian, represented by a Security Server (SS). SS is responsible for ODI's key management and message signing on the organisation's behalf. The key can be either managed by an external service provider or by a Member itself. To eliminate the centralised management, we use threshold signatures to increase the number of key controllers from one to N, where at least K $\leq$ N controllers are needed to create a valid signature. Each controller can validate different policy rules for using its key.

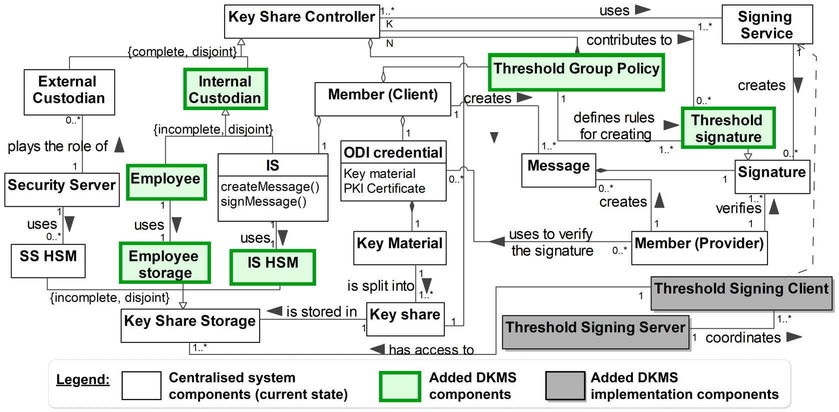

Fig. 3. DKMS for organisational digital identity in X-Road (N – the size of a group, K – threshold number of controllers contributing to a signature, K $\leq$ N).

In the X-Road use-case, we propose to share the Member's identity key material between the three key share controllers which already participate in the process of message exchange: (*i*) Member's information system used for creating a message; (*ii*) Client's employee making a data request so to ensure that the request comes from Clients' identity; (*iii*) Security Server (SS) used by a Member for the message exchange. For multiparty threshold signatures, we use an open-source threshold signing platform. The employees may store their key share on a cryptographic smartcard, while the external custodial (represented by an SS) and Member's IS may use Hardware Security Modules (HSMs) as key share storage. The threshold signing scheme [27] ensures full backward compatibility with the latest X-Road version 7.0.0. The selected scheme requires a trusted dealer for generating and dealing with the individual key shares [27]. We chose the Member's appointed representative (e.g. administrator) to be a dealer for ODI keys; as such, it has control over whom the individual shares are available. After the shares are generated, the dealer must securely delete the private parameters used during the generation of the shares. Fig. 3 depicts the design of the DKMS for the Client's side only. Such implementation assures the integrity of data requests sent on behalf of the Client's ODI. The analogous DKMS can also be implemented at the Provider to ensure the authenticity of the response to the Client's request.

4.3 DKMS for X-Road Implementation

In the DKMS proof of concept implementation [4], we connect the threshold signing platform to X-Road to achieve the proposed design through the supported cryptographic interface PKCS #11 protocol. The chosen threshold signature scheme is compatible with the current scheme, Rivest-Shamir-Adleman (RSA), used in X-Road. The threshold signing platform includes a coordination server and the clients (signing service). The threshold signing server can be deployed either by the Member or by its SS, and clients must be accessible to the signing parties.

To evaluate the quality of the developed proof of concept, first, we conduct the qualitative assessment to check the meeting of the targeted goals and system characteristics. Additionally, we do the quantitative assessment of time for message signing and transfer, compared to other common signing tokens.

Testing scenario. We consider the following two-member X-Road scenario to validate that the proposed DKMS achieves the goals stated in Sec. 4.1. One Member is a governmental data repository provider (e.g. a health agency), and another is a hospital. The hospital (Client) has an information system IS_C. The client uses Security Server SS_C for the data exchange in X-Road. The health board (Provider) has an information system IS_P that stores health-related data. Provider uses Security Server SS_P for the data exchange. IS_C has a user interface for internal usage by three roles: employees who can be doctors, receptionists,

[4] The repository with the proof of concept implementations is available at https://github.com/crocs-muni/xroad-threshold-signatures

and interns who are practising students from the medical school. Interns work in the hospital for a short time during their practice time and leave the hospital afterwards. Interns only have a temporary need to use the system.

Setup. The Client's ODI key material should be distributed between $\mathrm{IS_C}$, $\mathrm{SS_C}$, and employees' roles. Thus, there should be at least five shares, where each role representative holds the same key share. To allow excluding some of the key shares, we propose to derive more than one key share for interns so that after their practice time passes, the interns' key share is defined as 'deprecated' (signing service does not allow to use it anymore) and other pre-generated key shares are made available to the next round of new interns. In particular, we have chosen a 3-out-of-5 threshold signing group.

Goals achievement. During the operational phase, the keys used to create a signature on behalf of ODI are distributed among at least three entities – SS, IS, and some employees ($\mathbf{G}_1$). The set-up threshold group policy used by the signing service guarantees that key share controllers from the threshold group all participate in the signature creation. Thus, Members can partially trace back the entities (to the level of roles) who initiated the message ($\mathbf{G}_2$). An ODI representative who defines a threshold group policy can define among which internal entities key shares are distributed, how the shares are derived, and how to deprecate access. Thus, the signing service using the threshold group policy controls that only the active key shares can be used for the signature creation ($\mathbf{G}_3$). The implementation uses an RSA-based threshold signature signing scheme that is backwards compatible with the standard RSA verification algorithm ($\mathbf{G}_4$). The set-up threshold group policy used by a signing service guarantees that IS, SS and employee contributed to the signature creation ($\mathbf{G}_5$). As a result, the proposed DKMS for X-Road allows us to satisfy the stated goals and enables distribution control over Memeber's ODI during the operational phase of keys.

System assessment. For time measurements, the previously described testing scenario with Client's $\mathrm{IS_C}$, $\mathrm{SS_C}$ and Provider's $\mathrm{IS_P}$, $\mathrm{SS_P}$ is used. All the components are running on a single device as various virtual machines. The Provider uses X-Road's built-in software token (SoftToken) in $\mathrm{SS_P}$, and the Client uses a 3-out-of-5 threshold signing group. The group's coordination server is deployed to $\mathrm{IS_C}$ and connected through PKCS#11 protocol to $\mathrm{SS_C}$. The signing applications are individual running processes that automatically sign any incoming requests. This local deployment allows us to measure the system's throughput with a negligible network delay. In a real-world deployment, the timing would also be affected by the network requests within the signing group. At the expense of implementation changes in X-Road, the request to $\mathrm{SS_C}$ could be partially signed to avoid extra network requests within the group. The Round Trip Time (RTT) measurement starts with the $\mathrm{IS_C}$ requesting data from the $\mathrm{IS_P}$ and stops when the response is received. The mean RTT over 1000 measurements is in Table 2. Also, we have done the exact measurements for the Client using SoftToken, test hardware security module implemented in software (SoftHSM), Yubikey (5C NFC) and a Trusted Platform Module (TPM NTC 7.2.3.1). Using

SoftTokens on both the Client's and Provider's side is a baseline measurement. Excluding a network delay, the tested DKMS introduces less than 200ms overhead compared to the centralised signing solution. Even less overhead (10-60ms) is seen compared to commercial hardware security modules, such as TPM NTC or YubiKey.

Table 2. Round Trip Time (RTT) comparison for Client-Provider data exchange (the Provider uses SoftToken, and the Client's signing token varies; the mean is across 1000 measurements, SoftToken is used as a baseline).

Client's token:	SoftToken	SoftHSM	YubiKey 5	TPM NTC 7.2.3.1	*this work*
mean RTT	82ms	75ms	216ms	260ms	276ms
mean slowdown	1.0x	0.92x	2.65x	3.18x	3.38x

The main limitation of the presented implementation of DKMS for X-Road is relying on a trusted central party (dealer) during the pre-operational phase. Thus, our system design assumes trust in an organisation representative responsible for key generation, certification and key shares distribution. Hence, the presented proof of concept is prone to a single point of failure in the pre-operational key phase when the identity's keys are generated, and the threshold group policy and threshold signing service are not enabled. At the expense of backwards compatibility, using distributed key generation would lift this limitation.

5 Discussion

5.1 Related Work

For managing the keys related to ODI, multiple access control models can be used (e.g. role-based, attribute-based, and discretionary [26]). But to the best of our knowledge, none of the traditional access control models considers the control over keys in view of the zero trust paradigm and the context of cross-organisational data exchange. The closest to our research focus is the architecture vision for the data exchange platform Simpl [2]. There, the authors review three Identity Management (IdM) models: centralised based on PKI [2], hybrid based on PKI and extensions for verifiable credentials [2], and self-sovereign identity with a distributed ledger [2,7,3]. However, the three models differ in the procedure of trust establishment between the organisations, leaving out of scope users who act on behalf of an organisation. Identification of such physical entities is proposed to be handled as a separate task through proprietary identification solutions (e.g. Microsoft Single sign-on, OpenID) [2], internal access control systems, or employee's wallet (digital or smart cards). Therefore, in this paper, the proposed DKMS bridged two IdM systems – (*i*) for end-users of organisational IS and ODI, and (*ii*) for ODIs and cross-organisational data exchange. As a result, the novelty of the proposed DKMS is in its ability to enforce organisational identity policies for the system users through threshold signature, performed in opt-in and backward-compatible way directly applicable to existing information systems. The solution cryptographically guarantees compliance with the policies and, thus, enables zero trust within the IS.

5.2 Limitation

The paper does not consider the legal implications of the proposed design. The proposed DKMS allows the usage of smart cards and personal devices as hardware security modules for key shares. Thus, signatures created by physical custodians can be legally binding in such a way. The reason for such a conclusion is that the same technology is used in Smart-ID [5], an electronic authentication tool used for governmental services in Estonia. As Smart-ID is a qualified signature creation device, the generated signatures are legally binding. However, using a threshold $K < N$ eliminates the non-repudation nature of digital signatures, which could have legal consequences.

The implementation of the proposed DKMS may vary – employees' active approval involvement may be required or semi-automated – depending on the organisation's internal policy. Waiting on the employee's approval is a limitation on one hand, but complies with the four-eyes security principle on the other.

Integrating the proposed DKMS adds overhead for the operator of the threshold signing platform and the individual clients' signing application. Optionally, storing the signing shares on dedicated tokens, such as JavaCards, improves the security, but brings additional financial costs.

6 Conclusion

This paper has investigated how organisations can secure their digital identity (ODI) from abuse by a centralised controller during cross-organisational data exchange. The theoretical contribution of the paper is a key management system for information systems which enables cryptographically assured access policies enforcement that brings zero trust to the control over organisational identity. The practical contribution includes proof-of-concept implementation of the proposed design for the X-Road data exchange system and the library for RSA-threshold implementation [6]. In this study, we have reached several insights.

The existing methods for managing organisational identity either assume complete trust in the selected internal or external entities for operating on behalf of the organisation [26,2] or require extensive infrastructural changes to avoid centralisation if self-sovereign identity ecosystems [2,7] are used. As an alternative, this paper proposes a Distributed Key Management System (DKMS) that allows partial custody by employing threshold signatures to distribute control over the identity between multiple semi-trusted entities. The system ensures

[5] https://www.id.ee/en/article/smart-id/

[6] https://github.com/crocs-muni/pretzel

the authenticity of the messages sent on behalf of an organisation and removes centralisation. The performance results show that threshold signatures are comparable to state-of-the-art key management solutions.

The implementation of DKMS demonstrates that the system design helps to achieve decentralisation and traceability of signed message origin, as well as the ability to involve multiple ODI users whose access rights can be controlled to prevent privilege escalation. Moreover, the proposed system is backwards-compatible without major changes for all the parties involved in the message exchange. Though the system evaluation is done using RSA threshold signature, the approach applies to other signature schemes with existing threshold variants, e.g. ECDSA [14]. Thus, the approach can be generalised to other data exchange systems, where ODI is verified through Public Key Infrastructure (PKI), e.g. Dutch NLX [30]. Even if the system is not prone to a single point of failure but lacks zero trust, the proposed DKMS enable ODI usage policy enforcement that can cryptographically guarantee the authenticity of the data exchanges.

Finally, though the study has been conducted in the context of the problem of centrally issued PKI certificates and certificate-based key management, the review of mechanism and system characteristics is agnostic to the trust model. Thus, the proposed DKMS design can be used for self-sovereign identity-based organisational digital identity where verifiable credentials are issued through a distributed ledger but based on the decentralised public key infrastructure [6].

Acknowledgements We would like to thank reviewers for helpful comments contributing to significant paper improvements. This paper is supported by the European Union under Grant Agreement No. 101087529.

References

1. Gaia-X: A Federated Secure Data Infrastructure. https://gaia-x.eu/
2. Preparatory work in view of the procurement of an open source cloud-to-edge middleware platform. Tech. rep., European Commission (2022)
3. Abraham, A., Koch, K., More, S., Ramacher, S., Stopar, M.: Privacy-preserving eID derivation to self-sovereign identity systems with offline revocation. In: IEEE TrustCom 2021. pp. 506–513 (2021)
4. Alkeilani Alkadri, N., Das, P., Erwig, A., Faust, S., Krämer, J., Riahi, S., Struck, P.: Deterministic wallets in a quantum world. In: CCS 2020. p. 1017–1031. ACM (2020). https://doi.org/10.1145/3372297.3423361
5. Bakhtina, M., Kvapil, J., Svenda, P., Matulevicius, R.: Review of key management mechanisms (Mar 2024). https://doi.org/10.5281/zenodo.10886209
6. Bakhtina, M., Leung, K.L., Matulevičius, R., Awad, A., Švenda, P.: A decentralised public key infrastructure for X-Road. In: ARES 2023. ACM (2023)
7. Bakhtina, M., Matulevičius, R., Awad, A., Kivimäki, P.: On the shift to decentralised identity management in distributed data exchange systems. In: SAC 2023. p. 864–873. ACM (2023). https://doi.org/10.1145/3555776.3577678
8. Barker, E.: NIST SP 800-57. Recommendation for key management (2016)

9. Buck, C., Olenberger, C., Schweizer, A., Völter, F., Eymann, T.: Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust. Computers & Security **110**, 102436 (2021)
10. Cybernetica: Unified eXchange Platform (UXP). https://cyber.ee/
11. Das, P., Erwig, A., Faust, S., Loss, J., Riahi, S.: The exact security of bip32 wallets. In: CCS 2021. p. 1020–1042. ACM (2021)
12. Guthoff, C., Anell, S., Hainzinger, J., Dabrowski, A., Krombholz, K.: Perceptions of distributed ledger technology key management – an interview study with finance professionals. In: IEEE SP 2023. pp. 588–605 (2023)
13. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. **28**(1), 75–105 (2004)
14. Johnson, D., Menezes, A., Vanstone, S.A.: The elliptic curve digital signature algorithm (ECDSA). Int. J. Inf. Sec. **1**(1), 36–63 (2001)
15. Kersic, V., Vidovic, U., Vrecko, A., Domajnko, M., Turkanovic, M.: Orchestrating digital wallets for on- and off-chain decentralized identity management. IEEE Access **11**, 78135–78151 (2023). https://doi.org/10.1109/ACCESS.2023.3299047
16. Komlo, C., Goldberg, I.: Frost: Flexible round-optimized schnorr threshold signatures. In: Selected Areas in Cryptography. pp. 34–65. Springer, Cham (2021)
17. Krimmer, R., Dedovic, S., Schmidt, C., Corici, A.A.: Developing cross-border e-governance: Exploring interoperability and cross-border integration. In: Electronic Participation. pp. 107–124. Springer, Cham (2021)
18. McBride, K., Kamalanathan, S., Valdma, S.M., Toomere, T., Freudenthal, M.: Digital government interoperability and data exchange platforms: Insights from a twenty country comparative study. In: ICEGOV 2022. p. 90–97. ACM (2022)
19. Nair, V., Song, D.: Decentralizing custodial wallets with mfkdf. In: IEEE ICBC 2023. pp. 1–9 (2023). https://doi.org/10.1109/ICBC56567.2023.10174998
20. Nair, V., Song, D.: Multi-Factor Key Derivation Function (MFKDF) for Fast, Flexible, Secure, & Practical Key Management (2023)
21. NIIS: X-Road Documentation. https://docs.x-road.global/
22. NIIS: X-ROAD®. https://x-road.global/
23. Pieter Wuille: BIP 0032. Hierarchical Deterministic Wallets. https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki
24. Preukschat, A., Reed, D.: Self-sovereign identity. Manning Publications, Shelter Island, NY (2021)
25. Rose, S., Borchert, O., Mitchell, S., Connelly, S.: NIST SP 800-207. Zero trust architecture (2020)
26. Sarfaraz, A., Chakrabortty, R.K., Essam, D.L.: Accesschain: An access control framework to protect data access in blockchain enabled supply chain. FGCS **148**, 380–394 (2023). https://doi.org/10.1016/j.future.2023.06.009
27. Shoup, V.: Practical threshold signatures. In: Advances in Cryptology – EUROCRYPT 2000. pp. 207–220. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
28. Soltani, R., Nguyen, U.T., An, A.: Decentralized and privacy-preserving key management model. In: ISNCC 2020. pp. 1–7 (2020)
29. Verizon Business: 2023 data breach investigations report (2023)
30. VNG Realisatie: NLX: Documentation. https://docs.fsc.nlx.io/
31. Windley, P.J.: Learning Digital Identity: Design, Deploy, and Manage Identity Architectures. O'Reilly Media, Incorporated (2023)