Pseudonymity for Personal Data Stores: Pseudonymous WebIDs and Decentralized Identifiers

Gertjan De Mulder(✉) and Ben De Meester

IDLab, Department of Electronics and Information Systems, Ghent University – imec, Technologiepark-Zwijnaarde 122, 9052 Ghent, Belgium
{gertjan.demulder,ben.demeester}@ugent.be

Abstract. Personal Data Stores like Fedora and Solid let users become data holders, controlling their personal data and Web interactions through interoperable standards. Pseudonyms protect privacy during data sharing while still allowing holders to later prove their true identity, making them key privacy-enhancing tools. However, pseudonyms are rarely tackled in existing decentralized personal data sharing standards. In this paper, we present, analyze, and evaluate pseudonymity methods within Solid – a maturing set of personal data sharing standards – applied to a job application use case. This use case consists of three flows: a pseudonym generation flow, a diploma verification flow using that pseudonym and data minimization using the Verifiable Credential standard, and a Proof of Ownership identity binding between the pseudonym and the user's true identity. We compare two pseudonym generation solutions: a Solid-native solution that depends on an external party to lease (Web-resolvable) pseudonyms, and a solution that leverages a static resolving method (DID:Key) to generate ephemeral pseudonyms. The data flow diagrams, and STRIDE and LINDDUN analysis indicate that static identifiers are better for pseudonymous use cases, as they avoid reliance on external parties. The requirement validation show both solutions meet most needs, though the WebID solution remains observable and the DID:Key solution lacks support for deleting or managing pseudonyms. With this pseudonymity work, we aim to provide a next step to combine personal data storage incentives with Wallet incentives (such as those put forward by the EUDI).

Keywords: DID · Personal Data Stores · Pseudonymity · Solid · Verifiable Credentials · WebID

1 Introduction

The amount of control centralized platforms exhibit emphasizes the need and importance for users to regain their control and, thus, their privacy [17,18]. This resulted in an uprise of **personal data stores (PDS)**: interoperable decentralized ecosystems where **users become data *holders***, exercising control over

B. Coppens et al. (Eds.): ARES 2025 Workshops, LNCS 15997, pp. 111–129, 2025.
https://doi.org/10.1007/978-3-032-00639-4_7

their personal data and their interactions on the Web. Maturing PDS standards are the Fedora Repository[1] and the Solid protocols [2].

This decentralization effort has expanded to the term Self-Sovereign Identity (SSI) [32], where users are in control of their digital identity without depending on a (central) authority. This technically implies that **all interactions between a data holder and a data *verifier*** (i.e., the actor that the holder interacts with) **should also be decentralized**, i.e., both holder and verifier have a choice in how they identify and authenticate themselves, and how to authorize other parties (this as opposed to, e.g., the OIDC de facto industry standard [1], where the verifier specifies which identity provider(s) can be used by the holder). This has given rise to the European Digital Identity (EUDI) Wallet [15]: a set of specifications and (reference) implementations to manage personal credentials such as driving license and diploma from your personal device.

PDS standards are related to SSI and EUDI Wallets, but not the same: PDS is use-case agnostic, focussed on interoperability and meant to store and share any kind of data (not only credentials, not only about the holder), whereas the EUDI Wallet is focussed on specific holder credentials, used within authentication flows.

Both Fedora and Solid propose a Web-resolvable identity to represent a holder. Specifically, the identifier (i.e., a *WebID*) is an HTTP URL that resolves to an identity document on the Web (i.e., a WebID Profile Document) describing the holder. A WebID can be obtained by registering one at an Identity Provider (IDP), or by directly leasing a domain name from a Domain Name Space (DNS) registrar.

Being able to globally and uniquely identify entities enables attribution (by associating claims), reputation building, and accountability [9]. However, using the same identifier for different purposes, within different contexts, and across various services increases the risk of tracking and profiling [28].

Pseudonymity is an important privacy-enhancing technology [16]. Holders can protect their privacy when interacting on the Web by using a *pseudonym* that reveals no identifying information, unless disclosure is required [13,30]. However, pseudonymity is rarely tackled in existing PDS standards.

We consider the following motivating use case.

Alice, a data holder who controls her diploma data, is interested in a job requiring a Master's degree. Being aware of gender bias in the hiring process, the recruiter allows applicants to prove their degree without revealing further personal information. Alice wants to avoid using her public identifier (e.g., her public WebID https://id.flanders.be/Alice), as it may expose more data than necessary[2]. *To minimize bias, she chooses to apply using a pseudonym.*

[1] https://fedorarepository.org/.

[2] This example uses gender-based bias for clarity although one can assume that gender-based information is rarely encoded in an identifier. However, in an international setting, one could imagine a similar kind of bias towards applicants of very different origin – and thus using very different domain names in their identifier. Similarly, the presented use case abstracts away additional data interactions typically needed for applicant background checks.

In this paper, we address the following research question: "How can we introduce pseudonymous interactions in a WebID-based system, and how does this affect a user's privacy?".

After introducing our background in Sect. 2 and related works in Sect. 3, we introduce the process flows for the use case: a pseudonym generation flow, a diploma verification flow using that pseudonym and data minimization using the Verifiable Credential standard, and Proof of Ownership method to prove the binding between the pseudonym and the user's true identity (Sect. 4). In Sect. 5, we compare two pseudonym generation solutions: a WebID-native solution that depends on an external party to lease (Web-resolvable) pseudonyms, and a solution that leverages a statically resolvable identifier using the DID:Key method to generate ephemeral pseudonyms. We present the implementation in Sect. 6. We then evaluate and compare these solutions based on a functional, a security, and a privacy evaluation in Sect. 7. We conclude in Sect. 8.

2 Background

In this section, we introduce the entities and concepts relevant for the discussion of this paper. The relationship between those entities is visualized in Fig. 1.

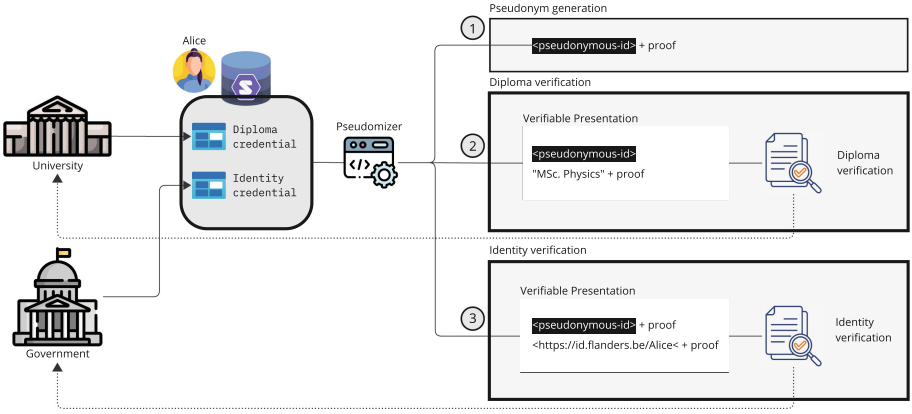

Fig. 1. Pseudonymous job application use case, involving three process flows. First, holder Alice creates a pseudonym. Second, Alice discloses a minimal set of required attributes (i.e., her diploma title) under that pseudonym, to apply for a job. The verifying recruiter can validate with the issuing university that the diploma is genuine. Third, assuming that Alice's pseudonym was selected for the job, Alice proves her governmentally issued identity and the binding with the pseudonym that was selected for the job. The recruiter can validate the pseudonym locally and the identity with the issuing government.

2.1 Entities

We adhere to the following definitions of More et al. [27], unless explicitly stated otherwise:

- **Issuer**: Issuers issue signed credentials to users, e.g., the *university* and *government* are Issuers of diploma Credentials and identity Credentials, respectively.
- **Credential**: A digital and cryptographically verifiable set of claims that is issued by an Issuer to a Holder [36].
- **Holder/User**: A user that manages credentials, which it can use to present verifiable sets of claims to a Verifier, e.g., *Alice* is a Holder that manages her diploma and identity Credentials.
- **Pod**: A personal data store on which users can store digital documents such as diploma and identity Credentials.
- **Pseudomizer**: A service or application that allows Holders to create and use a pseudonym when interacting with a verifier. We assume the Pseudomizer to run on the Holder's personal device.
- **Verifier/Service Provider/Relying Party**: An actor in the network that typically provides a service to a Holder if that Holder can prove specific claims, e.g., the *recruiter* verifies Alice's diploma when she applies for a job.

2.2 Data Minimization

The *data minimization principle*[3] requires the processing of personal data for each purpose to be *adequate* (sufficient to properly fulfil your stated purpose), *relevant* (has a rational link to that purpose), and *limited* (you do not hold more than you need for that purpose). Our use case (Sect. 1) shows the relevance of pseudonymity, based on the data minimization principle and multiple overlapping and complementary use cases stemming from Flemish career-related information sharing via Athumi[4] (the Flemish data utility company) and from the DID use cases[5].

Without a data minimization approach in place, a recruiter can request an applicant to provide access to the entire diploma. Not only does this increase the risk for outsider attacks (e.g., when the entire degree document has been compromised) but also the risk for attacks from the inside: for example, the recruiter can use the information for purposes for which the applicant did not agree (e.g., discovering more information about the applicant by looking up their name, university, etc.).

[3] GDPR Article 5(1)(c).
[4] https://athumi.be/en/personal-smart-data-spaces/my-career.
[5] https://www.w3.org/TR/did-use-cases/#featureBenefitGrid.

2.3 Selective Disclosure

Selective disclosure is the ability to reveal individual claims from a credential [27]. This minimizes the risk of privacy breaches and prevents unauthorized access to sensitive data. For example, to be adequate and relevant for the purpose of selecting an applicant for a job that is known to suffer from gender bias, the recruiter requires applicants to selectively disclose only the degree title (e.g., "Master of Physics"), rather than an entire diploma. *Zero-knowledge proofs* can be used to take selective disclosure a step further to only request *predicates* on an attribute (e.g., age > 18), which allows to demonstrate the validity of a statement without revealing the actual value of the attribute [27].

3 Related Work

We first discuss decentralized identity standards on the Web (sometimes coined as *user-centric* identity standards[6]), and their relation to (existing and new) authentication protocols (Sect. 3.1). We focus on open standards and relevant technologies are highlighted in bold. Then, we discuss privacy-related state-of-the-art of these standards, focusing on pseudonymity (Sect. 3.2).

3.1 Decentralized Identity

There are currently two main decentralized identity standards: WebID, and Decentralized Identifiers (DID).

The Solid Protocol [2] leverages Web standards governed by multiple W3C working and community groups, each focusing on different aspects of the ecosystem: identity using WebID [3], authentication using Solid-OIDC [12], and storage using the Linked Data Platform (LDP) [10]. A server implementation that provides the storage component is called a *pod*. **A WebID is a Web-resolvable URI** that resolves to a WebID Profile Document [3]. Thus, WebIDs require a domain registrar to register and look up the identifier over HTTP(S).

The Decentralized Identity Foundation (DIF)[7] develops the technical foundations to establish Self-Sovereign Identity (SSI) [32] where holders are in control of their digital identity, without depending on a (central) authority. To this end, Decentralized Identifiers (DIDs) [35] are standardized. Different DID methods exist[8] to resolve DIDs to DID documents, e.g., `did:ethr`-identifiers resolve to entries on the Ethereum blockchain, while `did:web`-identifiers can be transformed into Web-resolvable URIs, making them closely related to WebIDs.

Many DID methods rely on registries based on distributed ledger technologies (e.g., blockchain, distributed hash tables, etc.), but **static DID methods**[9] avoid the need for complex infrastructures, external dependencies, and network

[6] To keep consistency with Sect. 2, we will consistently refer to users as holders.
[7] https://identity.foundation/.
[8] https://w3c.github.io/did-spec-registries/#did-methods.
[9] https://w3c-ccg.github.io/did-method-key/.

operations, as they contain sufficient information to construct the corresponding DID Document. When used as short-lived identifiers, static DID methods are particularly suitable for pseudonymity use cases[10]. `did:peer`[11] and `did:key`[12] are common static DID methods. While `did:peer` allows for embedding multiple keys and service endpoints, `did:key` is minimized, only allowing for embedding a single public key in the identifier. Creating a `did:key` DID requires generating a new cryptographic key pair: the DID is a specific encoding of the public key of that generated key pair.

DIDs are typically used in conjunction with W3C recommended **Verifiable Credentials (VC)** [36]: a data model for cryptographically verifiable digital credentials. Within the VC standard, the term Verifiable Presentation (VP) is introduced as a package selectively created by the holder to present one or more VCs to a verifier, possibly with **proof of authenticity and selective disclosure of only the necessary claims**. For the scope of this paper, we focus on the VC open standards and not other (compatible) digital credential formats such as JSON Web Tokens (JWTs), and [36].

The European Digital Identity (EUDI) Wallet is a European Union initiative that aims to solve **(high-level) requirements for decentralized identity**, part of the eIDAS 2.0 regulation (Regulation (EU) 2024/1183) [15]. The EUDI Wallet is typically implemented as a secure mobile application that allows individuals to store, manage, and selectively disclose digital credentials and attributes. It uses DIDs and VCs as interoperable standards for identifiers and verifiable credentials.

Authentication protocols currently proposed in these decentralized identity standards typically rely on a trusted third party (typically called an Identity Provider or IDP), and are often aligned with the OpenID-Connect (OIDC) standards [1]. WebID authentication currently uses Solid-OIDC: an OIDC extension that allows holders to choose their own IDP[13]. The EUDI wallet uses OIDC for Verifiable Presentations (OIDC4VP)[14]: an OIDC extension that allows verifiers to request specific VPs from holders.

Initial WebID authentication was based on mutual TLS (mTLS) [8], however, this requires X.509 certificates for both server and clients, for which (browser-)support is limited. Other related authentication protocols are typically peer-oriented (e.g., based on DIDComm[15]), which makes them fit less in a Web-based context.

[10] https://ref.gs1.org/docs/2025/VCs-and-DIDs-tech-landscape#layerComparison.
[11] https://identity.foundation/peer-did-method-spec.
[12] https://w3c-ccg.github.io/did-method-key/.
[13] In OIDC, IDPs must be preregistered by the verifier, e.g., many applications allow logging in via Google, Facebook, or Microsoft, but you are not able to choose your own IDP to log in with.
[14] https://openid.net/specs/openid-4-verifiable-presentations-1_0.html.
[15] https://identity.foundation/didcomm-messaging/spec/.

3.2 Pseudonyms in Decentralized Identity

A security and privacy threat analysis of Solid's identification, authentication, and authorization protocols using LINDDUN [13] shows that the Solid protocols mitigate 54% of the security threats and only 4% of the privacy threats [26].

In general, privacy is an open issue for the WebID specification [38]. *Anonymous WebIDs* via a trusted third party (TTP) has been studied by Heitmann et al. [21], however, this is not suitable for pseudonymity, as it excludes proving the association between the original WebID and an anonymous WebID [20].

Meanwhile, DIDs and VCs are put forward as important technological specifications to have a secure decentralized (authentication) protocol [6].

Selective disclosure is extensively applied and tested in decentralized identity, typically for privacy-preserving (WebID) authentication [7,21,39]. Sharing a minimal set of attributes protects the privacy of the holder, however, the WebID of the VC's holder still needs to be disclosed for authentication. In recent VC standardization efforts, selective disclosure has been natively supported in crypto suites, i.e., selectively disclosed VCs can be locally verified via cryptographic means and without need for additional interactions with the issuer, holder, or other third party. Examples for such crypto suites are those using Elliptic Curve DSA (ECDSA) [23] or an extension of Boneh âĂŞ Boyen âĂŞ Shacham (BBS) [5] that allows for selective disclosure and zero-knowledge proofs (BBS Cryptosuite) [4].

The EUDI presents a set of *pseudonymity requirements*[16] on top of GDPR, and general privacy requirements have been posed by More et al. [27].

Typical pseudonymization approaches involve a TTP or some form of distributed ledger that generates and manages the pseudonyms [14,25,29,37], of which pairwise pseudonymous identifier flow is integrated in OIDC (requiring trusting a TTP), and where the European Blockchain Services Infrastructure (EBSI) provides a Key Attestation specification[17] (requiring trusting an EBSI Key Attestation issuer). TTPs typically keep track of the mapping between the original identifier and the generated pseudonymous identifier [16,19], enabling them to re-identify pseudonyms. To combat potential insider attacks, "data-owner based pseudonymization" – in which the data owner creates the pseudonyms – is proposed [19].

3.3 Requirement Analysis

Based on the above related work, we derive minimal requirements for a digital pseudonym. On the legal side, we focus on the EU's GDPR and EUDI Wallet pseudonymity requirements. We extend these requirements with related work on pseudonymity [27].

[16] https://eu-digital-identity-wallet.github.io/eudi-doc-architecture-and-reference-framework/1.9.0/annexes/annex-2/annex-2-high-level-requirements/#a2311-topic-11-pseudonyms.

[17] https://code.europa.eu/ebsi/json-schema/-/tree/main/schemas/vcdm2.0/key-attestations.

- **R01: Purpose**: Personal data shall be collected for specified, explicit and legitimate purposes (GDPR, [27]: LREQ1).
- **R02: Data Minimization**: Personal data collected shall be limited to what is necessary in relation to the purposes for which they are processed (GDPR, [27]: LREQ2)
- **R03: Consent**: No credentials should be disclosed or shown unless a holder explicitly operates the wallet software for that purpose (GDPR).
- **R04: Pseudonym generation** is possible (EUDI Wallet: PA_01).
- **R05**: **Pseudonym generation is independent** of the verifier or pod provider (EUDI Wallet: PA_03).
- **R06**: **Multiple pseudonyms can be generated** for the same verifier (EUDI Wallet: PA_04).
- **R07**: **Attach information to a pseudonym** (e.g., an alias) (EUDI Wallet: PA_05).
- **R08: Delete a pseudonym** (EUDI Wallet: PA_07).
- **R09: Manage pseudonyms**. Inform users about when and which verifier used their pseudonyms (including canceled or unsuccessful transactions) (EUDI Wallet: PA_08).
- **R10: Overview of pseudonyms**. Users can see all existing pseudonyms, including the associated verifiers (EUDI Wallet: PA_09).
- **R11: Unlinkability** between the identity and (between) its pseudonym(s) (GDPR, [27]).
- **R12: Unobservability**: Only the user and the verifier learn that a showing takes place. Neither the issuer nor any other authority (e.g., the government) should learn where the user is using their credentials, or even that the user is using their credentials (GDPR, [27]).

4 Process Flows

Alice can protect her privacy in a job application use case by controlling what information she shares: her identity and diploma data. This is achieved by i) using a pseudonym to hide her true identifier, and ii) disclosing only the minimum verifiable degree information, signed with that pseudonym. If selected, she can later prove the link between her pseudonym and real identity.

The use case has three process flows. *Flow 1: Pseudonym Generation*: Alice uses the Pseudomizer to generate her pseudonym for the next flows. *Flow 2: Diploma Verification*: Alice hides her true identity using a pseudonym and provides the minimal set of diploma attributes required by the recruiter. *Flow 3: Identity Verification*: Alice proves the binding between her true and pseudonym to the recruiter. In Flow 2 and 3, we rely on crypto suites to attest and locally verify the claims. We provide a high-level Data Flow Diagram representing the entities, interactions, and assets needed to handle these flows (Fig. 2).

We assume the following: i) Alice has a pod linked to her governmentally issued WebID; ii) Alice obtained her government-issued identity credential and

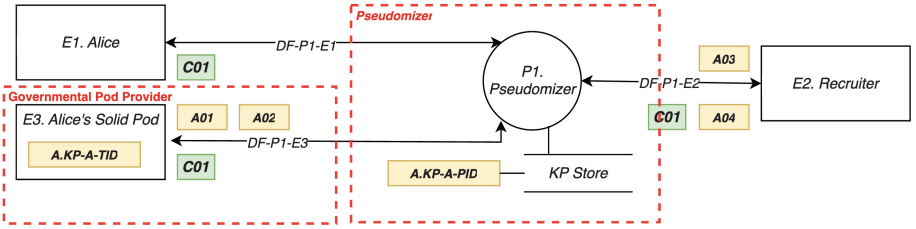

Fig. 2. High-level Data Flow Diagram. Rectangular boxes represent entities, and circles represent processes. The edges between them resemble data flows. Green labels indicate *security controls* that apply to a particular data flow, yellow labels represent *assets*, at rest, or transferred over a data flow. (Color figure online)

university-issued diploma credential and stored them in her pod; iii) all interactions occur over a secure channel (TLS, indicated by the green-colored tag `C01`); and iv) a new pseudonym is generated per job application.

The following assets are part of one or more flows.

- `A01`: a *diploma credential* containing sensitive and personally identifiable information.
- `A02`: an *identity credential* containing personally identifiable information.
- `A03`: a verifiable presentation (the *diploma VP*) that contains selectively disclosed attributes from the diploma credential (`A01`) as minimally required by the recruiter (e.g., only the "degree title").
- `A04`: a verifiable presentation to prove the binding between Alice's identity and her pseudonym (the *identity binding VP*).
- `A.KP-A-TID`: the key pair associated with Alice's identity, issued by her government and stored on her pod.
- `A.KP-A-PID`: the key pair associated with Alice's pseudonym, generated and managed by `P1. Pseudomizer`.

During the *pseudonym generation flow*, `E1. Alice` uses `P1. Pseudomizer` to generate a new pseudonym (and `P1. Pseudomizer` generates a new key pair in the process).

During the *diploma verification flow*, `E1. Alice` uses `P1. Pseudomizer` (`DF-P1-E1`) to apply for a job at `E2. Recruiter`. `P1. Pseudomizer` fetches `E1. Alice`'s *diploma credential* (`A01` over `DF-P1-E3`), creates a minimized diploma credential (using selective disclosure) under the previously generated pseudonym, and presents as a (pseudonym-signed) VP it to `E2. Recruiter` (`A03` over `DF-P1-E2`). `E2. Recruiter` can verify the diploma VP locally, using standardized VC cryptographic methods that support selective disclosure.

During the *identity verification flow*, `E1. Alice` uses `P1. Pseudomizer` (`DF-P1-E1`) to send an identity binding credential to `E2. Recruiter`. `P1. Pseudomizer` fetches `E1. Alice`'s *identity credential* (`A02` over `DF-P1-E3`), selectively discloses identifying attributes from the identity credential, binds them with the previously generated pseudonym, and sends that to `E2. Recruiter`

(A04 over DF-P1-E2). E2. Recruiter can verify the identity binding credential locally without additional interactions, using standardized VC cryptographic methods.

5 Solution

In this paper, we focus on a comparison of two pseudonym generation solutions: a WebID-native solution that depends on an external party to lease (Web-resolvable) pseudonyms, and a solution that leverages a statically resolvable identifier using the DID:Key method to generate ephemeral pseudonyms. Both solutions apply the same data minimization methods for the diploma data, and the pseudonym document only contains a public key to later prove the identity binding. The key difference is in creating and using the pseudonymous identifier.

Our first solution uses a third-party IDP that leases URIs to be used as pseudonymous WebIDs (Fig. 3): a new keypair is generated by P1. Pseudomizer of which the public key is subsequently forwarded to P1.2 IDP Pseudo (over DF-P1.PG-P1.2). P1.2 IDP Pseudo leases a URI that only discloses the newly generated public key when being resolved, effectively creating a pseudonymous WebID (A05). The recruiter resolves the identity document from that IDP (A05 over DF-P1.2-E2) to obtain the public key material needed to verify the digital signatures of both the diploma VP and identity binding VP identity document (cfr. DF-P1.2-E2).

Our second solution uses the static did:key method to generate a pseudonym: a new keypair is generated by P1. Pseudomizer of which the public key is encoded in the did:key identifier. Hence, no new dataflow is required for the DID:Key solution (Fig. 2): The recruiter does not need additional interactions to resolve the identity document and can locally verify the digital signatures of both the diploma VP and identity-binding VP.

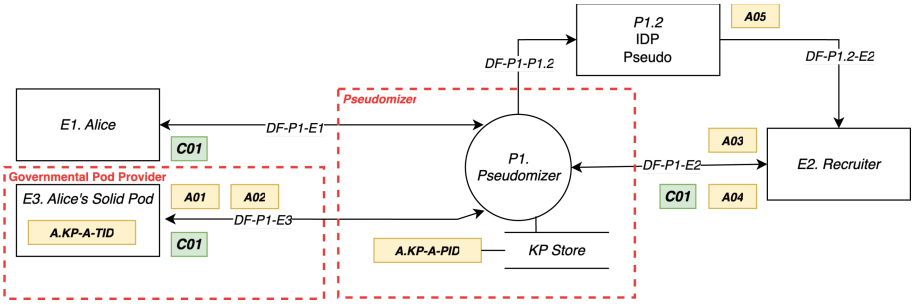

Fig. 3. Data Flow Diagram: WebID solution

We provide **identity binding** by enabling Alice to provide *Proof of Ownership* – a method that allows proving ownership over an asset by demonstrating

control over the private cryptographic key associated with that asset [11] – for her governmentally issued identity, her generated pseudonym, and a self-asserted claim associating her identity with her pseudonym.

Alice creates two cross-signed VCs that both explicitly attest the link between the true identity and the pseudonym. However, one is fetched from Alice's pod, signed using the private key associated with Alice's true identity ($VC_1 = vc(sk_{true}, ID_{true}, ID_{pseudo})$) and the other is signed by `P1. Pseudomizer` using the private key associated with her pseudonym ($VC_2 = vc(sk_{pseudo}, [ID_{true}, ID_{pseudo}])$). These VCs are then encapsulated into a VP, together with a selectively disclosed identity credential that only discloses Alice's true identifier ($VP_{binding} = vp(sk_{true}, [VC_1, VC_2, vc(sk_{true}, ID_{true})])$). As this VP consists of VCs expressing the identity binding between two different key pairs, which is only possible by an entity that controls both key pairs, *Proof of Ownership* is provided.

6 Implementation

To showcase the feasibility of the proposed solutions, we implemented both pseudonymity solutions in TypeScript, MIT-licensed open source, and available at[18]. We designed a generic VC actor allowing to perform common operations of a credential lifecycle (i.e., creating, signing, deriving, and verifying). This actor also abstracts the cryptographic suites and the controlling identity (i.e., the identity used when performing credential operations), allowing us to create a variety of concrete class implementations, each employing different identity types (e.g., WebID, DID) and cryptographic suites (e.g., `eddsa-rdfc-2022`, `bbs-2023`).

We also performed a preliminary performance evaluation on a local machine[19], where we benchmarked the use case's credential operations and measured the execution time. The results are available online[20]. We ran an experiment simulating the steps of the proposed use case, averaging the results over 100 runs: when the recruiter verifies the diploma credential, the mean execution time is 91.15 ms for the WebID solution, and 44.35 ms for the DID:Key solution. This is because `did:key` identity documents can be directly resolved from the identifier, while WebID Profile Documents need to be fetched over the network. We can conclude that both solutions remain feasible on commodity hardware.

7 Evaluation

After validating which requirements our system adheres to, we perform an initial security (Sect. 7.2) and privacy threat analysis (Sect. 7.3) and discuss (Sect. 7.4).

[18] https://github.com/SolidLabResearch/pseudonymity-demo.
[19] MacBook Pro 16-inch (2021) with Apple M1 Max (10 cores) and 32GB of RAM.
[20] https://github.com/SolidLabResearch/pseudonymity-demo/blob/main/performance/PERFORMANCE-preliminary-analysis.ipynb.

7.1 Requirement Validation

Alice is in control when to engage with the recruiter (R03), and for which purpose (i.e., only for the purpose of job solicitation, R01). When doing so, she can select which part(s) of her diploma to share (R02), verifiable using a generated pseudonym (R04, R07). The pseudomizer is independent of the service provider or pod provider (R05), and does not limit Alice in how many pseudonyms she can generate, even for the same service provider (R06). In the case of a leased WebID, the IDP could allow for pseudonym deletion and management, which is not possible for the statically generated DID using the `did:key` method (R08, R09). In both cases, an overview of which pseudonym was generated for which service provider could be created (R10). The pseudomizer allows to create unlinkable pseudonyms, as only a uniquely generated public key is exchanged with each pseudonym (R11). In the case of a leased WebID, the IDP could observe which service provider is using which pseudonym, as resolving for the public key involves an HTTP request to the IDP (R12). Where both solutions cover most requirements, the leased WebID solution does not cover R12 due to the resolvance over HTTP, whereas the DID:Key solution does not cover R08 and R09 due to the ephemeral nature of the `did:key` method.

7.2 Security Analysis

To evaluate the security of our solution, we perform a STRIDE analysis [33] – one of the most widely used threat modeling approaches – where we identify possible attack scenarios and provide potential mitigations, based on the applicant's perspective (as other perspectives are deemed out of scope for this paper). We scope our security threat analysis to the application layer of the TCP/IP model, covering the session, presentation, and application layers of the OSI model), and assume all Web interactions are done over uncompromised HTTPS channels.

Spoofing. The risk of impersonation where an attacker attempts to act like someone else (e.g., authenticating as another user or using another user's credentials). On the one hand, this can happen by *acquiring control over the identity's key pair*, stored on the holder's PDS (`A.KP-A-TID`)[21]. In the WebID solution, this pseudonym key pair is stored by the Pseudomizer's Web service (which, due to its remote access possibilities, inhibits a larger threat). In the DID:Key solution, the key pair resides on the holder's local device. On the other hand, this can happen due to *first-party fraud*: Alice can create a derived diploma credential from a friend's (better matching) diploma credential. This can be **mitigated by relying on trusted issuers for these kinds of VPs, or by requesting proof of bound attributes**: assuming the identity document and diploma contain matching identifying data (e.g., both contain a social security number), the recruiter could request another zero-knowledge proof that the national identifier

[21] Either logically, i.e., at software-level, e.g., through ransomware; or physically, i.e., via physical access to the holder's device.

of the identity credential equals the national identifier of the diploma credential, without disclosing the national identifier.

Tampering. The risk of unauthorized alteration of data (e.g., an applicant altering a diploma credential to better match the desired vacancy) [24]. **Tampering is inherently mitigated by using Verifiable Credentials** as a collection of cryptographically verifiable claims of which the associated digital signatures prevent unauthorized alteration. Even when a VC is tampered with upstream, local verification upon receipt will unveil any kind of tampering.

Repudiation. The risk that an entity can deny performing an action. For example, Alice performs a Sybil attack in which a large number of pseudonyms are created to flood the recruiter with requests or present (minimized) diploma credentials that do not match the vacancy's requirements. Even when an audit log is provided [31][22], a holder could i) intentionally disclose its secret key to deny malicious actions, or ii) delete its pseudonyms so that requests can no longer be verified or bound to the holder. On the one hand, **disclosure of the secret key should be prevented by the keypair managing hardware or software**[23]. On the other hand, **deleting malicious pseudonyms remains a risk to the WebID solution** but is not an issue for the DID:Key solution, as the ephemeral nature of the `did:key` method makes it impossible to remove any DID pseudonym [24].

Information Disclosure. The risk that more information is disclosed than intended. The proposed solutions mitigate threats to information disclosure using: i) a pseudonym which prevents exposing an applicant's true identity; and ii) selective disclosure which allows applicants to present a minimized version of their data to requesting parties. However, the recruiter may discover Alice's identity by tracking her pseudonym. This is a threat specific to the WebID solution as resolving an identifier over a network discloses domain information. Furthermore, responses to probing requests for a particular WebID can disclose information (e.g., status code) from which more information can be inferred (e.g., whether a particular WebID still exists). In contrast, the DID:Key solution's identifier contains no identifiable information and can be resolved locally. For the WebID solution, this can be partially mitigated using herd privacy: a Pseudo IDP can choose to mint many WebID pseudonyms under the same HTTPS URI (e.g., by minting a new WebID using a hash identifier, similar to the VC's revocation list[24]: https://pseudo.com/webids/#1, https://pseudo.com/webids/#2, etc.).

Denial of Service. The risk of attempting to disrupt an actor's functioning. Related to the Sybil attack mentioned in the Repudiation discussion, attacks may involve sending intentionally large, invalid, or inefficiently structured credential payloads to exceed the memory/computation resources of a verifier. **Mit-**

[22] In which special attention should be given to make sure these audit logs cannot create binding data between pseudonyms and true identities.
[23] For example, https://www.ledger.com/.
[24] https://w3c-ccg.github.io/vc-status-rl-2020/.

igations for such a scenario typically require countermeasures at network and application levels (e.g., firewall rules, preemptive validity checks, rate limiting, throttling, appropriate error handling, and Challenge-Response mechanisms (e.g., CAPTHCA)).

Elevation of Privilege. The risk of gaining privileged access to compromise the system, which in our case coincides with the Spoofing risk: Alice providing a falsified diploma credential to get the job.

7.3 Privacy Analysis

We apply the LINDDUN methodology [34] to perform the privacy threat analysis, complemented with a detailed linkability analysis. LINDDUN categorizes threats in seven threat types: *Linking* (associating data items or user actions to learn more about an individual or group), *Identifying* (Learning the identity of an individual, through leaks, deduction, or inference), *Non-repudiation* (being able to attribute a claim to an individual), *Detecting* (deducing the involvement of an individual through observation), *Data disclosure* (excessively collecting, storing, processing or sharing personal data), *Unawareness & Unintervenability* (insufficiently informing, involving or empowering individuals in the processing of their personal data), and *Non-compliance* (deviating from security and data management best practices, standards and legislation). In our analysis, we assume that the pseudomizer i) sufficiently informs its users about how their personal is being processed; and ii) is compliant with the GDPR legislation. Therefore, we consider both solutions to yield no threats to type *Unawareness & Unintervenability* and *Non-Compliance*. Below, we discuss the differences between the two solutions that were not yet covered in the STRIDE analysis[25].

Linking. `L.1.1` (*Unique identifier*) is a threat for the WebID solution: the leased pseudonymous WebIDs' domain name can be linked with the Pseudo IDP, increasing its risk of becoming a target for hackers.

Identifying. Similar to `L.1.1`, `I.2.3` (*Distinguishable attributes*) is a threat for the WebID solution: although a new pseudonymous WebID is being used for every job application, these pseudonymous WebIDs can be distinguished based on the URI domain.

Non-repudiation. `Nr.1` (*Attributable data evidence*) is a threat in both solutions once a derived credential (e.g., the diploma VP) arrives at the recruiter. A credential is cryptographically tied to the holder's private key. Assuming this private key has not been compromised, only the legitimate holder can create a derived credential. Therefore, a user cannot deny having created a derived credential.

Detecting. D.1-3 (*Observed communications*, *Application side effect*, and *System responses*) are threats in the WebID solution: this solution needs flow `P1.2` where the recruiter needs to interact with the pseudonymous IDP, Fig. 3).

[25] The more detailed analysis can be found at https://github.com/SolidLabResearch/pseudonymity-demo/blob/main/performance/LINDDUN-analysis.xlsx.

Linkability Analysis. To properly discuss the pseudonymity potential, we performed a linkability analysis of data items (Table 1), allowing us to discuss which data links can be made by which actors. The following acronyms and symbols are used. /: Not applicable. ∅: Empty set. P: Pseudomizer. R: Recruiter. PP: Pod provider. T_ID: True identifier. P_ID: Pseudonymous identifier. D: DNS service. The asset identifiers (Axx) are reused from Sect. 4.

Table 1. Which data items can be linked by an actor (column "Link(A,B)", reflexive and transitive relation) considering following scenarios: i) status quo, i.e. users employ their true identity (column: "No-solution"); ii) pseudonymous WebID solution; iii) DID:Key solution. The following acronyms and symbols are used. /: Not applicable. ∅: Empty set. P: Pseudomizer. R: Recruiter. PP: Pod provider. T_ID: True identifier. P_ID: Pseudonymous identifier. D: DNS service. The asset identifiers (Axx) are reused from the LINDDUN analysis.

Link(A,B)	WebID	DID:Key
Link(T_ID, A01)	PP	PP
Link(T_ID, A02)	PP	PP
Link(T_ID, A03)	/	/
Link(P_ID, A03)	P or R	P or R
Link(P_ID, A05)	P or R	P or R
Link(P_ID, A05a)	P or R	∅
Link(T_ID, P_ID)	P or (R, P_IDP, D)	P

- *Link(T_ID, A01) and Link(T_ID, A02)*: In all cases, PP is able to link T_ID with A01 andA02. Hence, a vast amount of trust is required in PP.
- *Link(T_ID, A03)* In the "no solution" scenario, both PP and R can directly link T_ID with A03. For the other two pseudonymous scenarios, PP and R can only create a link between P_ID and A03.
- *Link(P_ID, A05a)* The WebID solution allows R to discover more information, A05a, i.e. information about the *party hosting the pseudonym document* (A05), for example, through DNS probing. This is not the case for the DID:Key solution, as R can algorithmically generate A05.
- *Link(T_ID, P_ID)* P allows users to create and use P_ID. This service is thus able to associate one's T_ID with a created P_ID. As the WebID solution also involves the IDP leasing the pseudonymous IDs and the DNS service(s) handling requests to the corresponding domain, this introduces additional attack vectors that can lead to link T_ID with a P_ID.

7.4 Discussion

The requirement validation showed that both solutions cover most requirements (although the WebID solution remains observable, and the DID:Key solution

does not allow deleting or managing deleting pseudonyms). The STRIDE analysis showed a couple of threats for both the WebID and DID:Key solution, that can be (partially) mitigated by the suggested countermeasures. The LINDDUN privacy analysis results, however, indicate that the WebID solution has an increased risk of linking, identifying, non-repudiation and detecting, due to its reliance on an external IDP to host the pseudonym.

8 Conclusion

Users can protect their privacy using a pseudonym. In this paper, we show how we can combine Solid and DID standards to provide pseudonymity, and which (partial and open) threats were identified.

As demonstrated with our use case, both WebIDs and DIDs can be used as pseudonymous identifiers. However, WebIDs cannot be created autonomously, as they inevitably depend on other actors (i.e., IDPs and DNS Registrars), and require the pseudonym document to be resolved from the Web. Using the `did:key` method, we can build completely self-standing pseudonymous DID identifiers.

We did not tackle the trust that is needed from the issuers or pod providers in our case. Indeed, a lot of trust is needed in the pod provider or pod providing software, as this is the holder's storage for all its credentials, and the pod has the required functionality to sign credentials (e.g., to create the cross-signed VC during identity binding). Furthermore, our STRIDE analysis was only performed from the viewpoint of the holder. Alternative viewpoint analyses (e.g., the recruiter's) are needed to make our analysis more complete.

The EUDI Wallet regulations are an incentive to improve personal data sharing while maintaining privacy. With this pseudonymity work, we hope to provide a next step to combine personal data storage incentives with Wallet incentives.

In future work, we aim to advance our threat analyses by leveraging specialized threat-analysis tooling (e.g., SPARTA), and further investigate alternative deployments, e.g., by hosting a pod (proxy) on the holder's local device.

Acknowledgements. The described research activities were supported by SolidLab Vlaanderen (Flemish Government, EWI and RRF project VV023/10). imec ICON project SHARCS (Agentschap Innoveren en Ondernemen project nr. HBC.2022.0543), and Interreg project SecuWeb (0100085). The authors thank Vincent Naessens, Ruben Verborgh, and Pieter Colpaert for their valuable feedback.

Declarations. The authors declare no conflict of interest.

References

1. OpenID Connect Protocol. https://auth0.com/docs/authenticate/protocols/openid-connect-protocol
2. Solid Technical Reports (2021). https://solidproject.org/TR/

3. Balseiro, V., Turdean, T., Zucker, J.: Solid WebID Profile (2023). https://solid.github.io/webid-profile/
4. Bernstein, G., Sporny, M.: BBS cryptosuite v2023 (2023). https://www.w3.org/TR/2023/WD-vc-di-bbs-20231218/
5. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures (2004). https://doi.org/10.1007/978-3-540-28628-8_3
6. Braun, C.H.J., Horne, R., Käfer, T., Mauw, S.: SSI, from Specifications to Protocol? Formally verify security! In: ACM Web Conference (2024). https://doi.org/10.1145/3589334.3645426
7. Braun, C.H.J., Käfer, T.: Attribute-based access control on solid pods using privacy-friendly credentials. In: Poster and Demo Track of the 18th International Conference on Semantic Systems (SEMANTiCS) (2022). https://ceur-ws.org/Vol-3235/paper1.pdf
8. Campbell, B., Bradley, J., Sakimura, N., Lodderstedt, T.: OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens. RFC 8705 (2020). https://doi.org/10.17487/RFC8705
9. Capadisli, S.: Linked Research on the Decentralised Web (2019). https://csarven.ca/linked-research-decentralised-web
10. Capadisli, S., Berners-Lee, T., Verborgh, R., Kjernsmo, K.: Solid Protocol (2022). https://solidproject.org/TR/protocol
11. Chaum, D., Larangeira, M., Yaksetig, M., Carter, W.: W-OTS+ up my sleeve! A hidden secure fallback for cryptocurrency wallets (2021). https://doi.org/10.1007/978-3-030-78372-3_8
12. Coburn, A., Pavlik, E., Zagidulin, D.: Solid-OIDC (2022). https://solidproject.org/TR/oidc
13. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requir. Eng. (2011). https://doi.org/10.1007/s00766-010-0115-7
14. Deng, X., Tian, C., Chen, F., Xian, H.: Designated-verifier anonymous credential for identity management in decentralized systems. Mob. Inf. Syst. **2021** (2021). https://doi.org/10.1155/2021/2807395
15. European Commission: EU Digital Identity Wallet – Architecture and Reference Framework (2025). https://eu-digital-identity-wallet.github.io/eudi-doc-architecture-and-reference-framework/2.0.0/
16. European Network and Information Security Agency: Pseudonymisation techniques and best practices: recommendations on shaping technology according to data protection and privacy provisions (2019)
17. Fallatah, K.U., Barhamgi, M., Perera, C.: Personal data stores (PDS): a review. Sensors (3) (2023). https://doi.org/10.3390/s23031477
18. Florea, M., Esteves, B.: Is automated consent in solid GDPR-compliant? An approach for obtaining valid consent with the solid protocol. Information (12) (2023). https://doi.org/10.3390/info14120631
19. Gabel, A., Schiering, I.: Privacy Patterns for Pseudonymity (2019). https://doi.org/10.1007/978-3-030-16744-8_11
20. Hackett, M., Hawkey, K.: Security, privacy and usability requirements for federated identity. In: Workshop on Web (2012)
21. Heitmann, B., Kim, J.G., Passant, A., Hayes, C., Kim, H.G.: An architecture for privacy-enabled user profile portability on the web of data. In: 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems (2010). https://doi.org/10.1145/1869446.1869449

22. Hofmeier, M., Pöhn, D., Hommel, W.: DistIN: analysis and validation of a concept and protocol for distributed identity information networks. In: 19th International Conference on Availability, Reliability and Security (2024). https://doi.org/10.1145/3664476.3669930
23. Johnson, D., Menezes, A., Vanstone, S.: The elliptic curve digital signature algorithm (ECDSA). Int. J. Inf. Secur. **1**(1), 36–63 (2001). https://doi.org/10.1007/s102070100002
24. Kersic, V., Vidovic, U., Vrecko, A., Domajnko, M., Turkanovic, M.: Orchestrating digital wallets for on-and off-chain decentralized identity management. IEEE Access (2023)
25. Kim, T., Seo, D., Kim, S.H., Lee, I.Y.: A comprehensive approach to user delegation and anonymity within decentralized identifiers for IoT. Sensors **24**(7) (2024). https://doi.org/10.3390/s24072215
26. Mirzamohammadi, O., Jannes, K., Sion, L., Van Landuyt, D., Abidin, A., Singelée, D.: Security and privacy threat analysis for solid. In: IEEE Secure Development Conference (SecDev) (2023). https://doi.org/10.1109/SecDev56634.2023.00033
27. More, S., Heher, J., Fasllija, E., Mathie, M.: Service provider accreditation: enabling and enforcing privacy-by-design in credential-based authentication systems. In: 19th International Conference on Availability, Reliability and Security. ARES 2024 (2024). https://doi.org/10.1145/3664476.3669934
28. Mourby, M., Mackey, E.: Identity, profiles and pseudonyms in the digital environment, chap. 14 (2024). https://doi.org/10.2307/jj.12124947.16
29. Niu, Y., Wei, L., Zhang, C., Liu, J., Fang, Y.: Towards anonymous yet accountable authentication for public Wi-Fi hotspot access with permissionless blockchains. IEEE Trans. Veh. Technol. **72**(3) (2023). https://doi.org/10.1109/tvt.2022.3218528
30. Pfitzmann, A., Köhntopp, M.: Anonymity, unobservability, and pseudonymity - a proposal for terminology. In: Designing Privacy Enhancing Technologies, Intl. Workshop on Design Issues in Anonymity and Unobservability (2000). https://doi.org/10.1007/3-540-44702-4_1
31. Pillitteri, V.Y.: Assessing security and privacy controls in information systems and organizations (2022)
32. Schardong, F., Custódio, R.: Self-sovereign identity: a systematic review, mapping and taxonomy. Sensors (15) (2022). https://doi.org/10.3390/s22155641
33. Shostack, A.: Threat Modeling: Designing for Security (2014)
34. Sion, L., Wuyts, K., Yskout, K., Van Landuyt, D., Joosen, W.: Interaction-based privacy threat elicitation. In: IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) (2018). https://doi.org/10.1109/EuroSPW.2018.00017
35. Sporny, M., Guy, A., Sabadello, M., Reed, D.: Decentralized identifiers (DIDs) V1.0 (2022). https://www.w3.org/TR/did-core/
36. Sporny, M., Jr, T.T., Jones, M., Cohen, G., Herman, I.: Verifiable credentials data model V2.0 (2025). https://www.w3.org/TR/2025/CRD-vc-data-model-2.0-20250225/
37. Sucasas, V., Aly, A., Mantas, G., Rodriguez, J., Aaraj, N.: Secure multi-party computation-based privacy-preserving authentication for smart cities. IEEE Trans. Cloud Comput. **11**(4) (2023). https://doi.org/10.1109/tcc.2023.3294621

38. Verborgh, R.: End-user identity in Solid: the interoperability problem space (2023). https://solidlab.be/wp-content/uploads/2023/04/End-user-identity-in-Solid-the-interoperability-problem-space.pdf
39. Wild, S., Wiedemann, F., Heil, S., Tschudnowsky, A., Gaedke, M.: ProProtect3: an approach for protecting user profile data from disclosure, tampering, and improper use in the context of WebID. In: Hameurlain, A., Küng, J., Wagner, R., Bianchini, D., De Antonellis, V., De Virgilio, R. (eds.) Transactions on Large-Scale Data- and Knowledge-Centered Systems XIX. LNCS, vol. 8990, pp. 87–127. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46562-2_4