One to Bind Them: Binding Verifiable Credentials to User Attributes

Alexander Mühle*, Katja Assaf* and Christoph Meinel
Hasso Plattner Institute, University of Potsdam, Germany

Keywords: Binding Credentials, Linking Credentials, Privacy Enhancing Technology, Accumulators, BBS+, SSI.

Abstract: The Self-Sovereign Identity ecosystem is defined by its flexibility and heterogeneity. While this can be an advantage for users, as they can freely choose their identifiers and attribute providers, it also bears risks. When credentials are being issued, issuers often rely on other previously issued attributes to base their issuance decision on, either personal identifiable information or attestations of requirements. In this paper, we propose two approaches for binding such user attributes in a privacy-preserving way to credentials to prevent fraudulent usage by unauthorised users and enable further auditability of credential requirements and ownership. We propose a selective disclosure-based approach relying on BBS+ signatures. However, as the usage of BBS+ signatures is not yet widespread, we also propose an approach that does not rely on selective disclosure and instead utilises cryptographic accumulators to bind user attributes to the issued credentials.

1 INTRODUCTION

Not all credentials, analogue or digital, contain all the information necessary for appropriate verification. Looking at access control cards, such as library cards, we find that the revealed information is often insufficient for determining whether the person showing the card should get access. That is because the card only reveals the owner's name. As a relying party, here a librarian, checking a photo ID card as well is a necessary step to prevent fraud. The name on the card binds the credential to another credential with which the owner proves their identity, such as a driver's license or ID card. Binding the use of one credential to another is also useful for mapping requirements of the form "*Credential A is valid only if credential B is valid.*" This kind of requirement arises in contexts like access control:

- Access to this lab is only granted (Credential A) if the owner has had the necessary safety training within the last year (Credential B).

or education:

- The holder owns a valid master's degree (Credential A) only if the holder owns a valid bachelor's degree as well (Credential B).

Such an extension to a credential can be especially beneficial if credentials A and B have different issuers and the issuer of credential A fears that credential B expires or is revoked during the validity period of credential A.

Even if the same institution issues credentials A and B, a split into separate credentials can be beneficial to make credentials purpose-specific, following the principle of data minimisation. Additionally, a credential without any personally identifiable information (PII) is easier to issue and has fewer restrictions for data handling from the GDPR. Depending on the contained data, credentials, in general, have different security requirements, such as a credential with PII requires special protection in terms of privacy, while a credential allowing access to restricted areas, physical or digital, requires special protection regarding integrity, and short-lived credentials require less protection than long-lived credentials.

However, stripping the asserted attributes in a credential to a bare minimum raises the need to bind the credential to additional preconditions, which can be part of any other credential. The idea of *binding a credential* usually refers to the binding to a natural person or hardware. We extend the definition of binding a credential to encompass the binding to an attribute as follows:

Definition 1 (Credential Binding)**.** *A credential X is bound to an attribute a if a verifier accepts $Show(X,Y)$ for any credential Y with $a \in Y$.*
A credential X is securely bound to an attribute a if, for any adversary, the probability of providing a credential Y', with $a \notin Y'$, such that a verifier accepts

*These authors contributed equally to this paper

MÃijhle, A., Assaf, K. and Meinel, C.
One to Bind Them: Binding Verifiable Credentials to User Attributes.
DOI: 10.5220/0012057900003555
In *Proceedings of the 20th International Conference on Security and Cryptography (SECRYPT 2023)*, pages 345-352
ISBN: 978-989-758-666-8; ISSN: 2184-7711

$Show(X, Y')$ is negligible.

In most use cases, the attributes bound to a credential are claims asserted by other credentials. However, should a use case require it, it is also possible to bind a specific credential to another credential using a credential ID or hash as an attribute.

Contribution. We extend Verifiable Credentials (VCs) so that they can be bound to user attributes in a privacy-preserving way. The extension allows issuers to define verification requirements and enables relying parties to verify them appropriately. We propose two approaches that allow for privacy-preserving attribute binding: a cryptographic accumulator-based approach and an approach based on selective disclosure enabled through BBS+ signatures. Subsequently, we discuss the security, usability and performance of the approaches.

2 RELATED WORK

Usually, the term 'credential binding' refers to the binding of a credential to a person. Babel and Sedlmeir (Munilla Garrido et al., 2022; Babel and Sedlmeir, 2023) use the term 'holder binding' to emphasise the binding to the credential's holder. However, binding can also mean binding a credential to some hardware (Grassi et al., 2016) for example, to enable two-factor authentication or to enable trusted computing (Camenisch, 2006), via Direct Anonymous attestation (Brickell et al., 2004).

There is no clearly defined terminology in the literature to describe our use case. Some authors use the term 'credential linking' or 'linked credentials' to refer to the ability to verify that credentials from different issuers were issued to the same person (Chase et al., 2022). In the SSI ecosystem, this approach is being used by Hyperledger AnonCreds[1]. Other authors refer to the term as a privacy risk due to the leakage of more information than initially intended (Verheul, 2001). The second interpretation is closely related to the term 'traceability', referring to the ability of an adversary to trace someone via their credential usage information even though the credential itself may not contain any personally identifiable information.

The term 'credential chain' is coined by Hardman and Harchandani (Hardman and Harchandani, 2022) as data in a verifiable credential (VC) traceable to its origin while retaining its verifiable quality. Their extension of VCs allows for adding provenance data when copying data fields from another credential. The most prominent use case for a credential chain is the delegation of authority, similar to X.509 certificates.

The variety of terms ('binding', 'linking', 'chains', 'combining'), often with several interpretations for each term, makes mapping the academic landscape hard. Although research in the field of Verifiable Credentials and their properties is vast, it misses considerations on posing additional requirements checked during verification. Yildiz et al. (Yildiz et al., 2022) provide an overview paper on the interoperability of Self-Sovereign Identities with a useful overview of the different implementation capabilities, such as *Selective Disclosure* and *Credential Binding*. However, *Credential Binding* only refers to the binding of a credential to an identifier controlled by a user, such as used in W3C Verifiable Credentials (VC) (Sporny et al., 2022a) with Decentralised Identifiers (DID) (Sporny et al., 2022b).

Closest to our use case is the work of Babel and Sedlmeir (Babel and Sedlmeir, 2023). They describe the necessity of combining a vaccination credential with a strongly bound, government-issued ID card, called *credential linking*. By hashing the corresponding attributes in both attestations, they allow for a privacy-preserving binding of two credentials. However, they only describe their approach briefly since it is straightforward. A primary difference to our idea is that the issuer defines the requirements which shall be checked during verification in our scheme. Babel and Sedlmeir assume that the verifier already knows what they have to check.

3 SYSTEM OVERVIEW

Self-Sovereign Identity (SSI) is a design principle for identity management systems that has gained popularity in recent years. The W3C standard for verifiable credentials (Sporny et al., 2022a) has matured to *the* standard for credentials within the SSI community. We, therefore, use the verifiable credential standard and the associated system model as our basis.

Self-Sovereign Identity. Full control by the user over their digital identity is the stated goal of Self-Sovereign Identity (Mühle et al., 2018). This goal is achieved by issuers attesting to identity attributes using credentials that are handed over to the user/subject. These credentials are then controlled and managed by the user. The user can, without the direct involvement of the issuer, present these credentials to a relying party which verifies the signatures and current status of the credential. A high-level

[1] hyperledger.github.io/anoncreds-spec

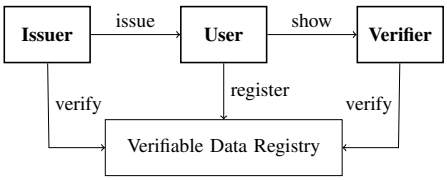

Figure 1: SSI Overview.

overview is presented in Figure 1.

Verifiable Credentials. The W3C Verifiable Credential (VC) comprises credential metadata, one or more claims and a proof. The metadata holds information such as the issuer and expiration date but can be extended to include further information, such as a revocation mechanism. A claim in a VC is also called *credentialSubject* and typically includes an identifier of the claim subject as well as any data relevant to the claim. Finally, the VC is cryptographically signed in the proof portion of the credential. Here, different algorithms can be used.

Verifiable Credential Signatures. The standard of VC has been designed flexibly. It allows for different data formats, such as XML, JSON and JSON-LD, and various signature schemes securing these credentials, such as RSA, ECDSA, or BBS+. BBS+ signatures allow for advanced cryptographic features such as selective disclosure, consequently enabling more privacy-preserving protocols. The practicality of BBS+ signatures has already shown (Looker and Steele, 2023), laying the groundwork for further research (Yamamoto et al., 2022). Based upon the draft and the implementation[2], available at that point, Yamamoto et al. further extended the schema to allow for the selective disclosure of verifiable credentials of different issuers in one verifiable presentation (Yamamoto et al., 2022).

Decentralised Identifiers. In the SSI ecosystem, the standard of Decentralised Identifiers (DID) (Sporny et al., 2022b) has emerged as the identifier mechanism for the issuer and claim subject in VCs. It can be described as an interoperability layer between different identifier approaches. A DID is an identifier that can be resolved to a DID document which contains information on the DID subject, namely cryptographic material used for authentication. The user can choose and generate different identifiers for themselves as they see fit. The user typically manages key material in a specific DID wallet. This user-centric design also bears some risks. Different credentials can be issued to different identifiers. Hence, there needs to be a mechanism to bind multiple credentials with different identifiers to the same natural person to prevent fraudulent usage.

4 BINDING CREDENTIALS

While other systems, such as AnonCreds of the Hyperledger project, enable the linking of credentials in a privacy-preserving way, we aim to expand on this functionality to not only link credentials but bind them with a natural person's attributes. For this purpose, we propose not to use a link secret or persistent identifier that could potentially be compromised or sold on by a malicious user but to utilise various binding attributes such as photo, name, birthdate, and birthplace similar to the analogue use of credentials. As the binding attributes will be included in the issued credential, we extend the VC data model and add an attribute *"requirement"* to the credential subject as shown in Figure 2.

In the following, we describe the necessary adaptions for our extensions in the issuance and presentation process.

4.1 Issuance

During the issuance process, the user will present the issuer with a collection of VCs that attest to the attributes selected for binding. These VCs need to satisfy the level of assurance that the issuer requires. After successful verification, the issuer can issue a new VC with the binding attributes in the "requirements" field. Including the attributes in plain text would already satisfy the basic functionality of credential binding. However, including personally identifiable information goes against the principle of data minimisation and poses a considerable privacy risk for the user. In order to prevent this, the binding attributes should not be revealed unless necessary, and the user consents to the process. In order to achieve this, we propose two different approaches. Verifiable Credentials, in most cases, still rely on common signature schemes such as EdDSA, ECDSA or RSA[3]. As such, they do not offer selective disclosure. In this case, we propose to use a function *acc* to accumulate a list of required attributes $[a_1, ..., a_n]$ in a privacy-preserving way. Figure 2 shows the structure of such a VC. Using the accumulator, verifying if a specific attribute is required without revealing other

[2]https://github.com/mattrglobal/jsonld-signatures-bbs

[3]w3c-ccg.github.io/vc-extension-registry/

Figure 2: Exemplary VC with Requirements.

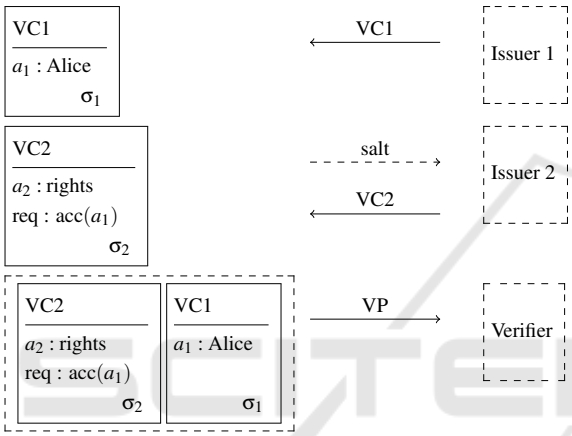

Figure 3: Flow with Accumulated Requirements.

attributes in the requirements field is possible. The data flow is shown in Figure 3. A further discussion on the algorithms suitable for *acc* is presented in Section 4.1.1. However, there have also been efforts to bring selective disclosure to VCs through BBS+ signatures, as presented in Section 4.1.2. When using BBS+ signatures, the attributes $[a_1,...,a_n]$ can be directly written in the *"requirement"* field since they can be hidden during presentation through the means of selective disclosure.

4.1.1 Accumulators

The concept of a one-way membership hash function has been introduced by (Benaloh and De Mare, 1994). Since then, other designs have emerged using hash functions, RSA and bilinear mappings (Lauradoux et al., 2021). With the different designs and underlying cryptographic primitives, different properties can be achieved for the accumulators. The addition and removal of items are possible in *dynamic* accumulators as introduced by (Camenisch and Lysyanskaya, 2002). *Positive* accumulators can prove set membership. In contrast, *negative* accumulators can only create a proof of non-inclusion. For *universal* accumulators, both are possible.

As an additional property, accumulators can be zero knowledge (Ghosh et al., 2016), meaning the element for which the membership is being proven remains hidden. In the following, we briefly present different implementations of symmetric and asymmetric accumulators and discuss their suitability for our binding approach.

Symmetric Accumulators. Bloom filters were introduced in 1970 (Bloom, 1970) and have been used in privacy-sensitive environments (Gomez-Barrero et al., 2016). They are a probabilistic data structure that utilises hash functions to enable a space-efficient set membership check. Each element added to the filter will be hashed with k different hash functions. The hash function output is mapped on the filter itself, which is a bit array of size m. An increased number of elements in the set increases the false positive rate. Due to the one-way property of the used hash functions, Bloom filters are not dynamic, and removing items is not feasible. Cuckoo filters have been a significant iteration of the Bloom filter concept and function similarly while having lower false-positive rates and space overhead (Fan et al., 2014).

However, our use-case of credential binding typically will not involve large numbers of attributes. The false positive rate is, therefore, not a major drawback. If the bloom/cuckoo filter is correctly constructed concerning the number of hash functions used and available bits in the filter array, it can be used for our binding approach. As stated before, symmetric accumulators have no additional witness that needs to be stored by the user. While this is certainly an advantage from a usability perspective, the usage of a witness also has some advantages. Relying on a witness requires the user's involvement during the set membership verification. Without the user-created witness, a relying party can not check which attributes have been bound to a particular credential. The user's cooperation can be seen as a form of consent that aligns with the principles of Self-Sovereign Identity. In order to still have this form of consent when using symmetric accumulators, we propose to salt the attributes before including them in the filter. Salting is a trade-off from a usability perspective, as this salt needs to be stored and presented by the user during verification. However, it prevents the unauthorised membership check of undisclosed attributes by a relying party.

Nyberg has proposed a non-trapdoor, and therefore, symmetric accumulator that is not probabilis-

tic (Nyberg, 1996). However, due to the reliance on the Random Oracle Model and severe space-inefficiency, it has not been considered for practical use (Fazio and Nicolosi, 2002), and we will not consider them for usage in our concept.

Asymmetric Accumulators. Accumulators based on asymmetric cryptography typically utilise RSA or bilinear maps. The first such accumulator was proposed by (Benaloh and De Mare, 1994). Baric and Pfitzmann have developed an RSA-based accumulator that is collision-free in contrast to Benaloh and de Mare's proposal (Barić and Pfitzmann, 2001). Tartary et al. have built upon a proposal by Nguyen based on bilinear maps (Tartary et al., 2008). The advantage of the above asymmetric accumulators is a constant witness size. However, the creation time of the required witnesses increases with the number of elements in the set. Once again, however, our use case will typically not run into this limitation as the set will not grow above a handful of elements. Another advantage of RSA- and bilinear map-based accumulators is their dynamic property. Adding and removing items is relatively easy. However, this is not a property we require for our binding approach. The accumulator is only created once during the issuance of the credential and can only be updated afterwards if the VC is reissued. The persistence is ensured by securing the value in the VC using a cryptographic signature. An update should only be performed if the issuer is aware of the attributes and verifying the validity and association with the credential subject to their satisfaction. Due to the alignment of users' involvement indicating their consent, asymmetric cryptographic accumulators are a good fit for our binding approach.

4.1.2 Selective Disclosure

The idea of selective disclosure was introduced to the W3C VC standard with the addition of advanced concepts, which was released in February 2019[4], although, the term *Zero-Knowledge Proofs* is used. CL-signatures (Camenisch and Lysyanskaya, 2001) serve as a basis. This type of signature, not to be confused with the newer, group-based signatures (Camenisch and Lysyanskaya, 2004), is the earliest implementation in a line of research allowing for privacy-preserving signatures related to the capability of selective disclosure and blind signatures. In this field, BBS (Boneh et al., 2004) and BBS+ signatures (Au et al., 2006) emerged as well.

[4]https://www.w3.org/TR/2019/WD-verifiable-claims-data-model-20190208/

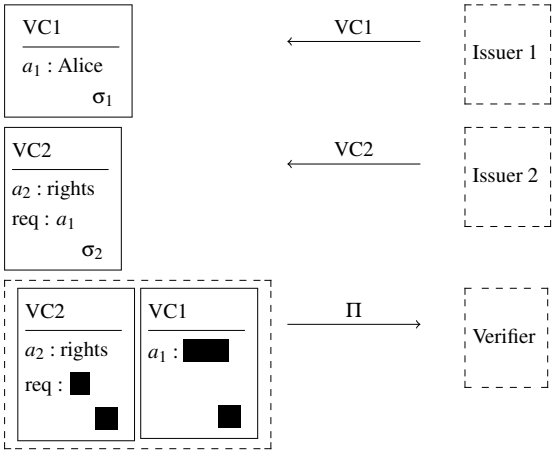

Figure 4: Flow with Selective Disclosure.

Although newer signature schemes with advantages in size or complexity exist, such as (Pointcheval and Sanders, 2016) and (Fuchsbauer et al., 2019), no noticeable efforts have been undertaken yet to use these signature types for VCs. This lack of implementations might also be due to the need for standardised, production-ready libraries containing one or the other signature scheme.

Due to the current efforts to use BBS+ signatures in VCs, we choose them as the basis for our selective disclosure approach. In this case, using the accumulator to hide the original requirements is unnecessary (see Figure 5). The user can derive a partially hidden presentation and zero-knowledge proof for the requirements with the selective disclosure feature as shown in Figure 4. The main effort is hidden in the Zero-Knowledge Proof Π created during the presentation phase (Section 4.2).

We take the algorithms defined by Yamamoto et al. (Yamamoto et al., 2022) as the basis of our scheme:

$\underline{\text{Setup}(1^\lambda, L) \rightarrow (\text{par, isk, ipk, usk})}$

During the asynchronous setup phase, some trusted third party, such as a governance body or the issuer itself, generates the public parameters *par* from a security parameter 1^λ and an upper bound L for the number of attributes. Given *par*, the issuer generates a key pair (isk, ipk) and the user a secret key usk whenever necessary.

$\underline{\langle \mathsf{obtain}(usk, ipk, G), issue(isk, G) \rangle \rightarrow \langle \sigma, - \rangle}$

This is an interactive protocol where the user, upon input of their private key usk, the issuer's public key ipk and attributes in the structured form of a graph G, runs obtain. The issuer runs issue with its secret key isk and the graph G and generates a signature σ or presignature, which the user can turn into a signature.

```
{
  "context": [...],
  "id": "https://issuer.lib.city/credentials/78912634",
  "type": ["VerifiableCredential", "LibraryCard"],
  ...,
  "credentialSubject": {
    "id": "did:example:b345acb345d",
    "type": ["LibraryCard", "Person"],
    "givenName": "Alice",
    "familyName": "Liddell",
    "requirement": [a1, ..., an]
  }
  "proof": {
    "type": "BbsBlsSignature2020",
    "created": "2023-02-23",
    "verificationMethod": "did:example:489398593#test",
    "proofPurpose": "assertionMethod",
    "proofValue": "G7uMuJzasdAEqj4j+HPTvoe6KAW...",
    "requiredRevealStatements": [ ]
  }
}
```


Figure 5: Exemplary VC with Requirements.

The attributes in G together with the signature σ are called a credential C.

$$\langle \text{show}(\text{usk}, (C_i, \phi_i)_i) \text{ , verify}((C'_i)_i, \Pi) \rangle \rightarrow \langle \Pi, 0/1 \rangle$$

For revealing only a subset of credentials C_i, the user generates a proof Π from the credentials $(C_i)_{i \in I}$ and the corresponding reveal functions $(\phi_i)_{i \in I}$. The revealed part of the credential C'_i, together with the proof Π, serves as input for the verifier to ascertain the validity of the credentials.

We omitted several details of the original scheme. Especially the distinction between bound and unbound credentials is not necessary for our use case since we consider only bound credentials.

4.2 Presentation

The relying party can specify what credentials they want to see in an established format[5]. As the required attributes are hidden through an accumulator or selective disclosure mechanism, the relying party must first know what type of attributes are included in the requirement field. For this purpose, we propose that the issuer publishes the credential schema to be accessible for relying parties akin to Credential Definitions as utilised by Hyperledger Indy[6].

As with the current presentation workflow, the user first needs to initiate the Verifiable Presentation (VP) exchange, in which the relying party creates a challenge and presentation request. The user then signs the selected VCs and challenges in order to prove control of the credentials. With our scheme, in addition to proving control of the credential, the user also presents valid credentials for all required attributes and shows that the presented attributes match the required attributes. The presentation is done differently depending on the utilised approach.

When utilising symmetric accumulators, the inclusion of the presented attributes in the requirement field of the credential can be checked directly without the use of a witness. However, as discussed before, the attributes have been salted. In addition to the credentials, including the required attributes, the associated salt must be provided for the relying party to verify the set membership. Similarly, the witness required for set membership verification when using asymmetric accumulators must be generated and provided to the relying party.

When using BBS+ signatures, no additional witness or salt needs to be presented. Instead, the credential is partly revealed, and the fitting proof Π is created (see Figure 4). However, depending on the number of requirements, the creation or verification of the proof can become quite computation-heavy. A signature proof of knowledge must be created for each VC added to the presentation. Additionally, for each required attribute a, listed in the requirement field of any presented credential VC2 , a zero-knowledge proof must be generated that a is contained in another presented credential VC1. In the simplest case with two VCs, as shown in Figure 4, the proof could look like this:

$$\text{ZKP}\{(a_1, \sigma_1, \sigma_2) : \\ \text{verify}(VC1, \sigma_1) \wedge \text{verify}(VC2, \sigma_2) \wedge a_1 \in VC2\}$$

with verify being the verification function of the BBS+ signature.

5 EVALUATION

5.1 Security

A key feature of the Self-Sovereign Identity ecosystem, as proposed by the W3C, is the freedom of the user to choose identifiers and manage them independently. Consequently, combining different credentials into a single presentation is not as straightforward. In the multi-issuer ecosystem of SSI, different issuers will issue credentials to different identifiers for the same natural person. An attacker should not be able to combine their credentials with the credentials of another user and use them fraudulently in a single presentation together. In the following, we will

[5] w3c-ccg.github.io/vp-request-spec/

[6] hyperledger.github.io/anoncreds-spec/

briefly look at security considerations regarding the three main actors in the system and how our scheme relates to them.

Issuer. It is always possible that an issuer is compromised, and should be expected when designing such a system. While we cannot prevent the issuance of an unauthorised credential by a compromised issuer, with our scheme, we have a non-repudiation mechanism for the issuance process. The issuer will not be able to repudiate that he indeed issued a credential to a specific user, as the required attributes for issuance are bound to the issued credential and will be verified when presented to a relying party. While a compromised issuer could in theory include attributes of an authorised user, the unauthorised user presenting the credential will not be able to produce a proof of control for the included attributes, therefore failing verification at the relying party.

Relying Party. A compromised relying party is just as much of a concern as it is with issuers. Manipulating the relying party enables circumvention of proper credential verification, and an attacker could therefore gain unauthorised access to the resources of the relying party. However, such an attack is possible regardless of the credential scheme employed. Suppose a user has gained access to resources he was not authorised for. In that case, the issuer can prove that he did not issue the credential to an unauthorised user but that the relying party was at fault for not correctly verifying the required attributes.

Credential Holder. The main concern, apart from the compromise of the issuer or relying party, is users' malicious behaviour. For example, Alice might conspire with another holder Bob and pass on their library card for Bob to use as his own. The same problem can also arise when Alice has been compromised, and her private key material is leaked for Bob to use. Our scheme can be used to prevent this fraudulent usage by selecting the appropriate attributes to include as requirements for credential usage. During issuance, attributes uniquely identifying the authorised user (Alice) are included, such as a photo, name, birthdate or similar. If the library card gets passed on, Bob must produce the matching attributes, which requires additional effort. While the user in SSI generally can choose their own identifiers, issuers can stipulate a specific type of identifier to be used. Credentials with a high level of assurance, such as national ID cards, for example, might prescribe the usage of hard-to-pass-on identifiers tied to physical ID cards or similar mechanisms. Passing on access to these would have significant implications and would dissuade such an attack.

5.2 Efficiency

Kumar et al. (Kumar et al., 2014) have surveyed the landscape of cryptographic accumulators and have conducted a systematic performance analysis using the example of CRLs. They noticed that while symmetric accumulators such as Bloom filters are often used, ECC-based asymmetric accumulators offer significantly reduced memory usage and improved verification. RSA-based accumulators were much slower during verification than both ECC-based and Bloom filters.

6 CONCLUSION

In this work, we tackle the problem of binding Verifiable Credentials to natural persons' attributes in a privacy-preserving fashion to model requirements checked during verification. We proposed two distinct approaches which rely on different cryptographic primitives. Using cryptographic accumulators to include attributes required during verification is an approach that can be implemented with current technology in a secure and usable way. We also propose an approach using BBS+ signatures which has been recognised as a promising selective disclosure approach; however, it is still in the experimental stage. Compared to the current ecosystem, we have shown the security benefits of using a binding scheme as proposed in this paper. However, implementation and comparison of real-world results would be the logical next step.

ACKNOWLEDGEMENTS

This work has been funded through the Federal Ministry for Education and Research (BMBF) under grant M534800. We want to thank our partners at the TU Munich and the German Academic Exchange Service (DAAD) for the discussions on the topic and Dennis Dayanikli for the helpful thoughts and comments he provided.

AUTHOR STATEMENT

Katja Assaf provided the main part of the selective disclosure-based approach. Alexander Mühle pro-

vided the main part of the accumulator-based approach. Both authors contributed equally.

REFERENCES

Au, M. H., Susilo, W., and Mu, Y. (2006). Constant-size dynamic k-taa. In *SCN 2006*, pages 111–125. Springer.

Babel, M. and Sedlmeir, J. (2023). Bringing data minimization to digital wallets at scale with general-purpose zero-knowledge proofs. *arXiv preprint arXiv:2301.00823*.

Barić, N. and Pfitzmann, B. (2001). Collision-free accumulators and fail-stop signature schemes without trees. In *EUROCRYPT'97*, pages 480–494. Springer.

Benaloh, J. and De Mare, M. (1994). One-way accumulators: A decentralized alternative to digital signatures. In *EUROCRYPT'93*, pages 274–285. Springer.

Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426.

Boneh, D., Boyen, X., and Shacham, H. (2004). Short group signatures. In *CRYPTO 2004*, volume 3152, pages 41–55. Springer.

Brickell, E., Camenisch, J., and Chen, L. (2004). Direct anonymous attestation. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145.

Camenisch, J. (2006). Protecting (anonymous) credentials with the trusted computing group's trusted platform modules v1. 2. In *SEC 2006*.

Camenisch, J. and Lysyanskaya, A. (2001). An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT 2001*, pages 93–118. Springer.

Camenisch, J. and Lysyanskaya, A. (2002). Dynamic accumulators and application to efficient revocation of anonymous credentials. In *CRYPTO 2002*, volume 2442, pages 61–76. Springer.

Camenisch, J. and Lysyanskaya, A. (2004). Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO 2004*, pages 56–72. Springer.

Chase, M., Orrù, M., Perrin, T., and Zaverucha, G. (2022). Proofs of discrete logarithm equality across groups. *Cryptology ePrint Archive*.

Fan, B., Andersen, D. G., Kaminsky, M., and Mitzenmacher, M. D. (2014). Cuckoo filter: Practically better than bloom. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pages 75–88.

Fazio, N. and Nicolosi, A. (2002). Cryptographic accumulators: Definitions, constructions and applications. *Paper written for course at New York University: www. cs. nyu. edu/nicolosi/papers/accumulators. pdf*.

Fuchsbauer, G., Hanser, C., and Slamanig, D. (2019). Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. *Journal of Cryptology*, 32:498–546.

Ghosh, E., Ohrimenko, O., Papadopoulos, D., Tamassia, R., and Triandopoulos, N. (2016). Zero-knowledge accumulators and set algebra. In *ASIACRYPT 2016*, pages 67–100. Springer.

Gomez-Barrero, M., Rathgeb, C., Galbally, J., Busch, C., and Fierrez, J. (2016). Unlinkable and irreversible biometric template protection based on bloom filters. *Information Sciences*, 370:18–32.

Grassi, P. A., Fenton, J. L., Newton, E. M., et al. (2016). Draft nist special publication 800-63b digital identity guidelines. *National Institute of Standards and Technology (NIST)*, 27.

Hardman, D. and Harchandani, L. (2022). Aries rfc 0104: Chained credentials. Technical report, Tech. Rep., 2019, https://github. com/hyperledger/aries-rfcs/blob/main ....

Kumar, A., Lafourcade, P., and Lauradoux, C. (2014). Performances of cryptographic accumulators. In *39th Annual IEEE Conference on Local Computer Networks*, pages 366–369. IEEE.

Lauradoux, C., Limniotis, K., Hansen, M., Jensen, M., and Eftasthopoulos, P. (2021). Data pseudonymisation: Advanced techniques and use cases. Technical report, European Union Agency for Cybersecurity.

Looker, T. and Steele, O. (2023). Bbs cryptosuite v2023. https://w3c.github.io/vc-di-bbs/.

Mühle, A., Grüner, A., Gayvoronskaya, T., and Meinel, C. (2018). A survey on essential components of a self-sovereign identity. *Computer Science Review*, 30:80–86.

Munilla Garrido, G., Sedlmeir, J., and Babel, M. (2022). Towards verifiable differentially-private polling. In *ARES 2022*, pages 1–11.

Nyberg, K. (1996). Fast accumulated hashing. In *Fast Software Encryption: Third International Workshop Cambridge, UK, February 21–23 1996 Proceedings 3*, pages 83–87. Springer.

Pointcheval, D. and Sanders, O. (2016). Short randomizable signatures. In *CT-RSA 2016*, pages 111–126. Springer.

Sporny, M., Longley, D., and Chadwick, D. (2022a). Verifiable credentials data model. Technical report, World Wide Web Consortium.

Sporny, M., Longley, D., Sabadello, M., Reed, D., Steele, O., and Allen, C. (2022b). Decentralized identifiers. Technical report, World Wide Web Consortium.

Tartary, C., Zhou, S., Lin, D., Wang, H., and Pieprzyk, J. (2008). Analysis of bilinear pairing-based accumulator for identity escrowing. *IET Information Security*, 2(4):99–107.

Verheul, E. R. (2001). Self-blindable credential certificates from the weil pairing. In *ASIACRYPT 2001*, pages 533–551. Springer.

Yamamoto, D., Suga, Y., and Sako, K. (2022). Formalising linked-data based verifiable credentials for selective disclosure. In *EuroS&PW 2022*, pages 52–65. IEEE.

Yildiz, H., Küpper, A., Thatmann, D., Göndör, S., and Herbke, P. (2022). A tutorial on the interoperability of self-sovereign identities. *arXiv preprint arXiv:2208.04692*.