Enabling Semi-trusted Proxies for Data Spaces

Nikos Fotiou* and George Xylomenos†
* ExcID
Athens, Greece
fotiou@excid.io
†Mobile Multimedia Laboratory
Athens University of Economics and Business
Athens, Greece
xgeorge@aueb.gr

***Abstract*—Data spaces enable secure and interoperable data exchange across organizational boundaries, typically via context brokers and standardized APIs such as ETSI NGSI-LD. In emerging data space deployments over Information-Centric Networking (ICN), intermediary proxies are introduced to enhance performance, enable decentralized data access, and support content-level filtering. Such proxies face a tension between flexibility and trustworthiness, as they must often transform or redact content en route to consumers. In this paper, we present a novel architecture that distinguishes between trusted publishing proxies and semi-trusted subscribing proxies, enabling both to operate on signed content while preserving data integrity and verifiability. Our solution leverages W3C Decentralized Identifiers (DIDs), specifically the did:self method, and a hash-based selective disclosure signature scheme to allow verifiable partial content delivery. Although designed for ICN-enabled data spaces, our approach is agnostic to the underlying network and applicable to a broad class of distributed data-sharing systems.**

***Index Terms*—Decentralized Identifiers, ETSI NGSI-LD API, Information-Centric Networking, Selective Disclosure**

I. Introduction

Data spaces are an emerging paradigm for enabling secure, interoperable, and self-sovereign data exchange across organizational boundaries. They aim to establish a technical and governance framework in which participants retain control over their data, while allowing authorized third parties to access, process, or enrich it under agreed policies. Data spaces have been investigated in diverse contexts (e.g., [1]–[4]). Within the data spaces paradigm, content providers share data with content consumers via intermediaries known as *context brokers*, which serve as communication and integration points.

A widely adopted specification for implementing data spaces is the ETSI NGSI-LD API [5]. This API defines a RESTful, HTTP-based interface for managing the lifecycle of content entities. These entities are modeled using JSON-LD [6], allowing semantically rich and interoperable representations. The API provides operations to create, read, update, and delete entities, attributes, and relationships, supporting both direct interactions and subscription-based updates. This standard plays a critical role in ensuring semantic and syntactic interoperability across data space participants.

To enhance the flexibility and capabilities of data spaces, additional intermediaries, such as proxies, are increasingly being introduced between the context broker and end-users. These proxies can serve multiple roles: they can optimize performance (e.g., through caching), enforce access control policies, enrich or transform data in transit, and enable new network-level architectures.

In this paper, we focus on proxies that integrate data spaces with *Information-Centric Networking* (ICN) [7] architectures, as explored in our prior work [8]. ICN introduces significant benefits, including improved performance through content-based routing and caching, as well as the possibility to support federated deployments through multiple decentralized brokers. However, this architecture introduces new challenges related to trust. Proxies must often modify the content flowing from the context broker to the consumer, for instance, by filtering attributes based on a consumer's request. As a result, traditional integrity mechanisms, such as end-to-end digital signatures from the original content owner, become impractical: such signatures would either become invalidated by content modifications, or would prevent proxies from performing necessary content transformations.

Beyond proxies, the content owners themselves frequently need to perform fine-grained updates to content, such as modifying a single attribute of an entity. Re-signing the entire content item after each such change is costly and operationally complex. Therefore, a mechanism is needed to allow authorized proxies to sign outgoing data in a verifiable way, while allowing downstream proxies to perform selective disclosure of data attributes.

In this paper, we propose a novel approach that enables secure and verifiable communication through proxies in data spaces. Specifically, we distinguish between two types of proxies: (i) trusted *publishing proxies*, which are explicitly authorized by content owners to sign outgoing data on their behalf, and (ii) semi-trusted *subscribing proxies*, which are empowered to perform downstream attribute filtering, based on consumer requests. Our solution leverages the W3C *Decentralized Identifier* (DID) framework [9] and, in particular, the `did:self` method proposed in [10], to implement verifiable signing rights delegation. For data authenticity and integrity, we adopt the lightweight digital signature mechanism introduced in [11], which was designed for selective disclosure. Although our proposed architecture is applied in the context of data spaces over ICN, it is agnostic to the underlying network substrate. It can be applied in any environment where inter-

mediaries are required to process and partially disclose signed content, while maintaining verifiability and data integrity.

The remainder of this paper is organized as follows. Section II provides an overview of the Data Spaces over ICN architecture and the enabling technologies on which our solution builds. Section III presents the design of our approach in detail. In Section IV, we describe our implementation and report some results from our evaluation. Related work is discussed in Section V. Finally, Section VI concludes the paper and outlines directions for future research.

II. Background

A. Data Spaces over ICN

The architecture presented in [8] explores the integration of data spaces with Named Data Networking (NDN) [12], a prominent realization of Information-Centric Networking (ICN). It aims to combine the semantic interoperability and standardized API support of NGSI-LD-based data spaces with the performance, scalability, and robustness of ICN.

A core feature of the design is the decoupling of content provision and distribution. Content providers use the ETSI NGSI-LD API via a standard HTTP interface to create or update entity representations. These API calls are handled by edge-located proxies, which serve two purposes: (i) maintaining an up-to-date view of the provider's content items, and (ii) advertising content availability in the ICN network via appropriate messages. Content consumers similarly use the ETSI NGSI-LD API to query or subscribe to data. Their requests are processed by proxies, which translate NGSI-LD API calls into the corresponding ICN messages. The proxies then forward these packets through the ICN network to locate and retrieve the requested content. Once retrieved, the content is reconstructed into a valid NGSI-LD response and returned to the consumer via the standard HTTP interface.

Unlike traditional IP-based data space architectures, the ICN-enabled design allows content to be retrieved from multiple potential sources, including in-network caches. Furthermore, ICN supports request aggregation and native multicast, which enables multiple consumers requesting the same data to share a single network transmission, thereby reducing bandwidth consumption and improving scalability.

This architecture supports both semantic interoperability through the NGSI-LD model and efficient content dissemination through ICN primitives, enabling high-performance, federated data spaces that are compliant with the NGSI-LD standard, while benefiting from content-centric optimizations.

B. W3C Decentralized Identifiers and `did:self`

Decentralized Identifiers (DIDs) are a W3C standard for globally unique, cryptographically verifiable identifiers that do not require centralized registration authorities. Each DID is associated with a DID Document, a structured data object that describes the entity's public keys, authentication mechanisms, verification methods, and service endpoints. DID Documents are expressed in JSON and can be retrieved by resolving the identifier using a DID method, which defines the syntax, resolution mechanism, and lifecycle operations for a specific class of DIDs.

Unlike traditional identifiers (e.g., domain names or email addresses), DIDs are designed to be self-sovereign and privacy-preserving. They allow entities to control their identity material directly and rotate keys without relying on external registries. This makes DIDs particularly suitable for decentralized systems and trust architectures that span multiple administrative domains.

The `did:self` method, introduced in [11], offers a lightweight, mechanism for creating, distributing and maintaining self-certifying DIDs. In this method, a DID is derived from the thumbprint of a *JSON Web Key* (JWK) public key [13], and the corresponding DID Document is implicitly constructed using an X.509 certificate created by the DID *owner*. Specifically, the DID *owner* generates an X.509 certificate in which the *Subject Alternative Name* (SAN) extension encodes the `did:self` identifier, and the public key is included in the certificate body. The authenticity and integrity of the resulting DID Document are guaranteed by the digital signature of the certificate itself, without requiring any additional trust anchors or root certificates.

This embedding of DIDs into X.509 certificates enables seamless integration of `did:self` identifiers into existing digital signature frameworks and *Public Key Infrastructure* (PKI) tools. Systems that already support X.509-based authentication can directly verify signatures made by `did:self` entities without any changes to the underlying cryptographic mechanisms. This makes `did:self` particularly attractive for retrofitting decentralized identity features into legacy environments, including the proxy architectures explored in this work.

C. Signing with Support for Selective Disclosure

A key challenge in environments where intermediaries may filter or transform content before delivery is maintaining verifiability of the resulting data, while allowing selective disclosure. Traditional digital signatures bind the signer to the entire payload, making them incompatible with scenarios in which only a subset of the content is revealed to the end recipient – for example, when a proxy filters attributes based on a consumer's access rights or preferences.

To address this limitation, we integrate the solution presented in [10], which introduces a digital signing approach that enables verifiable selective disclosure through the use of cryptographic hashes. Instead of signing the entire JSON object, the signer constructs a list of *disclosures*, where each entry includes an attribute of the content along with a cryptographic salt. A second list is then created by hashing each disclosure entry individually. This list of hash values forms the payload of a *JSON Web Signature* (JWS) [14], which serves as the *integrity proof* for the content (see also Fig. 1). A *Zero-Knowledge Proof* (ZKP)-based approach was also considered in [10]. However, we opted out of the ZKP-based approach, as it required proxies to know the exact order of attributes in the original JSON object.

When a proxy or intermediary wishes to disclose only a subset of the original content, it only includes the selected

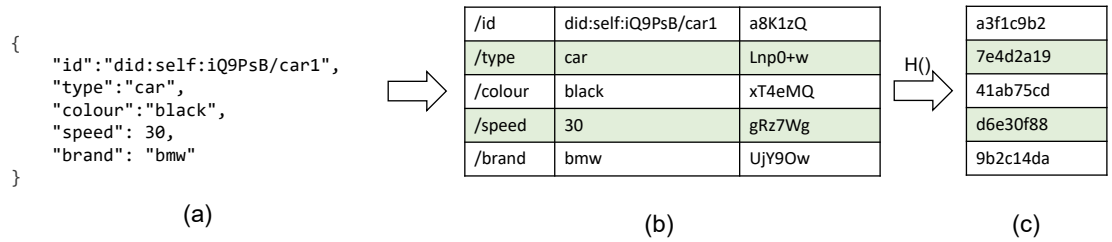

Fig. 1. The signing process adopted by our solution. A JSON object (a) is transformed into a list of disclosures (b) each of which includes an attribute name, an attribute value, and a salt. Each entry is individually hashed (c). The list of hashes is then used as the payload of a JWS.

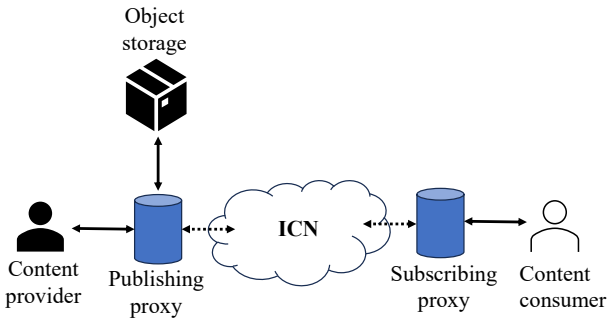

Fig. 2. Overview of the components of the considered architecture.

attributes and their corresponding salt values, along with the generated integrity proof (which includes all hashes and a signature over them). A verifier can then independently recompute the hash values of the disclosed attributes and check that they are included in the payload of the integrity proof. This process ensures that the disclosed subset is authentic and unmodified, even though the full entity is not available.

III. Design

A. Overview

Our design assumes the same entities as the architecture presented in [8] (see also Fig. 2). The system comprises *content providers* and *content consumers*, both operating in traditional IP-based environments, while content is exchanged over an ICN network. The interface between the IP and ICN domains is handled by *proxies*, which are positioned at the edge of the ICN domain. These proxies serve as gateways between the two communication paradigms. They receive HTTP(s)-based NGSI-LD API calls from content consumers and translate them into appropriate ICN packets. Conversely, they retrieve ICN content objects and reconstruct NGSI-LD-compliant HTTP responses before forwarding them to the consumers. This allows seamless interaction between NGSI-LD clients and the ICN-based content delivery infrastructure.

In our solution, each content provider establishes a trust relationship with one or more proxies, referred to as the *publishing proxies* of that provider. These publishing proxies are authorized to act on behalf of the provider and are responsible for signing the transmitted content items. At the other end, each content consumer interacts with the data space through a *subscribing proxy*, which acts as an intermediary, processing consumer requests and delivering filtered responses. A proxy may serve in both roles – acting as a publishing proxy for some providers and as a subscribing proxy for some consumers – depending on its deployment context and trust relationships.

Each content provider is identified by a `did:self` identifier, which is assumed to be well-known or discoverable by consumers. Similarly, each content item is identified by a URL prefixed by the identifier of the provider. Content providers store and manage items using the corresponding NGSI-LD HTTP requests, which are handled by publishing proxies. A content item is encoded using JSON-LD and stored by a publishing proxy into an *object storage* component; this may range from a simple database to a distributed storage system. At the reception of a request from the ICN network, the publishing proxy retrieves the corresponding JSON objects from the object storage component and transforms them into the appropriate disclosures.

To authorize a proxy to act on its behalf, a content provider issues a certificate binding the proxy's public key to the provider's identifier. This certificate is used by the proxy for the signing process described in Section II-C. Content items are transmitted as a list of *disclosures*. The actual responses sent to consumers include only the requested attribute-value pairs of an item and an integrity proof, which enables the consumer to independently verify the authenticity of the received content, despite not seeing the full entity.

Our design relies on several key trust assumptions regarding the behavior of proxies in the system. First, we assume that end-users trust publishing proxies to not maliciously modify the content they sign. These proxies are considered trustworthy to produce valid digital signatures over content disclosures, using the mechanisms described previously. Second, we assume that all end-users trust all other subscribing proxies, to correctly perform the filtering of content items.

B. Setting up and Publishing Proxy Authorization

Our solution begins with a setup phase during which each content provider establishes its identity and configures its trusted publishing proxies. Let $\mathcal{P}$ denote a content provider. Provider $\mathcal{P}$ generates a public/private key pair $(\mathsf{pk}_\mathcal{P}, \mathsf{sk}_\mathcal{P})$, and computes its `did:self` identifier as the thumbprint of $\mathsf{pk}_\mathcal{P}$:

$$\mathsf{DID}_\mathcal{P} = \mathsf{Thumbprint}(\mathsf{pk}_\mathcal{P})$$

This identifier $\mathsf{DID}_\mathcal{P}$ is assumed to be discoverable by any content consumer wishing to interact with $\mathcal{P}$. The provider then self-signs an X.509 certificate $\mathsf{Cert}_\mathcal{P}$, embedding $\mathsf{DID}_\mathcal{P}$ in the *Subject Alternative Name* (SAN) field.

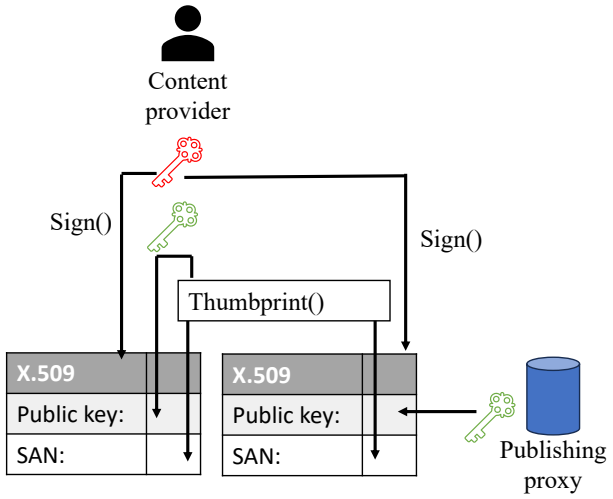

Fig. 3. Proxy authorization process. The proxy obtains a certificate chain which includes two X.509 certificates. Both certificates are signed using the private key of the content provider and include in their SAN field the thumbprint of the content provider's public key.

To delegate signing authority, $\mathcal{P}$ establishes a trust relationship with one or more proxies, each denoted as a *publishing proxy* $\mathcal{X}_i$. Each proxy $\mathcal{X}_i$ also generates its own key pair $(\mathsf{pk}_{\mathcal{X}_i}, \mathsf{sk}_{\mathcal{X}_i})$. The content provider then issues a certificate $\mathsf{Cert}_{\mathcal{P}\to\mathcal{X}_i}$ that includes $\mathsf{pk}_{\mathcal{X}_i}$ and is signed using $\mathsf{sk}_{\mathcal{P}}$ (see also Fig. 3). The SAN field of $\mathsf{Cert}_{\mathcal{P}\to\mathcal{X}_i}$ specifies the namespace over which $\mathcal{X}_i$ is authorized to act. For coarse-grained delegation, the SAN may contain only $\mathsf{DID}_{\mathcal{P}}$, thereby authorizing the proxy to act on behalf of the entire content space of $\mathcal{P}$. For fine-grained access control, the SAN may include a subset $\{c_1, \ldots, c_n\}$ of content item identifiers, explicitly restricting the proxy to those items.

When a publishing proxy signs content disclosures, it includes either (i) the complete certificate chain $\{\mathsf{Cert}_{\mathcal{P}}, \mathsf{Cert}_{\mathcal{P}\to\mathcal{X}_i}\}$, or (ii) a URL from which this chain can be retrieved. The latter option is preferable in ICN environments, as the overhead of an additional fetch can be amortized through in-network caching.

To validate the certificate chain, a verifier must perform the following checks:

1) Apply standard X.509 certificate validation on the chain.
2) Confirm that the SAN of the root certificate $\mathsf{Cert}_{\mathcal{P}}$ equals $\mathsf{DID}_{\mathcal{P}}$.
3) Confirm that the thumbprint of the public key in $\mathsf{Cert}_{\mathcal{P}}$ matches $\mathsf{DID}_{\mathcal{P}}$:

$$\mathsf{Thumbprint}(\mathsf{pk}_{\mathcal{P}}) = \mathsf{DID}_{\mathcal{P}}$$

4) Confirm that the SAN of $\mathsf{Cert}_{\mathcal{P}\to\mathcal{X}_i}$ either equals $\mathsf{DID}_{\mathcal{P}}$ or uses it as a prefix (i.e., it lies within the provider's namespace).

This setup process ensures that publishing proxies are cryptographically linked to the provider's identity and that consumers can independently verify the authorization and scope of proxy-signed content disclosures.

C. Content Filtering and Retrieval

Content consumers interact with the data space by issuing NGSI-LD `GET` requests to retrieve content items. These requests are first received by a subscribing proxy, which translates them into corresponding ICN messages. In accordance with the NGSI-LD specification, consumers may request either specific content items or all items of a particular type. Moreover, the request may include all attributes of the item or specify a subset of attributes to be returned.

As described in the architecture of [8], attribute filtering is performed at the subscribing proxy, i.e., a publishing proxy transmits the whole item and the subscribing proxy is responsible for hiding the attributes not requested by the consumer. Filtering at the subscribing proxy enables higher cache hit ratios within the ICN underlay, as complete content items may be reused to respond to different consumers, even if each consumer requests a different subset of their attributes.

When a content request is routed to a publishing proxy (as determined by ICN name resolution), the proxy executes the request against its local object storage. The retrieved JSON entity is transformed into a list of disclosures, as described in Section II-C. The proxy then generates a digital signature over the hashes of the disclosures (i.e., the integrity proof) and transmits the full set of disclosures.

On the consumer side, the subscribing proxy receives the ICN data packet containing the full set of disclosures. If necessary, it performs attribute filtering based on the original NGSI-LD request. The resulting subset of disclosures is then returned to the content consumer, along with the corresponding integrity proof. Upon receiving the response, the consumer verifies the integrity proof and reconstructs a JSON object. This integrity proof verification process guarantees that the content was produced by an authorized publishing proxy and has not been tampered with in transit. To be more specific, the integrity of the disclosed data is verified against the hashes contained in the payload of the integrity proof, while the certificate chain ensures that the publishing proxy was properly authorized by the content provider.

IV. Implementation and Evaluation

A. Performance Evaluation

We have implemented our solution using the `did:self` Python implementation[1] and the `jwcrypto` library[2]. All cryptographic operations are based on the NIST P-256 elliptic curve. Experiments were conducted on a macOS system equipped with an Apple M4 CPU and 16 GB of RAM[3].

To evaluate the performance of our solution, we generated 1000 random JSON objects, each containing 100 attributes. For each object, we constructed the corresponding list of disclosures and generated an integrity proof using the selective disclosure mechanism described in Section II-C. Table I reports the integrity proof generation times in milliseconds.

[1] https://github.com/excid-io/did-self-py
[2] https://pypi.org/project/jwcrypto/
[3] Our evaluation results can be replicated using the code available at https://github.com/mmlab-aueb/data-proxies

TABLE I
INTEGRITY PROOF GENERATION TIME (MS)

	Time (ms)
Minimum	0.854
Maximum	6.711
Average	2.433

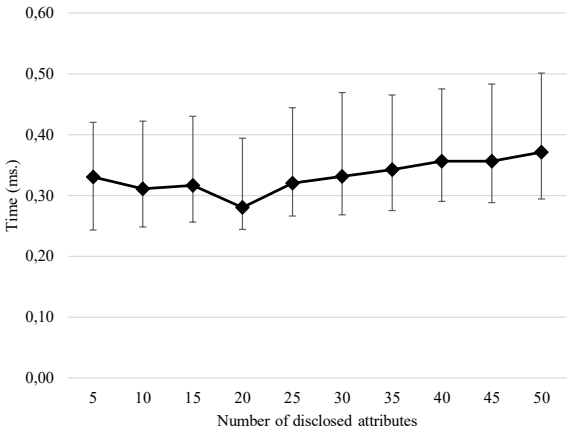

Fig. 4. Min, Max, and Average integrity proof verification time as a function of the number of disclosed attributes.

Next, we evaluated the integrity proof verification performance under partial disclosure. For each JSON object, we randomly selected between 5 and 50 attributes to be revealed, and then verified the resulting object against the corresponding integrity proof. Across all trials (see also Fig. 4), the average verification time was less than 0.5 ms, demonstrating that our approach supports efficient selective verification, even at varying disclosure sizes.

B. Security Evaluation

The primary security objective of our solution is to ensure the integrity of the disclosed attributes received by content consumers. We consider multiple adversarial scenarios and evaluate the resilience of our design against potential threats.

First, consider an attacker capable of monitoring and manipulating network traffic. Using our signing mechanism, any unauthorized modification – such as altering attribute values or injecting forged disclosures – would result in a signature mismatch. Consumers independently verify that the hash values of received disclosures match those in the signed payload. Thus, any tampering by a network-level attacker is easily detectable.

Next, we consider the case of a malicious subscribing proxy. Such a proxy could attempt to suppress selected attributes in a content item before delivering it to the consumer. This is particularly problematic when the omitted attributes are optional, leading the consumer to falsely conclude that they do not exist. Although this attack does not compromise the authenticity of disclosed attributes, it may mislead the consumer regarding the completeness of the entity.

To mitigate this, we propose a slight modification to the signature generation process. Instead of signing only the list of hashed disclosures, the publishing proxy can include in the signature a list of pairs $\{(a_i, h_i)\}$, where a_i is the attribute name and h_i is the hash of the salted value. This prevents any attribute from being silently omitted, as consumers have access to the names of the attributes included in an object.

Another critical security goal is to ensure that only authorized publishing proxies can generate valid signatures. In our design, a content provider generates a `did:self` identifier by creating a public/private key pair $(\mathsf{pk}_{\mathcal{P}}, \mathsf{sk}_{\mathcal{P}})$, and uses this identity to delegate signing authority to selected proxies via X.509 certificates. The probability of two independent providers generating the same key pair (i.e., a key collision) is negligible due to the large key space of elliptic curve cryptography. Consequently, as long as the private keys of the provider and its authorized proxies remain secure, no attacker can forge valid signatures or impersonate a trusted proxy.

V. RELATED WORK

Several research efforts have explored mechanisms for enabling selective disclosure of data (e.g., [15]–[17]). In addition to supporting selective disclosure, our solution incorporates access rights delegation by allowing authorized proxies to generate signatures on behalf of content providers. Furthermore, our approach is built upon the widely adopted *JSON Web Signature* (JWS) standard, facilitating seamless integration and interoperability with existing systems and protocols.

A growing body of research explores enabling proxies to operate on encrypted data, often leveraging techniques such as proxy re-encryption [18], with several efforts specifically targeting ICN environments (e.g., [19]–[21]). In these approaches, proxies re-encrypt content in such a way that individual end-users can decrypt it using their own decryption keys, thereby preserving data confidentiality while supporting flexible content distribution. Similarly, related work applies role-based encryption [22], [23], or attribute-based encryption [24] to allow semi-trusted proxies to implement access control. In these approaches, data is encrypted in such a way that only users abiding by a specific policy can decrypt it. Although our solution does not directly address confidentiality, it is complementary to these works. In particular, these cryptographic techniques can be integrated into our framework to add confidentiality. Our signing mechanism is compatible with encrypted attribute values in disclosures, allowing encrypted content to be selectively disclosed and verified, without modifying the core signing and verification logic.

Our system adopts *Decentralized Identifiers* (DIDs) as a foundational element for establishing trust without relying on centralized trusted entities such as Certificate Authorities. This approach enhances autonomy and resilience but introduces certain trade-offs – most notably, the use of non-human-readable identifiers, since DIDs in our framework are effectively cryptographic digests of public keys. While this design decision aligns with decentralized security principles, it may pose usability challenges in scenarios requiring human-friendly naming. Related work relies on *Identity-Based Encryption* (IBE) where a human-readable identity is used as a public key (e.g., [25], [26]). However, IBE suffers from the so-called key escrow problem as it requires a centralized trusted

entity which knows all private keys. Alternative approaches such as the system proposed in [27], which implements a decentralized trust management framework based on verifiable certificates, offer a compelling trade-off. These approaches retain a degree of decentralization while enabling more readable and structured trust relationships. Such mechanisms could serve as complementary or alternative components in future iterations of our system, depending on deployment requirements and usability considerations.

VI. Conclusions

In this paper, we presented a lightweight and efficient solution for securing proxy-based communication in data spaces. Our approach addresses two core requirements: (i) it enables the authorization of trusted proxies to sign content items on behalf of a content provider, and (ii) it allows semi-trusted proxies to selectively hide attributes of a content item without compromising the verifiability of the revealed content. The proposed mechanism builds on the use of `did:self` decentralized identifiers and a hash-based selective disclosure signature scheme, ensuring both data integrity and fine-grained delegation. Our evaluation shows that the solution incurs negligible performance overhead, making it suitable for real-time or resource-constrained environments.

We have applied our solution in the context of a proxy-based architecture that enables NGSI-LD API interactions over an Information-Centric Networking (ICN) infrastructure. However, the proposed design is fully agnostic to the underlying networking substrate and can be seamlessly integrated into alternative deployment models that require intermediary content filtering and verifiable data delivery.

Acknowledgment

The work reported in this paper has been partly funded by the EU's Horizon 2020 Programme through the subgrant Secure and Efficient Data Spaces (SeEDS) (NGISARGASSO-2024-CALL4-2-SeEDS) of project NGI SARGASSO (grant agreement No 101092887).

References

[1] A. Ionescu, K. Patroumpas, K. Psarakis, G. Chatzigeorgakidis, D. Collarana, K. Barenscher, D. Skoutas, A. Katsifodimos, and S. Athanasiou, "Topio: An open-source web platform for trading geospatial data," in *Proceedings of the International Conference on Web Engineering*. Springer, 2023, pp. 336–351.

[2] I. Koren, S. Braun, M. Van Dyck, and M. Jarke, "Dynamic strategic modeling for alliance-driven data platforms: The case of smart farming," in *Intelligent Information Systems*, S. Nurcan and A. Korthaus, Eds. Springer International Publishing, 2021, pp. 92–99.

[3] S. Scheider, F. Lauf, F. Möller, and B. Otto, "A reference system architecture with data sovereignty for human-centric data ecosystems," *Business & Information Systems Engineering*, pp. 1–19, 2023.

[4] B. Farahani and A. K. Monsefi, "Smart and collaborative industrial IoT: A federated learning and data space approach," *Digital Communications and Networks*, vol. 9, no. 2, pp. 436–447, 2023.

[5] "Context information management (CIM); NGSI-LD API," ETSI Draft, 2023.

[6] Manu Sporny et al., "JSON-LD 1.1, A JSON-based Serialization for Linked Data," W3C Recommendation, 2020.

[7] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[8] Y. Thomas, N. Fotiou, I. Pittaras, and G. Xylomenos, "Secure and efficient data spaces over named data networking," in *Proceedings of the IFIP Networking Conference*, 2025.

[9] Manu Sporny et al., "Decentralized Identifiers (DIDs) v1.0," W3C Recommendation, 2022.

[10] N. Fotiou, V. A. Siris, and G. C. Polyzos, "did:self – A registry-less DID method," in *Proceedings of the International Workshop on Trends in Digital Identity (TDI)*, 2025.

[11] N. Fotiou, G. Xylomenos, and Y. Thomas, "Data integrity protection for data spaces," in *Proceedings of the 17th European Workshop on Systems Security (EuroSec)*, 2024, p. 44–50.

[12] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the ACM CoNEXT*, 2009.

[13] M. Jones and N. Sakimura, "JSON Web Key (JWK) Thumbprint," Internet Requests for Comments, IETF, RFC 7638, September 2015. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7638.txt

[14] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Signature (JWS)," Internet Requests for Comments, IETF, RFC 7515, May 2015. [Online]. Available: https://tools.ietf.org/html/rfc7515

[15] K. Saito and S. Watanabe, "Lightweight selective disclosure for verifiable documents on blockchain," *ICT Express*, vol. 7, no. 3, pp. 290–294, 2021.

[16] A. De Salve, A. Lisi, P. Mori, and L. Ricci, "Selective disclosure in self-sovereign identity based on hashed values," in *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, 2022, pp. 1–8.

[17] R. Mukta, S. Pal, S. Mishra, H.-Y. Paik, S. S. Kanhere, and M. Hitchens, "A blockchain-based interoperable architecture for iot with selective disclosure of information," in *Proceedings of the IEEE Pacific Rim International Symposium on Dependable Computing (PRDC)*, 2023, pp. 53–63.

[18] R. Canetti and S. Hohenberger, "Chosen-ciphertext secure proxy re-encryption," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2007, p. 185–194.

[19] K. Suksomboon, A. Tagami, A. Basu, and J. Kurihara, "In-device proxy re-encryption service for information-centric networking access control," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, 2018, pp. 303–306.

[20] R. Simõcs da Silva and S. D. Zorzo, "On the use of proxy re-encryption to control access to sensitive data on information centric networking," in *Proceedings of the International Conference on Information Networking (ICOIN)*, 2016, pp. 7–12.

[21] M. Sepehri and A. Trombetta, "Secure and efficient data sharing with atribute-based proxy re-encryption scheme," in *Proceedings of the International Conference on Availability, Reliability and Security (ARES)*, 2017.

[22] N. H. Sultan, V. Varadharajan, S. Camtepe, and S. Nepal, "An accountable access control scheme for hierarchical content in named data networks with revocation," in *Computer Security – ESORICS 2020*. Springer International Publishing, 2020, pp. 569–590.

[23] Y. Miao, F. Li, X. Jia, H. Wang, X. Liu, K.-K. R. Choo, and R. H. Deng, "Reks: Role-based encrypted keyword search with enhanced access control for outsourced cloud data," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 3247–3261, 2024.

[24] P. He, K. Xue, J. Yang, Q. Xia, J. Liu, and D. S. L. Wei, "Fase: Fine-grained accountable and space-efficient access control for multimedia content with in-network caching," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4462–4475, 2021.

[25] X. Zhang, K. Chang, H. Xiong, Y. Wen, G. Shi, and G. Wang, "Towards name-based trust and security for content-centric network," in *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, 2011, pp. 1–6.

[26] B. Hamdane, S. G. El Fatmi, and A. Serhrouchni, "A novel name-based security mechanism for information-centric networking," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2014, pp. 2928–2933.

[27] T. Yu, X. Ma, H. Xie, Y. Kocaoğullar, and L. Zhang, "A new API in support of NDN trust schema," in *Proceedings of the ACM Conference on Information-Centric Networking (ICN)*, 2023, pp. 46–54.