How to Bind Anonymous Credentials to Humans

Julia Hesse, *IBM Research Europe - Zurich;* Nitin Singh, *IBM Research India - Bangalore;*
Alessandro Sorniotti, *IBM Research Europe - Zurich*

https://www.usenix.org/conference/usenixsecurity23/presentation/hesse

This paper is included in the Proceedings of the
32nd USENIX Security Symposium.

August 9–11, 2023 • Anaheim, CA, USA

978-1-939133-37-3

Open access to the Proceedings of the
32nd USENIX Security Symposium
is sponsored by USENIX.

How to Bind Anonymous Credentials to Humans

Julia Hesse*
IBM Research Europe - Zurich
jhs@zurich.ibm.com

Nitin Singh
IBM Research India - Bangalore
nitisin1@in.ibm.com

Alessandro Sorniotti
IBM Research Europe - Zurich
aso@zurich.ibm.com

Abstract

Digital and paper-based authentication are the two predominant mechanisms that have been deployed in the real world to authenticate end-users. When verification of a digital credential is performed in person (e.g. the authentication that was often required to access facilities at the peak of the COVID global pandemic), the two mechanisms are often deployed together: the verifier checks government-issued ID to match the picture on the ID to the individual holding it, and then checks the digital credential to see that the personal details on it match those on the ID and to discover additional attributes of the holder. This pattern is extremely common and very likely to remain in place for the foreseeable future. However, it poses an interesting problem: if the digital credential is privacy-preserving (e.g. based on BBS+ on CL signatures), but the holder is still forced to show an ID card or a passport to verify that the presented credential was indeed issued to the holder, what is the point of deploying privacy-preserving digital credential? In this paper we address this problem by redefining what an ID card should show and force a minimal but mandatory involvement of the card in the digital interaction. Our approach permits verifiers to successfully authenticate holders and to determine if they are the rightful owners of the digital credential. At the same time, optimal privacy guarantees are preserved. We design our scheme, formally define and analyse its security in the Universal Composability (UC) framework, and implement the card component, showing the running time to be below 200*ms* irrespective of the number of certified attributes.

1 Introduction

In the past decade, digital authentication has made considerable progress from academic proposal to real-world viability, owing to several factors. One factor is the cryptographic community, which has provided secure, efficient and privacy-preserving tools for *issuers* to issue *credentials* to *holders*, and let them prove their possession to *verifiers*. These tools, predominantly under the umbrella of anonymous credentials, offer guarantees such as untraceability, anonymity and data minimisation which are ideally suited to provide adequate privacy for holders.

A second factor is the blockchain revolution, which has provided a catalyst for taking those academic tools and bringing them to fruition as fully-fledged Self-Sovereign Identity (SSI) platforms. An example of such a platform is Hyperledger Indy [38] which provides a full-stack SSI solution. Other examples include Veramo [31] and Okapi [56].

Another factor is constituted by the COVID global pandemic, which in large parts of the globe required the introduction of digital passports that citizens had to show, in person, to prove facts about their health status to human verifiers. The latter could perform the verification aided by mobile phones and verifier apps. This has sparked awareness in the public about digital authentication, has fostered the creation of usable wallets for digital credentials and somewhat normalised the fact that citizens carry and present digital credentials to human verifiers. The pandemic has also seen a drastic increase in the level of public-sector investment in digital authentication [5].

In spite of all of these technological as well as behavioral advancements, authentication based on government-issued physical ID is still prevalent and shows no sign of giving way to its digital counterpart. This is because long-term incumbent authentication paradigms tend to be favoured [12], and also because government-issued physical ID is still to this day the fail-safe way to address the following problem:

If a person verifies a digital credential, how can they be sure that it was issued to the person who presents it?

This problem is typically addressed in one of two ways: either by extending the digital credential with biometric information (e.g. a picture of its rightful owner); or by requiring the person presenting the digital credential to show government-issued physical ID to the person verifying the digital cre-

*The author was supported by the Swiss National Science Foundation (SNSF) under the AMBIZIONE grant "Cryptographic Protocols for Human Authentication and the IoT.

dential (as mandated for instance for the verification of the European Digital Green certificate [36]).

Both approaches, however, fall short when it comes to their privacy guarantees. Indeed, the verification performed in person between prover and verifier typically involves a physical (e.g. looking at a face or inspecting a passport) and a digital (e.g. receiving a credential or scanning a QR code) component. While it is fair to assume transcripts of the physical verification are transient (e.g. the verifier will forget faces), digital transcripts can be easily stored, shared, stolen and abused. Consequently, we posit that whenever a solution creates digital traces that reveal "too much", untraceability is lost because whatever is revealed in digital form may be abused to enable the construction of full user profiles, where attributes disclosed over time by the holder can be accumulated by a set of honest-but-curious or malicious verifiers.

In this work we address this problem and propose a scheme where digital authentication enjoys ideal privacy guarantees, and where human verifiers are able to securely establish a match between the digital credential and the individual standing in front of them. We achieve this by relying on a new authentication token embodied by a smartcard. The card is issued to the holder, in person, by a trusted issuing authority. The card must be physically presented to verifiers upon verification of the digital credential to act as a binding between the digital (credential) and physical (individual) world. The card plays a similar role as the one played by government-issued physical ID today, with two important distinctions: i) the card only displays a picture of the holder to allow the verifier to match the holder, together with the necessary security features to determine the card's authenticity: crucially, no other information is present on the card, since any and all attributes of the holder will be disclosed digitally; ii) the card must have computation and transmission capabilities (typical of smartcards) to run a small yet mandatory part of the authentication protocol to prevent mix-and-match attacks.

Our contributions include:

- The formalisation of the concept of *Anonymous Credentials with Visual Holder Authentication*, a system that solves the problem described above.
- The proposal of a card-based Anonymous Credential (cbAC) scheme, a novel cryptographic primitive that minimally and efficiently extends the BBS+ [20] signature suite widely used in the SSI community [39, 49] and being currently standardised at the IETF [46]; our scheme achieves several desirable properties: i) it forces the involvement of the card in the authentication step; ii) it respects the asymmetry of resources between card and holder; and iii) the card always runs the same program, irrespective of who scans it.
- A formal security modeling and analysis of the primitives in the UC framework.
- A proof-of-concept of the viability of our approach, where we show that the slowest entity in our system – the smartcard – can generate its contribution in the order of hundreds of milliseconds on commodity hardware, irrespective of the number of attributes that are certified in the credential.

The rest of this paper is organised as follows: Section 1.1 discusses the related work. In Section 2 we describe the problem this work solves in more detail, outline our solution and detail system and threat model. Section 3 describes joint proofs of knowledge for BBS+ signatures, a core building block of our solution. In Section 4 we describe and define the cryptographic primitive that we identify to be sufficient for anonymous credentials with visual holder authentication. Section 5 presents our construction and its security analysis. A performance evaluation is given in Section 6. Due to space constraints, we defer preliminaries on proof systems and BBS+ signatures, the full formal protocol description and proofs to the full version of this work [43].

1.1 Related Work

Biometric authentication One approach to bind digital credentials to the appearance of the holder is to integrate biometric authentication into the presentation of the credential. The idea of fuzzy cryptography [44] is to derive a high-entropy secret from fuzzy authentication data, such as biometric readings. Such primitives are useful to integrate biometrics of the credential holder into the holder's authentication actions, in order to bind these actions to their appearance. PrivBioMTAuth [40] combines fuzzy extractors [34] and authentication: users can authenticate to remote services using their biometrics sampled by their own mobile phones. A high-entropy secret is generated using the fuzzy extractor, after successful match of a picture of the holder against a pre-generated template, and this secret is later used to produce a zero-knowledge proof that achieves authentication.

Rila et al. [54] propose a system where cardholders authenticate to cards using biometric data (and fingerprints in particular). The authentication occurs between the smartcard and the smartcard reader in adversarial setups, considering for instance replay attacks and active adversaries. All these biometric-based approaches have drawbacks that we aim to avoid in this work. They either encourage digital hubs of biometric information (targets for theft/leaks), or rely on holder devices to perform biometric matching.

Anonymous Credentials Introduced by Chaum [28] and Lysanskaya et al. [48], anonymous credentials (AC) have been refined and augmented with privacy enhancements such as unlinkability across multiple presentation, selective disclosure of attributes (or predicates on them), and support for revocation. The cryptographic primitives that underpin anonymous cre-

dentials include Camenisch-Lysanskaya signatures [22–24], the U-Prove protocol suite [14] and BBS+ signatures [20].

Direct Anonymous Attestation (DAA) An immediate application of the concepts that underpin anonymous credentials is direct anonymous attestation (DAA). DAA allows a platform consisting of a secure element (a Trusted Platform Module (TPM) or Trusted Execution Environment (TEE)) and a host to create anonymous attestations and prove that the attestation was generated using an authorized secure element. DAA was initially proposed in [17] and further refined in [15, 16, 20, 21]. The DAA scheme provides a *join* interface through which an issuer binds a host and a secure element as a platform, and certifies the platform by issuing a credential. Later the platform can create valid signatures on messages using *sign* interface. A full featured DAA scheme (e.g. [21]) also features a *verify* interface allowing parties to verify that signatures are created by certified platform, and a *link* interface to determine if two signatures were generated by the same platform. While our definitions resemble the ones in [21], there are several important differences. Most importantly, a DAA protocol outputs an attestation object which can be stored, transferred, and repeatedly (and locally) verified. The goal of credential verification in this work is, however, not to output such an object, but instead to convince the verifier of a certain "ad hoc" statement (i.e., let them output a bit) with the help of an interactive credential presentation procedure.

Anonymous Credentials on Smart Cards A straightforward approach to bind digital credentials to physical appearance is to delegate the presentation of the credential to a smartcard, which could also embed a tamper-resistant picture of the credential holder. Recent efforts target the real-time applicability of ACs on resource-constrained devices such as smartphones and smart cards. [50, 51] present a smartcard implementation of the U-Prove [52] AC system. Similarly, Idemix AC system [26] on a smartcard is presented in [8, 33, 57]. Batina et al. [4] propose a pairing-based AC system to be implemented on Java cards. A promising line of work [3, 19, 27] for smart card friendly anonymous credentials is *Keyed-Verification Anonymous Credentials* (KVAC) introduced by Chase et. al in [27], where the issuer is also the verifier. Intuitively, the setting in KVAC allows one to replace signatures with simpler message authentication codes.

Multi Device Anonymous Credentials A separate line of work to combine smartcards with attribute-based credentials does not attempt to run the entire protocol on the card: the card is instead used in conjunction with a prover app so that card and prover can jointly authenticate to a verifier. This approach is motivated either by attempts to respect the asymmetry of resources between user and card (e.g. by trying to keep the card's computational load independent of the number of attributes), or to leverage the secure element that is present in smartcards to prevent credential cloning or theft. U-prove's design [14] proposes splitting the certified attributes between card and holder to force the card's involvement: our work efficiently translates this approach to the BBS+ setting in a provably secure manner while permitting multi-show credentials, neither of which is supported by the original work. Lueks et al. [47] propose leverageing a central server to assist a user in anonymously presenting a BBS+ credential [11]. The idea is to share the BBS+ signing key between the user device and the server, and use a threshold version of BBS+ for presentation of a credential. Several other works on thresholdizing AC systems exist; however, they put equal load on each proving device [35], or require a majority of them to be honest [55]. Our system puts only minimal computation load on the card and tolerates corruption of both card and holder. Closest to our work is a system by Hanzlik and Slamanig [42], which leverages smartphones in conjunction with smart cards to let both jointly present shared credentials. Their scheme is shown to be efficient in practice and in particular ensures that the computational overhead of the core device is independent of the number of attributes in the credential. Our system achieves the same independency. However, their solution involves fairly recent cryptographic primitives such as signatures with flexible public keys (SFPK [2]) and signatures on equivalence classes (SPS-EQ [30, 41, 45]), which need to be coupled in a non-trivial manner: the message space of SPS-EQ scheme should match with the key space of the SFPK scheme. Additionally, the choice of the primitives makes it non-trivial to augment the scheme in [42] to support *blind signing* and predicates other than selective disclosure. Our work is based on more established primitives such as BBS+ signature [11] and Schnorr proofs of knowledge; in particular, our credentials *are* BBS+ signatures and thus compatible with existing wallet implementations. BBS+ is backed up by mature implementations [39, 49] and is currently undergoing standardization [46] by the Internet Research Task Force (IRTF). For concrete exposition, we describe our solution for the case of selective disclosure of attributes, though it can be trivially extended to support predicates, which can be efficiently verified using Schnorr proofs. Another notable difference to our work is that [42] demand a weaker notion of anonymity: in their system, cards only communicate with smartphones, and hence they only consider the joint privacy of card and smartphone facing potentially malicious third parties. In our work, we allow more general communication patterns and hence demand the anonymity of cards already stand-alone, meaning that an adversary getting read access to the card should not be able to detect whether he already talked to the same card before. This stronger anonymity guarantee allows the deployment of our scheme in crowded environments such as airports, where users have no control over which other devices are within, say, NFC connection range to their card.

2 Problem statement

Our work focuses on scenarios where an individual is required to authenticate, in person, to another individual. We will refer to the former as the *holder* (of a credential) and to the latter as the *verifier* (of that credential). Further, we focus on scenarios where this authentication uses digital means: the credential is thus not a physical artefact but a digital one. Digital credentials are flexible and convenient; furthermore, in contrast to their physical counterparts, digital credentials lend themselves well to the creation of advanced authentication schemes that preserve the anonymity and unlinkability of the holder and minimise the amount of information a holder has to disclose. For example, it is possible to build a scheme where the owner of a liquor store only learns the value of the boolean associated to the "purchaser is above 18 years of age" predicate when performing the age checks required by law.

Whenever this scenario occurs, the verifier faces a problem, namely that of determining whether the credential they verify was issued to the person presenting it, as opposed to someone else – a colluding malicious entity or the perpetrator of identity theft. The problem is often resolved by requiring the holder to also produce a physical piece of identification (e.g. an ID card): matching the personal details on the digital credentials with those on the ID card. Visually verifying that the picture on the ID card matches the individual provides the missing link in the verification chain.

This, however, all but thwarts any privacy ambition, since it forces the release of a full digital fingerprint (and not just the boolean value from the example above), defeating anonymity, unlinkability and data minimisation.

> **Design principle 1**: Matching digital credentials to humans by requiring traditional physical identification means (e.g. passport, ID card) violates the privacy of the holder.

One way to avoid this unwanted release of personal information would be to embed the picture of the holder as part of the digital credential. In the above example, the merchant would seemingly just learn the necessary boolean together with a digital picture needed to match the individual. This strategy poses two challenges: the first is that if the digital picture is sent to the verifier, anonymity, unlinkability and data minimisation are violated as before. This is true since the digital picture acts as a unique identifier for the user (violating anonymity), permits the linking of different presentations to a specific holder (breaking unlinkability) and creates the basis for tracing and profiling, since verifiers may decide to pool information learned about a specific holder from numerous interactions.

> **Design principle 2**: It is not desirable from a privacy standpoint to send digital visual information about the holder to the verifier.

The shortcomings of this approach may be avoided by not sending the digital version of the picture to any verifier-controlled device. This leaves keeping it on holder-controlled devices or sending it to third-party devices. In the former case, the system must ensure the integrity of the displayed picture, to protect against attempts by a malicious holder to authenticate with somebody else's credentials whilst displaying their own picture. This problem might be solvable by resorting to TEEs[1] but does not seem to have easy solutions otherwise, since the holder might create a rogue presentation app that displays any picture of their choice. Even if the picture is cryptographically protected as part of the credential, a rogue holder app will just skip any verification.

> **Design principle 3**: Holder devices should not be trusted to correctly display and/or verify visual information of the holder.

The problem has an easy solution if we assume the existence of a trusted third party that has rolled out trusted devices to which the holder can send visual information, alongside any authentication/integrity information that is appropriately used to determine its correctness. Aside from the fact that this assumption might not be very realistic in several settings, we also contend that it is a bad design from a privacy perspective since its design accepts a collection point for personal identifiable information (PII) that may either be exploited by adversaries, leading to serious privacy breaches, or that may later suffer from benign or malicious function creep.

> **Design principle 4**: Involving third parties to handle digital visual information is not desirable.

2.1 Solution overview

This paper focuses on the design, analysis and implementation of a system for *anonymous credentials with visual holder authentication*. The design will respect the principles established in this section, introducing visual holder authentication to the well-known authentication paradigm of anonymous credentials without compromising on any privacy objective.

Given the discussion in the previous section, we require the visual authentication of the holder to solely rely on physical means, and to be conducted personally by the verifier. We however know that – as per design principle 1 – we must not rely on existing physical identification artefacts, since they

[1] We have not investigated this research direction which we leave as future work.

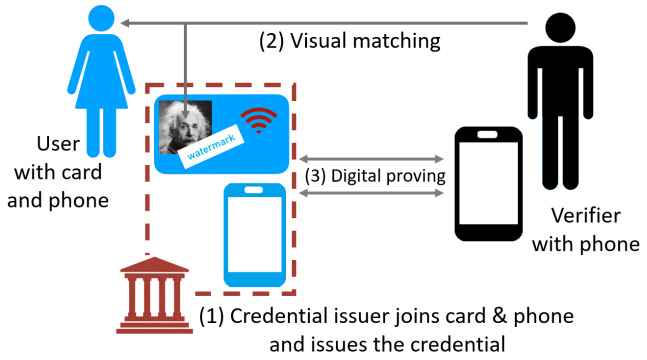

Figure 1: Overview of our anonymous credential system that binds digital proving (3) to visual authentication of the phone holder (2), with the help of picture-showing smartcards that are bound to phones by credential issueing authorities (1).

force the (digital) disclosure of additional information about the holder, and in so doing violate anonymity and unlinkability.

We therefore propose a solution where a picture of the holder is displayed on a new physical medium which we shall henceforth refer to as *card* or *smartcard* interchangeably. The card is issued by a trusted *card issuer*, which could be a government or an authority of similar standing. An overview of the entities and flows of our solution is depicted in Figure 1.

Similarly to other physical identification artefacts, this card must be hard to forge and must embed security features and markings that help credential issuing authorities to reliably ascertain its authenticity; however, we do not demand the same capability of the verifiers who may face adversarially programmed cards. Contrary to existing physical authentication artefacts such as identity cards or passports, we require that cards *do not* contain any information other than the picture of the holder: in particular, no personal data such as name, date of birth, etc., must be displayed on the card. This way we achieve data minimisation: digitally, the holder is able to disclose a minimum subset of attributes (using the selective disclosure property of attribute-based credentials); physically, the card only discloses a picture of the holder, whom the verifier sees in person anyway.

Helped by the card, our system enables two types of verification: i) an in-person, physical verification carried out by the verifier to check that the holder and their on-card picture match; ii) a digital verification of the credential. Note that the card must also take part in the latter verification in order to prevent mix-and-match attacks, where a malicious holder presents somebody else's credential and their own card. The card is thus required to possess an embedded electronic microprocessor and contactless smart card technology.

The card must additionally not behave differently with different parties (e.g. reveal secrets only if it talks to the holder). Aside from the additional complexity, such a design would also require the deployment of system-wide access control which may be exploited to violate the privacy of users.

2.2 High-level scheme design

An *anonymous credentials with visual holder authentication* system has the following actors: the *card*, the *holder*, the *verifier*, the *card issuer* and the *credential issuer*. The system can be comprised of arbitrarily many issuers, cards, holders, and verifiers. There are three types of interactions between the entities. First, a holder receives a card from a card issuer. Second, a card and a holder can jointly obtain an attribute-based credential from a credential issuer. We refer to this as the "join phase", during which the credential issuer verifies in person that the card matches its holder and is not a forgery, to then issue the credential. Third, a holder and a card can jointly convince any verifier of the possession of a credential over specific attributes. We call this the "presentation phase". We describe these three interactions here at a high-level:

- At first the holder authenticates, in person, to the card issuer and – upon success – obtains a smartcard from the latter; the card displays a picture of the holder, and other necessary markings to determine the card's authenticity. The card's digital infrastructure is equipped with a secret identifier uid (and other secret values) that will enable the minimal but mandatory contribution of the card in the digital authentication protocol; note that the identifier uid is *not* known by the holder.

- As in the previous phase, the holder is required to authenticate, in person, to a credential issuer, and produce a genuine card whose picture matches their visual traits. Then the holder obtains a digital credential from a credential issuer, wherein the credential issuer certifies a set of attributes of the holder. One of the attributes that is (blindly) certified is the card's identifier (uid). This requires the participation of the card: the credential issuer scans the card, obtaining the blind signature request component for the uid, which the credential issuer proceeds to sign, alongside the other attributes in the credential.

- The holder presents a digital credentials to a verifier. During a presentation the holder may choose to disclose certain attributes while keeping others secret (still proving their knowledge). This step is analogous to the traditional presentation step of anonymous credentials, but for one crucial difference: given that the uid is not known by the holder, it can neither be disclosed to the verifier, nor can its knowledge be proven by the holder. As a consequence, the card must be present during the authentication protocol, playing the following dual role: i) serve as reference for the visual authentication of the holder performed in person by the verifier; ii) be scanned

by the verifier and contribute to the cryptographic authentication protocol with the proof of knowledge of the undisclosed attribute uid.

2.3 Threat model

We describe here the threat model and objectives of the system.

2.3.1 Issuers

Card issuers are assumed to be honest and thus not corruptible by the adversary: i) individuals receive from them only cards with matching pictures (which also implies in-person verification and issuance); ii) uid values and other secrets are unique per card and are not shared with anyone else; iii) card issuers do not collude with any other entity in the system to violate, e.g., the privacy of holders or the anonymity of cards.

Credential issuers on the other hand need not be trusted. Formally, we allow credential issuers to be maliciously corrupted. The effect of such corruption is that the adversary fully controls the credential issuer. We note that, although a credential system with such a corrupt credential issuer cannot ensure unforgeability of credentials, our modeling of issuers still allows evaluation of whether, e.g., anonymity or privacy of holders is still guaranteed in the worst case of leaked issueing keys.

2.3.2 Holders and verifiers

Both holders and verifiers can arbitrarily deviate from the protocol. For example, corrupt holders could collude with each other to combine their credentials. Corrupt verifiers could, for example, enter the system with the sole purpose of learning holder's attributes, or of stealing their credentials and colluding with other verifiers in order to trace holders across multiple interactions and build holder profiles. Formally, this means we allow for static malicious corruptions of holders and verifiers. Contrary to credential issuers, we do not assume verifiers to be capable of detecting counterfeit cards. This means that verifiers could be presented with cards which the adversary programmed arbitrarily.

As discussed in the previous subsection, verifiers perform two verifications, a physical one to match holders and their picture, and a digital one to establish integrity and provenance of the credential. Our scheme guarantees anonymity and unlinkability of holders for the digital verification. Concerning the physical verification, a malicious verifier could try to remember the facial features of a holder they see often and generate an offline attribute profile based on that. They could also surreptitiously take pictures of holders and create a database where pictures (acting as primary keys) are linked to attributes. We consider these threats outside of the scope of our work: we will restrict our guarantees to the digital interactions, where biometric information is not digitally exchanged/recorded as part of routine processing, and where any and all transcripts of the physical verification are transient (humans forget, pictures are not taken, cctv tapes are eventually destroyed etc.). Our work also supports honest verifiers who want to avoid handling biometrics to prevent leaks/theft/liabilities. This is in contrast to protocols that rely on machine verifiers, where biometric data must be sent to the verifier digitally to perform the authentication successfully.

2.3.3 Smartcards

Smartcards are assumed to have computing/storage facility and NFC capabilities. Smartcards' local storage is expected to be tamper-resistant: secrets stored on the card are inaccessible to all actors. Smartcards are also assumed to guarantee the integrity of the processing logic: cards cannot be forced or tricked into deviating from the original program, and cannot be reprogrammed. These assumptions are consistent with current smartcard technology.

We assume that credential issuers are capable of detecting counterfeit cards, i.e., cards that have not been issued by the card issuer. On the other hand, verifiers are not necessarily required to be able to detect counterfeit cards: this means that verifiers could be presented with cards which the adversary programmed arbitrarily. Summarizing, we disallow corrupt cards to enter the join phase, but we allow malicious cards to enter the presentation phase. Crucially, such card corruptions are *static*, which means that the adversary cannot reprogram any smartcard issued by the card issuer, and it also does not get access to the internal values of any such card. We also assume that cards cannot (and must not) verify the identity of credential issuers or holders, and hence the adversary is allowed to interact with cards during all protocol phases.

We envision our system to be used in settings where a holder presents a smartcard to the issuer and to verifiers in person, i.e., we can assume them to be in physical proximity. Therefore, we can rule out the presence of network attackers, and hence we formally assume the availability of secure channels between all entities, in all three phases. We must also assume that relay/man-in-the-middle attacks can be prevented by the local proximity scanning settings, ensuring that the card being visually inspected is also generating the protocol messages: it should therefore be impossible for malicious holders to collude in order to show a counterfeit card and have it relay messages that are forwarded to and from the genuine card of another holder.

3 Joint Proof of Knowledge for BBS+

In this section, we present a novel proof of knowledge scheme for BBS+ signatures that requires two parties to contribute. This new primitive is necessary to construct our scheme since

we require credential presentations that require the joint participation of holder and card. In our design we strive to remain as compatible as possible to BBS+ signatures in order to preserve as much of the existing ecosystem (components, code, formats, standards) as possible.

The main idea is as follows. Instead of storing all attributes $(m_1,\ldots,m_\ell)$ on the holder, we "shave off" the attribute m_1 from the holder storage, and instead store m_1 on a smart card $\mathcal{C}$. Next, we modify the protocol for proving knowledge of a valid BBS+ message-signature pair in Section A.4 to enable the holder and the card to construct this proof jointly with *minimal* but *crucial* involvement of the card. Specifically, we decompose the prover algorithm $\mathcal{P}$ for the BBS+ proof of knowledge from Section A.4, into two PPT algorithms $(\mathsf{Prove}_{\mathcal{H}}(\tau,\cdot),\mathsf{Prove}_{\mathcal{C}}(\tau,\cdot))$ which share the state τ. Here $\mathsf{Prove}_{\mathcal{C}}$ is lightweight and executed by the smart-card $\mathcal{C}$ while $\mathsf{Prove}_{\mathcal{H}}$ is executed by the holder device $\mathcal{H}$. We provide details of the decomposition in Figure 2 and the overall protocol in Figure 3. The decomposition roughly works as follows: proving knowledge of BBS+ signature involves showing knowledge of exponents $(s,m_1,\ldots,m_\ell)$ over generators $h_0,\ldots,h_\ell$ which satisfies $h_0^s h_1^{m_1}\cdots h_\ell^{m_\ell}=P$ for publicly known P. In the above decomposition, card and holder generate shared randomness r using the PRF key K and then the card proves $h_1^{m_1}h_0^r=B$ and the holder proves $h_0^{s'}h_2^{m_2}\cdots h_\ell^{m_\ell}=PB^{-1}$ using $s'=s-r$, which convinces the verifier that they together know the entire vector. The shared state τ consists of a PRF key K and a non-hiding commitment $Q=h_1^{m_1}$ to message m_1 contributed by the card. All algorithms implicitly have public parameters as input. We state the security of our scheme in the following theorem and defer the proof to the full version [43].

Theorem 3.1. *Let* $q,k,n\in\mathbb{N}$ *and* $\mathsf{BBS}^+:=(\mathsf{KeyGen},\mathsf{Sign},\mathsf{Verify})$ *denote the BBS+ signature scheme with dimension* ℓ *over bilinear groups. If computation of discrete logarithms in* $\mathbb{G}_1$ *is hard,* SDL *is a signature of knowledge for relation* $\mathcal{R}_{q,k,n}$ (1)*, and* PRF *is a pseudorandom function, the protocol presented in Figure 3 satisfies* completeness*,* soundness *and* zero-knowledge *as defined below with respect to semi-honest verifier corruption and malicious card and holder corruption in the random-oracle model.*

- **Completeness***: the verifier* $\mathcal{V}$ *outputs* 1 *in the honest execution of the protocol whenever* $\mathsf{BBS}^+.\mathsf{Verify}(\mathsf{vk},(m_1,\ldots,m_\ell),(A,e,s))=1$.
- **Soundness***: There exists an efficient extractor* $\mathcal{E}$ *such that whenever a colluding card* $\mathcal{C}$ *and holder* $\mathcal{H}$ *succeed against an honest verifier* $\mathcal{V}$ *(*$\mathcal{V}$ *outputs 1),* $\mathcal{E}^{\mathcal{A}}(\mathbf{a}_V)$ *outputs* $(\mathbf{m},\sigma)$ *where* σ *is a verifying signature on* $\mathbf{m}\in\mathbb{Z}_p^\ell$ *with respect to public key* w*. Here* $\mathcal{A}$ *denotes the adversary corrupting* $\mathcal{H}$ *and* $\mathcal{C}$*.*
- **Zero Knowledge***: There exists a simulator* $\mathcal{S}$ *which simulates the view of a semi-honest verifier* $\mathcal{V}$ *in the protocol with honest* $\mathcal{C}$ *and* $\mathcal{H}$*.*

$\mathsf{Prove}_{\mathcal{C}}(K,m_1,n_1,n)$: // Executed by card

1. $r=\mathsf{PRF}_K(n)$.
2. $B=h_1^{m_1}h_0^r$. // h_0,h_1 from public params
3. $\pi\leftarrow\mathsf{SDL}\{(\alpha,\beta):h_1^\alpha h_0^\beta=B\}(n_1)$.
4. return (B,π).

$\mathsf{Prove}_{\mathcal{H}}(K,Q,(m_2,\ldots,m_\ell),(A,e,s),\mathbf{a}_V,n,B,n_2)$: // Executed by holder. All generators come from public parameters

1. Parse $\mathbf{a}_V=\{(i,v_i):i\in V\}$. Output $\perp$ if $m_i\neq v_i$ for some $i\in V$.
2. Set $r=\mathsf{PRF}_K(n)$. Output $\perp$ if $Qh_0^r\neq B$.
3. Set $H=\{2,\ldots,\ell\}\setminus V$.
4. $r_1\leftarrow\mathbb{Z}_p^*$, $r_2\leftarrow\mathbb{Z}_p$, $r_3=r_1^{-1}$, $s'=s-r_2r_3-r$.
5. $A'=A^{r_1}$, $b=g_1h_0^sQ\prod_{i=2}^{\ell}h_i^{m_i}$, $\bar{A}=A'^{-e}b^{r_1}$, $d=b^{r_1}h_0^{-r_2}$.
6. $\pi'\leftarrow\mathsf{SDL}\{(e,s',r_2,r_3,\{m_i\}_{i\in H}):A'^{-e}h_0^{r_2}=\bar{A}/d\wedge d^{-r_3}h_0^{s'}\prod_{i\in H}h_i^{m_i}=g_1^{-1}B^{-1}\prod_{i\in V}h_i^{-m_i}\}(n_1)$.
7. return $(A',\bar{A},d,\pi')$

Figure 2: Splitting a BBS+ proof of knowledge between card and holder

Remark: We only consider semi-honest verifier $\mathcal{V}$ in the above theorem as in the overall protocol the malicious behavior of $\mathcal{V}$ can be detected by the honest holder, which then outputs a *dummy* proof $\perp$. The overall protocol is outlined in Figures 5 and 6, with detailed description and proofs deferred to the full version of the paper [43].

4 A Model for Secure card-based Anonymous Credentials

We now present the core cryptographic building block of our anonymous credential scheme with visual holder authentication, which we call *card-based anonymous credentials* (cbAC). cbAC formally defines the interactions between holders, verifiers and credential issuers introduced in Section 2.2. We choose not to include card issuers as part of cbAC since visual verification and pictures of holders are no cryptographic procedures or objects.

Before proceeding, we introduce some notation regarding attribute formats. Throughout the paper, we assume attributes to sort into ℓ "categories" (e.g., birthdate, citizenship, or hair color). We denote by $L:=\{1,\ldots,\ell\}$ the full set of indices, and call any set $V\subseteq L$ an *index set*. Attributes from the category with index i, $i\in L$, can take values in *universe* $\mathcal{U}_i$. We often use shorthand notation $\mathbf{a}_V:=\{(i,m_i):i\in V\}$ for some $m_i\in$

- **Setup**: Generate a bilinear group $\mathsf{BG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, g_T, ep)$ and obtain $(\mathsf{sk}, \mathsf{vk}) := (x, (w, \bar{g}_1, \bar{g}_2, h_0, \ldots, h_\ell)) \leftarrow \mathsf{BBS}^+.\mathsf{KeyGen}$. Sample a PRF key $K \leftarrow \mathcal{K}$. Everything except x, K constitutes public parameters, denoted by pp.
- **Card's Inputs**: $0 \neq m_1 \in \mathbb{Z}_p$, PRF key K.
- **Holder's Inputs**: PRF key K, $Q = h_1^{m_1}$, message $m_2, \ldots, m_\ell \in \mathbb{Z}_p$, BBS+ signature $(A, e, s) \leftarrow \mathsf{BBS}^+.\mathsf{Sign}((m_1, m_2, \ldots, m_\ell), x)$.
- **Verifier's Inputs**: Attributes $\mathbf{a}_V = \{(i, v_i) : i \in V\}$, where V denotes the indices of attributes to be disclosed and v_i being the corresponding target values. We assume $1 \notin V$.
- **Protocol**: We denote card, holder and verifier by $\mathcal{C}$, $\mathcal{H}$ and $\mathcal{V}$ respectively.
  - $\mathcal{H} \to \mathcal{V}$: Nonce $n_{\mathcal{H}} \leftarrow \{0,1\}^\lambda$.
  - $\mathcal{V} \to \mathcal{C}$: $(n_{\mathcal{H}}, n_{\mathcal{V}})$ where $n_{\mathcal{V}} \leftarrow \{0,1\}^\lambda$.
  - $\mathcal{C} \to \mathcal{V}$: $n_{\mathcal{C}}, B, \pi$ where $n_{\mathcal{C}} \leftarrow \{0,1\}^\lambda$, $(B, \pi) \leftarrow \mathsf{Prove}_{\mathcal{C}}(K, m_1, n_{\mathcal{V}}, n_{\mathcal{C}} || n_{\mathcal{H}})$ (See Figure 2).
  - $\mathcal{V} \to \mathcal{H}$: $n_{\mathcal{C}}, n_{\mathcal{V}}, \mathbf{a}_V, B$.
  - $\mathcal{H}$: $\mathsf{proof} \leftarrow \mathsf{Prove}_{\mathcal{H}}(K, Q, (m_i)_{i=2}^{\ell}, (A, e, s), \mathbf{a}_V, n, B, n_{\mathcal{V}})$ for $n = n_{\mathcal{C}} || n_{\mathcal{H}}$ (an instantiation of $\mathsf{Prove}_{\mathcal{H}}$ is described in Figure 2).
  - $\mathcal{H} \to \mathcal{V}$: proof.
  - $\mathcal{V}$: If $\mathsf{proof} = \perp$ output 0, else parse proof as $(A', \bar{A}, d, \pi')$.
  - $\mathcal{V}$: Checks (i) $e(\bar{A}, g_2) = e(A', w)$ (ii) Proofs π and π' are valid. It outputs 1 if all the checks pass, and outputs 0 otherwise.

Figure 3: An interactive protocol for a card-based proof of BBS+ signature

$\mathcal{U}_i$ for each $i \in V$, in contexts where the concrete values of m_i are not relevant. We denote the "merge" of two attribute sets by $\{(i, m_i) : i \in V\} \oplus \{(j, m_j) : j \in H\} := \{(i, m_i) : i \in L\}$ which is only well-defined if the index sets V and H are disjoint, i.e., if the original attribute sets do not both contain attributes from the same category.

We start by giving an algorithmic description for cbAC. We use notation $\mathsf{Alg}(A : x, B : y) \to (B : z)$ to denote a (potentially interactive) procedure Alg where party A has input x, party B has input y, and party B outputs z.

Definition 4.1. *Let $\ell \in \mathbb{N}$, $L := \{0, \ldots, \ell\}$ and $\mathcal{U}_i$ denote attribute universes for all $i \in L$. A card-based anonymous credential (cbAC) system for $(\mathcal{U}_i)_{i \in L}$ is a set of three interactive procedures $(\mathsf{Setup}, \mathsf{Join}, \mathsf{Present})$, executed between an issuer I, and arbitrary cards $\mathcal{C}$, holders $\mathcal{H}$ and verifiers $\mathcal{V}$ as follows.*

$\mathsf{Setup}(I : \lambda) \to \mathsf{pp}$*: The setup algorithm is executed by the credential issuer and results in public parameters* pp *which are made available to all entities in the system.*

$\mathsf{Join}(\mathcal{H} : \mathbf{a}_H, \mathcal{C} : \perp, I : \mathbf{a}_I) \to (\mathcal{H} : b)$*: The join procedure is executed between one holder, one card, and the credential issuer, where both the holder and the credential issuer contribute attributes $\mathbf{a}_H \in (\mathcal{U}_i)_{i \in H}$ for $H \subseteq L$, $\mathbf{a}_I \in (\mathcal{U}_i)_{i \in I}$ for $I \subseteq L$. The output of the procedure is a bit b signaling either success or failure to the holder, and we require $b = 0$ if $L \setminus H \neq I$, i.e., ambiguity in attributes is not allowed.*

$\mathsf{Present}(\mathcal{H} : \perp, \mathcal{C} : \perp, \mathcal{V} : \mathbf{a}_V) \to (\mathcal{V} : f)$*: A presentation is executed between one verifier, one card, and one holder. Card and holder receive no input in this phase, but the verifier provides a set of attributes $\mathbf{a}_V \in (\mathcal{U}_i)_{i \in V}$ for some $V \subseteq L$. The result of the procedure is that the verifier outputs a bit f.*

We expect the following properties from a cbAC scheme.

Correctness/Completeness with selective disclosure Let $\mathbf{a}_H, \mathbf{a}_I$ denote two sets of attributes wrt. "index sets" H, I with $L \setminus H = I$. We say that a cbAC scheme satisfies *correctness with selective disclosure* if the following holds. Assume

$$\mathsf{Join}(\mathcal{H} : \mathbf{a}_H, \mathcal{C} : \perp, I : \mathbf{a}_I) = 1$$

for some holder $\mathcal{H}$, some card $\mathcal{C}$, and some attribute set $\mathbf{a}_H \in (\mathcal{U}_i)_{i \in H}, \mathbf{a}_I \in (\mathcal{U}_i)_{i \in I}$. Then it holds that for any $\mathbf{a}_V \subseteq (\mathbf{a}_H \oplus \mathbf{a}_I)$

$$\mathsf{Present}(\mathcal{H} : \perp, \mathcal{C} : \perp, \mathcal{V} : \mathbf{a}_V) = 1$$

Unforgeability We next demand unforgeability of credentials, namely that it be computationally infeasible for a holder $\mathcal{H}'$ and card $\mathcal{C}$ to convince any verifier of the possession of attributes that were not jointly issued to them.

More formally, assume $\mathsf{Present}(\mathcal{H}' : \perp, \mathcal{C} : \perp, \mathcal{V} : \mathbf{a}_V) = 1$ for $\mathbf{a}_V \in (\mathcal{U}_i)_{i \in V}$ for some $V \subseteq L$, some holder $\mathcal{H}'$, and some card $\mathcal{C}$. Then we say that the cbAC scheme is *unforgeable* if there exists a holder $\mathcal{H}$ with inputs $\mathbf{a}_H \in (\mathcal{U}_i)_{i \in H}$ for some $H \subseteq L$ who participated in a join procedure with $\mathcal{C}$ resulting in $\mathsf{Join}(\mathcal{H} : \mathbf{a}_H, \mathcal{C}' : \perp, I : \mathbf{a}_I) = 1$, where $\mathcal{H} = \mathcal{H}'$ or both $\mathcal{H}, \mathcal{H}'$ are corrupt, and $\mathbf{a}_V \subseteq \mathbf{a}_I \oplus \mathbf{a}_H$.

Anonymity From anonymity we understand the inability to recognize the repeated participation of an entity in the digital part of the protocol. We consider anonymity of both cards and holders, while previous works consider only joint anonymity [42], and we consider them from both the perspective of honest-but-curious credential issuers and from the perspective of malicious verifiers. However, as common in the AC literature (e.g., [4, 42, 52]), we only guarantee such anonymity for cards and holders that were joined by *the same* credential issuer. This restriction is natural since credentials

are not expected to hide the public key of the credential issuer. Naturally, anonymity can only be guaranteed provided that the set of disclosed attributes does not trivially deanonymize their holder, i.e. the set of disclosed attributes must be identical.

More detailed, we require cards and holders that were joined by the same credential issuer to remain anonymous during a presentation, i.e., a potentially malicious verifier cannot detect which card or holder participated. These guarantees must even hold if the verifier has access to the issuance transcripts, and the internal state of the issuer, e.g., its signing keys.

More formally, we call a cbAC scheme *anonymous during presentation* if the following holds. Consider executions of

$$\mathsf{Join}(\mathcal{H}^0 : \mathbf{a}_H^0, \mathcal{C}^0 : \perp, I : \mathbf{a}_I^0) = 1 \text{ and}$$

$$\mathsf{Join}(\mathcal{H}^1 : \mathbf{a}_H^1, \mathcal{C}^1 : \perp, I : \mathbf{a}_I^1) = 1$$

for inputs $(\mathbf{a}_H^0, \mathbf{a}_H^1, \mathbf{a}_I^0, \mathbf{a}_I^1, \mathcal{C}^0, \mathcal{C}^1, \mathcal{H}^0, \mathcal{H}^1) \leftarrow \mathcal{V}$ provided by any PPT adversary $\mathcal{V}^I$, who has access to the internal state and incoming/outgoing messages of the semi-honest issuer I. Let b denote a random bit. Then $\mathcal{V}^I$ participating in

$$\mathsf{Present}(\mathcal{H}^b : \perp, \mathcal{C}^b : \perp, \mathcal{V}^I : \mathbf{a}_V)$$

where $\mathbf{a}_V \subseteq ((\mathbf{a}_H^0 \oplus \mathbf{a}_I^0) \cap (\mathbf{a}_H^1 \oplus \mathbf{a}_I^1))$ outputs b with advantage negligibly close to $1/2$.

Holder privacy during presentation We next demand strong privacy properties for the holder when presenting attributes. Namely, even a malicious verifier does not learn more information about the attributes of the holder than what is revealed by the outcome of the presentation.

More formally, consider a PPT adversary $\mathcal{V}$ outputting $\mathbf{a}_L^0, \mathbf{a}_L^1, \mathbf{a}_V$ such that $\mathbf{a}_V \subseteq \mathbf{a}_L^0$ and $\mathbf{a}_V \subseteq \mathbf{a}_L^1$, and the identity of some honest $\mathcal{C}$. Let b denote a randomly chosen bit upon which we execute procedures

$$\mathsf{Join}(\mathcal{H} : \mathbf{a}_L^b, \mathcal{C} : \perp, I : \perp) = 1$$

and

$$\mathsf{Present}(\mathcal{H} : \perp, \mathcal{C} : \perp, \mathcal{V} : \mathbf{a}_V) = 1.$$

We say that a cbAC scheme has *holder privacy* if the advantage of any such PPT $\mathcal{V}$ outputting b is negligibly close to $1/2$ over the random coins of all the involved entities in the execution.

Unlinkability of presentations Unlinkability of presentations demands that a malicious verifier cannot link two presentations, i.e., tell whether the same card or the same holder was involved in them. Unlinkability implies anonymity and we can hence define it by strengthening the adversary in the anonymity definition above.

More formally, unlinkability is defined as anonymity but where the adversary $\mathcal{V}$ additionally gets take part in arbitrarily many executions of

$$\mathsf{Present}(\mathcal{H}^i : \perp, \mathcal{C}^i : \perp, \mathcal{V} : \mathbf{a}_V)$$

for $i \in \{0,1\}$ before being challenged and making his decision.

Blind issuance of attributes We finally define a property that we consider optional for cbAC in general but useful for some applications. Blind issuance of attributes demands that the holder be able to contribute attributes to the credential that are not seen by the credential issuer.

More formally, let $\mathcal{C}, \mathcal{H}, H, I \subseteq L$ with $L \setminus I = H$ and $\mathbf{a}_H^0, \mathbf{a}_H^1, \mathbf{a}_I$ all be given by any PPT adversary $\mathcal{A}$. Let b denote a uniformly sampled bit. Consider a run of

$$\mathsf{Join}(\mathcal{H} : \mathbf{a}_H^b, \mathcal{C} : \perp, I : \mathbf{a}_I) = 1$$

where $\mathcal{A}$ can observe the internal state of I and sees all messages that I receives. We say that a cbAC scheme supports *blind issuance of attributes* if the advantage of any such PPT $\mathcal{A}$ outputting b is negligibly close to $1/2$.

Use cases for blind issuance include issuance of sensitive attributes such as gender, or protection against credential leakage (e.g., holder's device stores attribute sk in secure storage, such that without knowledge of sk a captured credential is rendered useless).

Interpreting the formal properties for the real-world system. As explained in the beginning of this Section, our formal cbAC model captures the guarantees of the *cryptographic* part of our card-based credential system. Visual verification of pictures on smartcards, and entities such as holders and verifiers being humans who meet in person, are not part of the cryptographic protocol. Consequently, e.g., the anonymity guarantees described above only capture that cards leave no *digital* traces of their identity during a run of the protocol. That said, a verifier can always attempt to somehow *remember* a picture on a smartcard, to deanonymize that card in future presentations. The same obviously holds for employees working at an issueing authority, who can attempt to remember faces of the smartphone owners. Such threats need to be taken into account when deploying the actual system, for example verifiers should perform verification in front of the user, to ensure that no pictures of the smartcard are taken.

Further, care needs to be taken when translating the above guarantees into practice. For example, anonymity only holds for card-holder pairs that were joined by the same credential issuer, such that the anonymity set corresponds to all "customers" of a credential issuer. Consequently, our cbAC system does not provide any meaningful anonymity guarantees in settings where each card-holder pair receives their credential from a different credential issuer. It is an interesting avenue

for future work whether techniques to hide the identity of the credential issuer [9, 13, 18] could be applied to our work.

In Figure 4 we provide a formal modeling of cbAC in terms of an ideal functionality in the UC framework which captures all notions described here. Our proofs are carried out with respect to that functionality. For space constraints, we refer the reader to [43] for an explanation of $\mathcal{F}_{\mathsf{cbAC}}$.

5 Our card-based Anonymous Credential scheme

In this section we describe how we realise the three interactions described in Section 2.2: credentials received during the join phase are BBS+ signature from the credential issuer on a message vector $\mathbf{m} = (\mathsf{uid}, m_1, \ldots, m_\ell)$, where uid is a unique secret identifier of the card, and $m_1, \ldots, m_\ell$ represent the attributes of the holder. Issuing the credential in practice amounts to generating a blind signature over a set of shown and hidden messages. The set of hidden messages will contain at least the term uid, and possibly, but not mandatorily, other sensitive attributes that the holder does not want to reveal to the credential issuer.

During the presentation phase the holder discloses a set of certified attributes and proves knowledge of the complement of that set. This phase instantiates the joint proof of knowledge of a BBS+ signature of Figure 3, to let a holder who ignores the term uid generate a valid proof of knowledge of a BBS+ signature that contains it. This is enabled by the presence of some card $\mathcal{C}$.

One feature of our protocol is that cards never need to communicate with holders directly. This greatly simplifies the card's interface during a presentation session, and avoids the deployment of expensive authentication mechanisms that let cards distinguish between holders and verifiers.

We depict our protocol in Figures 5 (Join) and 6 (Presentation), where card and holder call the joint BBS+ proving algorithms from Figure 2. Note that Figure 6 is essentially a visualization of Figure 3. These figures do not show the setup phase and interaction with trusted parties, and they are cleaned from any "cluttering" that is introduced by the UC framework.

5.1 Security

Theorem 5.1 (cbAC security without blind issuance)**.** *The cbAC construction of Section 5 UC-emulates the functionality $\mathcal{F}_{\mathsf{cbAC}}$ parametrized with $\ell \in \mathbb{N}$ in the $(\mathcal{F}_{\mathsf{crs}}, \mathcal{F}_{\mathsf{cardAuth}})$-hybrid model under the following assumptions:*

- *The adversary does not corrupt any $\mathcal{C}$ that enters the join phase. All other corruptions are static and malicious.*
- *Holder inputs are restricted to $\mathbf{a}_H = \emptyset$ in* JOIN.
- *All channels are secure. Channels during presentation are additionally holder- and card-anonymous.*
- PRF *is a pseudorandom function with key space* $\{0,1\}^\lambda$*, and* SDL *is a signature of knowledge for relation $\mathcal{R}_{s,k,n}$ (see A.2.1).*
- *Computation of discrete logarithms is hard in $\mathbb{G}_1$ and the qSDH assumption holds in* BG.

Our construction supports blind issuance of attributes under a stronger assumption on the proof system SDL, namely *online extractability*, which we describe in the full version of this work [43].

Theorem 5.2 (cbAC security with blind issuance)**.** *Theorem 5.1 holds for $\mathcal{F}_{\mathsf{cbAC}}^{\mathsf{blind}}$ if additionally* SDL *is online extractable, and the restriction on holder inputs is dropped.*

The full formal proof and simulator code are deferred to the full version [43].

6 Evaluation

We implement the card-specific part of the scheme and test it on real smartcards to establish the scheme's practicality and determine its performance. We also implement a simplified verifier to determine whether the cards' messages are properly constructed and can be verified successfully[2].

We choose to implement the card as a Javacard [32] applet. Javacard is a Java framework for smartcards supporting a subset of the Java runtime. The applet supports byte-level I/O through smartcard application protocol data unit (APDUs). APDUs can contain selectors for different functions, and the applet is structured to handle the different functions. The Javacard framework supports operations on elliptic curves. Points on various elliptic curves might be built by selecting the curve type ($\mathbf{F}_p$ or $\mathbf{F}_{2^m}$) and specifying the relevant parameters. We structure our applet with the 5 following functions (with reference to Figure 2):

- a set of initialisation functions: SETUP, where the invoker sets the parameters of the curve. We choose to test on Fp256BN curve [29][3] and so the input APDU for the setup contains the value of the a and b coefficients, the value of field and order of the curve, and coordinates and cofactor of the generator; SETBASE where the invoker sets the public bases h_0 and h_1; SETUID where the invoker sets the secret value of m_1; SETSEED where the invoker sets the value of K, the PRF seed used by the card. We use AES in ECB mode to instantiate a PRF with domain and codomain of all 128-bit strings.
- a RUN function that executes $\mathsf{Prove}_C(K, m_1, n_1, n)$; K and m_1 are already set by SETSEED and SETUID, re-

[2] We refrain from implementing issuer, holder and verifier since their practicality and performance has already been established by prominent open source projects such as Hyperledger Ursa [39]

[3] We choose Fp256BN given the sundry available implementations, even though this curve no longer offers 128 bits of security.

$\mathcal{F}_{\mathsf{cbAC}}$ is instantiated with session identifier $\mathsf{sid} = (I, sid')$ for some sid', which we omit from all interfaces. $\mathcal{F}_{\mathsf{cbAC}}$ maintains join session records $\mathsf{JR}(\mathsf{jid}) = (\mathtt{C}, \mathtt{H}, \boxed{\mathtt{attH}}, \mathtt{attI})$ and presentation session records $\mathsf{SR}(\mathsf{vid}) = (\mathtt{C}, \mathtt{H}, \mathtt{attV})$. These records are initialized with all values set to $\perp$ when accessed for the first time. Let creds denote an initially empty list. Interfaces of Join and Show can only be called after Setup was completed. We assume $\mathcal{F}_{\mathsf{cbAC}}$ to ignore malformed inputs. $\mathcal{F}_{\mathsf{cbAC}}$ is parametrized by $\ell \in \mathbb{N}$ and we denote $L := \{1, \ldots, \ell\}$.

Setup Phase: Initialize the functionality instance.
On input (SETUP) from a party I

- S.1 Ignore if this is not the first SETUP query
- S.2 From now on, use I to denote the unique party that is allowed to call interface JOINISSUE.
- S.3 Send (SETUP) to $\mathcal{S}$ and a delayed output $(\mathsf{SETUPDONE}, \mathsf{sid})$ to I.

Join Phase: $\boxed{\text{Holder inputs attributes } \mathbf{a}_H = \{(i, m_i) : i \in H\},}$ issuer inputs attributes $\mathbf{a}_I = \{(i, m_i) : i \in I\}$ where $\boxed{I = L \setminus H}$ $\boxed{\text{---}\ I = L\ \text{---}}$. The card-holder pair is coupled to the attribute vector $\mathbf{m} = (m_1, \ldots, m_\ell)$ contained in $\boxed{\mathbf{a}_H}$ and $\mathbf{a}_I$.

`Holder Requests`: On input $(\mathsf{JOIN}, \mathsf{jid}, \mathcal{C}, \boxed{\mathbf{a}_H})$ from $\mathcal{H}$

- J-H.1 Drop the query if $\perp \neq \mathsf{JR}(\mathsf{jid}).\mathtt{C} \neq \mathcal{C}$ or if $\mathsf{JR}(\mathsf{jid}).\mathtt{H} \neq \perp$. Otherwise set $\mathsf{JR}(\mathsf{jid}).\mathtt{H} \leftarrow \mathcal{H}, \mathsf{JR}.(\mathsf{jid}).\mathtt{C} \leftarrow \mathcal{C}$, $\boxed{\text{and } \mathsf{JR}(\mathsf{jid}).\mathtt{attH} \leftarrow \mathbf{a}_H}$.
- J-H.2 Output $(\mathsf{JOIN}, \mathsf{jid}, H, \mathcal{H})$ to $\mathcal{S}$. // No anonymity for holder in join phase.

`Card joins`: On input $(\mathsf{JOINID}, \mathsf{jid})$ from $\mathcal{C}$

- J-C.1 Drop the query if $\perp \neq \mathsf{JR}(\mathsf{jid}).\mathtt{C} \neq \mathcal{C}$. Otherwise set $\mathsf{JR}.(\mathsf{jid}).\mathtt{C} \leftarrow \mathcal{C}$.
- J-C.2 Output $(\mathsf{JOINID}, \mathsf{jid}, \mathcal{C})$ to $\mathcal{S}$. // No anonymity for card in the join phase.

`Issuer Agrees`: On input $(\mathsf{JOINISSUE}, \mathsf{jid}, \mathbf{a}_I)$ from I

- J-I.1 $\boxed{\text{---}\ \text{Drop the query if the index set } I \text{ of } \mathbf{a}_I \text{ is not equal to } L\ \text{---}}$.
- J-I.2 Create record $\mathsf{JR}(\mathsf{jid})$ with $\mathtt{C} \leftarrow \perp$, $\mathtt{H} \leftarrow \perp$, $\boxed{\mathtt{attH} \leftarrow \perp}$, $\mathtt{attI} \leftarrow \mathbf{a}_I$ if no such record exists. Otherwise, set $\mathsf{JR}(\mathsf{jid}).\mathtt{attI} \leftarrow \mathbf{a}_I$ if $\mathsf{JR}(\mathsf{jid}).\mathtt{attI} = \perp$. // Add attributes contributed by issuer to the record.
- J-I.3 Send $(\mathsf{JOINISSUE}, \mathsf{jid})$ to $\mathcal{S}$ and send a delayed output $(\mathsf{JOINISSUE}, \mathsf{jid}, \mathbf{a}_I)$ to $\mathcal{H}$.

`Finalize the join`: On input $(\mathsf{JOINCOMPLETE}, \mathsf{jid})$ from $\mathcal{S}$

- J.1 Ignore if there is no record $\mathsf{JR}(\mathsf{jid})$, or any of its variables is $\perp$.
- J.2 $\boxed{\text{Parse } \mathsf{JR}(\mathsf{jid}).\mathtt{attH} \text{ as } \{(i, p_i) : i \in H\}, \mathsf{JR}(\mathsf{jid}).\mathtt{attI} \text{ as } \{(i, q_i) : i \in I\} \text{ for some } H, I \subseteq L \text{ and drop the query if } H \neq L \setminus I.}$
- J.3 Construct $\mathbf{m} = (m_1, \ldots, m_\ell)$ as follows: Set $m_i = q_i$ for $i \in I$, $\boxed{m_i = p_i \text{ for } i \in H.}$
- J.4 Add $(\mathsf{JR}(\mathsf{jid}).\mathtt{H}, \mathsf{JR}(\mathsf{jid}).\mathtt{C}, \mathbf{m})$ to creds. // Credential installed: $\mathcal{H}$ and $\mathcal{C}$ can from now on show $\mathbf{m}$.
- J.5 Delete record $\mathsf{JR}(\mathsf{jid})$ and send a delayed output $(\mathsf{JOINED}, \mathsf{jid})$ to $\mathcal{H}$.

Presentation Phase: During a presentation phase, the card-holder pair authenticates against a set of attributes $\mathbf{a}_V = \{(i, m_i) : i \in V\}$ specified by the verifier $\mathcal{V}$. They succeed if they have been previously coupled to vector $\mathbf{m} \in \mathbb{Z}_p^\ell$ such that $m_i = \mathbf{m}[i]$ for $i \in V$.

`Set Attributes`: On input $(\mathsf{SETATTRS}, \mathsf{vid}, \mathbf{a}_V)$ from $\mathcal{V}$

- P-V.1 Create record $\mathsf{SR}(\mathsf{vid}) = (\perp, \perp, \mathbf{a}_V)$ if no such record exists. Otherwise, set $\mathsf{SR}(\mathsf{vid}).\mathtt{attV} \leftarrow \mathbf{a}_V$ if $\mathsf{SR}(\mathsf{vid}).\mathtt{attV} = \perp$.
- P-V.2 Output $(\mathsf{SETATTRS}, \mathsf{vid}, \mathcal{V}, \mathbf{a}_V)$ to $\mathcal{S}$. // Verifier's identity and attributes are public.

`Set Card`: On input $(\mathsf{SETID}, \mathsf{vid})$ from $\mathcal{C}$

- P-C.1 Drop the query if $\mathcal{C}$ is honest and $(*, \mathcal{C}, *) \notin$ creds. // Uninitialized card.
- P-C.2 Create record $\mathsf{SR}(\mathsf{vid})$ with $\mathtt{C} \leftarrow \mathcal{C}$, $\mathtt{H} \leftarrow \perp$, $\mathtt{attV} \leftarrow \perp$ if no such record exists. Otherwise, set $\mathsf{SR}(\mathsf{vid}).\mathtt{C} \leftarrow \mathcal{C}$ if $\mathsf{SR}(\mathsf{vid}).\mathtt{C} = \perp$.
- P-C.3 If $\mathcal{H}$ is corrupt, and $(\mathcal{H}', \mathcal{C}, *) \in$ creds for corrupt $\mathcal{H}'$, output $(\mathsf{SETID}, \mathsf{vid}, \mathcal{C})$ to $\mathcal{S}$, else output $(\mathsf{SETID}, \mathsf{vid})$ to $\mathcal{S}$. // Card remains anonymous as long as holder is not corrupt and has already used that card during issuance.

`Set Credential`: On input $(\mathsf{SETCRED}, \mathsf{vid})$ from $\mathcal{H}$

- P-H.1 Create record $\mathsf{SR}(\mathsf{vid}) = (\perp, \mathcal{H}, \perp)$ if no such record exists. Otherwise, set $\mathsf{SR}(\mathsf{vid}).\mathtt{H} \leftarrow \mathcal{H}$ if $\mathsf{SR}(\mathsf{vid}).\mathtt{H} = \perp$.
- P-H.2 Output $(\mathsf{SETCRED}, \mathsf{vid})$ to $\mathcal{S}$. // Holder remains anonymous.

`Verify`: On input $(\mathsf{VERIFYCOMPLETE}, \mathsf{vid}, b)$ from $\mathcal{S}$

- V.1 Ignore if there is no record $\mathsf{SR}(\mathsf{vid})$, or any of its variables is $\perp$. // Card or holder missing.
- V.2 Parse $\mathsf{SR}(\mathsf{vid}).\mathtt{C}$ as $\mathcal{C}$, $\mathsf{SR}(\mathsf{vid}).\mathtt{H}$ as $\mathcal{H}$, $\mathsf{SR}(\mathsf{vid}).\mathtt{attV}$ as $\{(i, m_i) : i \in V\}$ for some $V \subseteq L$.
- V.3 Set $f = 0$. // Unforgeability: only change this to 1 below if there is a corresponding credential in creds.
- V.4 If $\mathcal{H}$ is honest and $b = 1$, set $f = 1$ if $(\mathcal{H}, \mathcal{C}, \mathbf{m}) \in$ creds such that $\mathbf{m}[i] = m_i \; \forall \, i \in V$. // Completeness
- V.5 If $\mathcal{H}$ is corrupt and $b = 1$, set $f = 1$ if $(\mathcal{H}', \mathcal{C}, \mathbf{m}) \in$ creds such that $\mathbf{m}[i] = m_i \; \forall \, i \in V$ and some corrupt party $\mathcal{H}'$. // Corrupt holders can exchange their credentials.
- V.6 If $\mathcal{H}$ and I are both corrupt and $b = 1$, set $f = 1$. // No unforgeability if issuer and holder collude.
- V.7 Delete record $\mathsf{SR}(\mathsf{vid})$ and send a delayed output $(\mathsf{VERIFIED}, \mathsf{vid}, f)$ to $\mathcal{V}$.

Figure 4: Functionalities $\mathcal{F}_{\mathsf{cbAC}}^{\mathsf{blind}}$ and $\mathcal{F}_{\mathsf{cbAC}}$ for card-based anonymous credentials. Instructions in $\boxed{\text{boxes}}$ appear only in $\mathcal{F}_{\mathsf{cbAC}}^{\mathsf{blind}}$ which allows the holder to contribute (blind) attributes during join. Instructions in $\boxed{\text{dashed boxes}}$ appear only in $\mathcal{F}_{\mathsf{cbAC}}$ where holders do not contribute attributes during join.

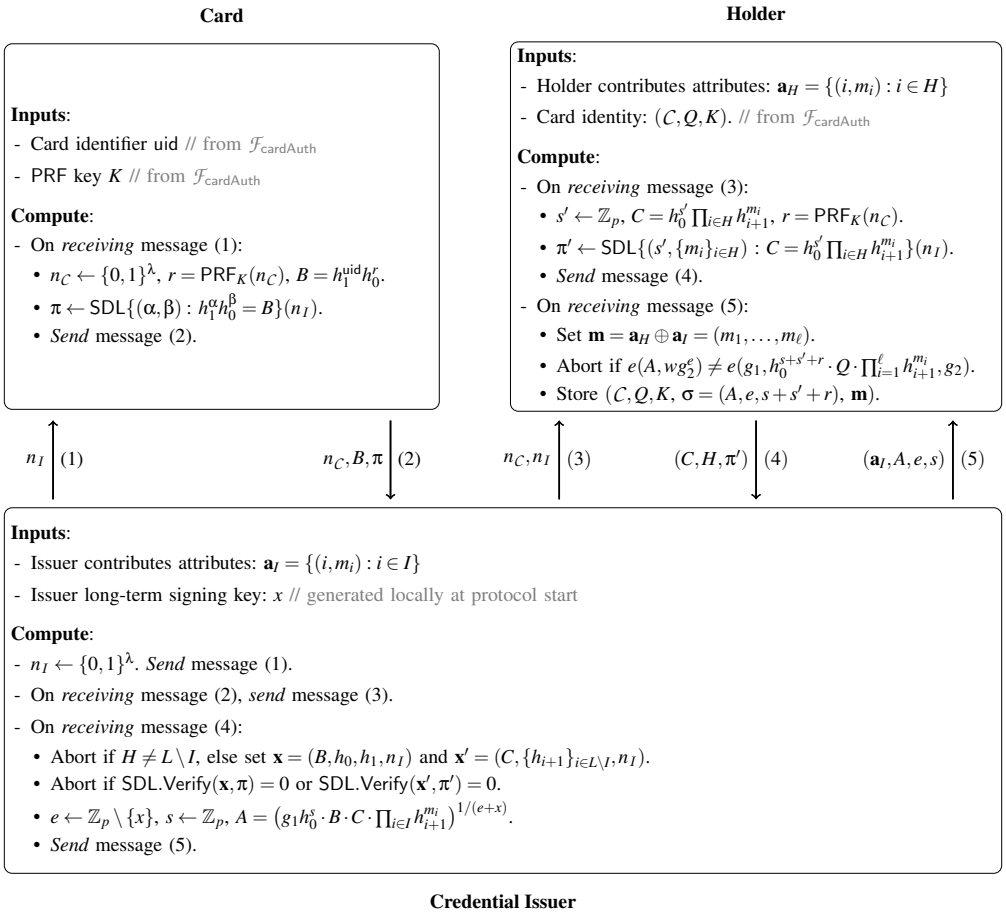

Figure 5: Outline of join protocol. The protocol delivers a credential certifying attributes assembled from holder input $\mathbf{a}_H$, issuer's input $\mathbf{a}_I$ for holder and the card identified by its uid. Detailed UC-style protocol description appears in full version of this paper [43].

spectively, so the input of the invocation are the verifier nonce n_1 and the PRF input n.

We assume that the initialisation functions can be invoked once before the card is issued. RUN can instead be invoked arbitrarily many times by whoever is in proximity of the card. RUN requires no access control and always responds in the same way, irrespective of the identity of the invoker.

For the deployment, we choose NXP 1ID white plastic cards with a Smart MX D600 chip (400KiB of available memory) running JCOP 4.5 OS with NXP's JCOPx extensions at version 1.1.4[4]. Cards have a dual interface (6 PIN contact, 1ID inlay 56pf contactless on input CAP). We use the contactless communication channel and rely on a uTrust 3700 F as a PCSC reader to program the cards, communicate with them and benchmark them. Alternative designs may employ the native NFC capabilities available in most modern mobile platforms.

[4]The extensions are needed to directly access the low-level API to perform scalar point multiplication.

We benchmark the RUN algorithm by executing it 100 times and determining average and standard deviation of all samples. To determine the breakdown of the running time we also benchmark a no-op version of the RUN algorithm where only the I/O part is implemented.

The RUN function completes on average in 185.08ms with a standard deviation of 4.06ms. This is an end-to-end measurement that includes APDU I/O, parsing, crypto and receipt of the response APDU. To determine a breakdown of the running time we construct a no-op version of the RUN function which is identical to the original but for the fact that all cryptographic processing is removed, thus resulting only in the APDU I/O, parsing and producing a response message of identical length to the original. The no-op version completes on average in 24.53ms with a standard deviation of 4.3ms.

These result confirm that the performance of the scheme is perfectly in line with that of other NFC-driven interactions users engage in on a daily basis (e.g. contactless payments), thus confirming the viability of our approach and its deploy-

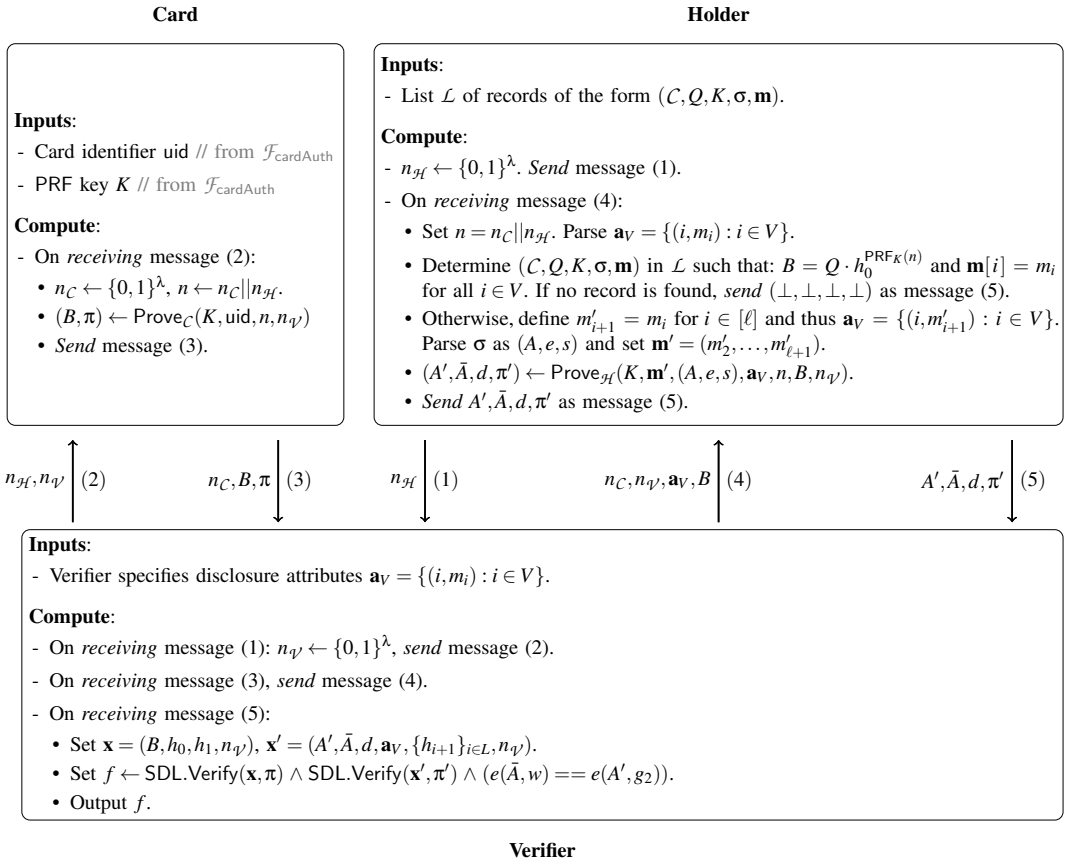

Figure 6: Outline of presentation protocol. The algorithms Prove_C and $\mathsf{Prove}_{\mathcal{H}}$ are as in Figure 2. In the above protocol, verifier specifies presentation predicate as $\mathbf{a}_V$. The holder generates proof of possession of a credential issued against the participating card. Detailed UC-style protocol description appears in full version of this paper [43].

ment readiness.

7 Conclusion

We present *Anonymous Credentials with Visual Holder Authentication*, a system that permits secure and privacy preserving verification of digital credentials. This system enhances digital anonymous credential system by permitting verifiers to determine whether the holder presenting the credential is its legitimate owner. This determination can be performed without any deterioration of the privacy guarantees offered by the underlying anonymous credential system. The key idea is to introduce plastic cards from a trusted issuer (e.g. a government), playing a role in both the physical (verifier inspects the picture on the card) and digital (verification requires contributions from both card and holder) side of the authentication.

To realise this system we present a primitive for joint proof of knowledge for BBS+ signatures, which is both central for our construction and of independent interest. We then formally define card-based Anonymous Credentials, our cryptographic building block, and analyse its security in the Universal Composability (UC) framework. We further implement the performance-sensitive aspects of our system, namely all interactions of the card, to determine real-world viability.

Our system achieves several desirable properties: i) it is compatible with BBS+ public keys and signatures, so that implementers are able to leverage the vast body of open-source projects handling those artefacts; ii) it keeps the design of the card simple and minimalistic, avoiding complex access control on the card; iii) it maintains the familiar pattern of authentication with phone and physical ID; and iv) it achieves ideal privacy by not forcing the disclosure of unnecessary holder attributes when determining a match with physical ID.

Acknowledgements

The authors would like to thank: the anonymous reviewers for the helpful and constructive comments; Ilie Circiumaru, Jim Coon, Todd Nuzum and Nick Nurse for their help in setting up the evaluation environment. Alessandro wishes to say *"arrivederci, mio meraviglioso papà, e grazie per tutto"*.

References

[1] M. H. Au, W. Susilo, and Y. Mu. Constant-size dynamic k-TAA. In R. D. Prisco and M. Yung, editors, *SCN 06: 5th International Conference on Security in Communication Networks*, volume 4116 of *Lecture Notes in Computer Science*, pages 111–125, Maiori, Italy, Sept. 6–8, 2006. Springer, Heidelberg, Germany.

[2] M. Backes, L. Hanzlik, K. Kluczniak, and J. Schneider. Signatures with flexible public key: Introducing equivalence classes for public keys. In T. Peyrin and S. Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 405–434, Brisbane, Queensland, Australia, Dec. 2–6, 2018. Springer, Heidelberg, Germany.

[3] A. Barki, S. Brunet, N. Desmoulins, S. Gambs, S. Gharout, and J. Traoré. Private eCash in practice (short paper). In J. Grossklags and B. Preneel, editors, *FC 2016: 20th International Conference on Financial Cryptography and Data Security*, volume 9603 of *Lecture Notes in Computer Science*, pages 99–109, Christ Church, Barbados, Feb. 22–26, 2016. Springer, Heidelberg, Germany.

[4] L. Batina, J. Hoepman, B. Jacobs, W. Mostowski, and P. Vullers. Developing efficient blinded attribute certificates on smart cards via pairings. In D. Gollmann, J. Lanet, and J. Iguchi-Cartigny, editors, *Smart Card Research and Advanced Application, 9th IFIP WG 8.8/11.2 International Conference, CARDIS 2010, Passau, Germany, April 14-16, 2010. Proceedings*, volume 6035 of *Lecture Notes in Computer Science*, pages 209–222. Springer, 2010.

[5] A. Beduschi. Rethinking digital identity for post-covid-19 societies: Data privacy and human rights considerations. *Data & Policy*, 3, 2021.

[6] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In D. E. Denning, R. Pyle, R. Ganesan, R. S. Sandhu, and V. Ashby, editors, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, Nov. 3–5, 1993. ACM Press.

[7] D. Bernhard, M. Fischlin, and B. Warinschi. Adaptive proofs of knowledge in the random oracle model. In J. Katz, editor, *PKC 2015: 18th International Conference on Theory and Practice of Public Key Cryptography*, volume 9020 of *Lecture Notes in Computer Science*, pages 629–649, Gaithersburg, MD, USA, Mar. 30 – Apr. 1, 2015. Springer, Heidelberg, Germany.

[8] P. Bichsel, J. Camenisch, T. Groß, and V. Shoup. Anonymous credentials on a standard java card. In E. Al-Shaer, S. Jha, and A. D. Keromytis, editors, *ACM CCS 2009: 16th Conference on Computer and Communications Security*, pages 600–610, Chicago, Illinois, USA, Nov. 9–13, 2009. ACM Press.

[9] J. Bobolz, F. Eidens, S. Krenn, S. Ramacher, and K. Samelin. Issuer-hiding attribute-based credentials. *IACR Cryptol. ePrint Arch.*, page 213, 2022.

[10] D. Boneh and X. Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, Apr. 2008.

[11] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55, Santa Barbara, CA, USA, Aug. 15–19, 2004. Springer, Heidelberg, Germany.

[12] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *2012 IEEE Symposium on Security and Privacy*, pages 553–567, San Francisco, CA, USA, May 21–23, 2012. IEEE Computer Society Press.

[13] D. Bosk, D. Frey, M. Gestin, and G. Piolle. Hidden issuer anonymous credential. *Proc. Priv. Enhancing Technol.*, 2022(4):571–607, 2022.

[14] S. A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, Cambridge, MA, USA, 2000.

[15] E. Brickell, L. Chen, and J. Li. Simplified security notions of direct anonymous attestation and a concrete scheme from pairings. Cryptology ePrint Archive, Report 2008/104, 2008. https://eprint.iacr.org/2008/104.

[16] E. Brickell and J. Li. Enhanced privacy ID: A direct anonymous attestation scheme with enhanced revocation capabilities. Cryptology ePrint Archive, Report 2007/194, 2007. https://eprint.iacr.org/2007/194.

[17] E. F. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In V. Atluri, B. Pfitzmann, and P. McDaniel, editors, *ACM CCS 2004: 11th Conference on Computer and Communications Security*, pages 132–145, Washington, DC, USA, Oct. 25–29, 2004. ACM Press.

[18] J. Camenisch, M. Drijvers, and M. Dubovitskaya. Practical uc-secure delegatable credentials with attributes and their application to blockchain. In B. Thuraisingham,

D. Evans, T. Malkin, and D. Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 683–699. ACM, 2017.

[19] J. Camenisch, M. Drijvers, P. Dzurenda, and J. Hajny. Fast keyed-verification anonymous credentials on standard smart cards. Cryptology ePrint Archive, Report 2019/460, 2019. https://eprint.iacr.org/2019/460.

[20] J. Camenisch, M. Drijvers, and A. Lehmann. Anonymous attestation using the strong Diffie Hellman assumption revisited. Cryptology ePrint Archive, Report 2016/663, 2016. https://eprint.iacr.org/2016/663.

[21] J. Camenisch, M. Drijvers, and A. Lehmann. Universally composable direct anonymous attestation. In C.-M. Cheng, K.-M. Chung, G. Persiano, and B.-Y. Yang, editors, *PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 9615 of *Lecture Notes in Computer Science*, pages 234–264, Taipei, Taiwan, Mar. 6–9, 2016. Springer, Heidelberg, Germany.

[22] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In B. Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118, Innsbruck, Austria, May 6–10, 2001. Springer, Heidelberg, Germany.

[23] J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In S. Cimato, C. Galdi, and G. Persiano, editors, *SCN 02: 3rd International Conference on Security in Communication Networks*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289, Amalfi, Italy, Sept. 12–13, 2003. Springer, Heidelberg, Germany.

[24] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In M. Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72, Santa Barbara, CA, USA, Aug. 15–19, 2004. Springer, Heidelberg, Germany.

[25] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups (extended abstract). In B. S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424, Santa Barbara, CA, USA, Aug. 17–21, 1997. Springer, Heidelberg, Germany.

[26] J. Camenisch and E. Van Herreweghen. Design and implementation of the idemix anonymous credential system. In V. Atluri, editor, *ACM CCS 2002: 9th Conference on Computer and Communications Security*, pages 21–30, Washington, DC, USA, Nov. 18–22, 2002. ACM Press.

[27] M. Chase, S. Meiklejohn, and G. Zaverucha. Algebraic MACs and keyed-verification anonymous credentials. In G.-J. Ahn, M. Yung, and N. Li, editors, *ACM CCS 2014: 21st Conference on Computer and Communications Security*, pages 1205–1216, Scottsdale, AZ, USA, Nov. 3–7, 2014. ACM Press.

[28] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. 28(10), 1985.

[29] I. J. S. . T. Committee. ISO/IEC 15946-5:2022: Cryptographic techniques based on elliptic curves. https://www.iso.org/standard/80241.html, 2022.

[30] A. Connolly, P. Lafourcade, and O. P. Kempner. Improved constructions of anonymous credentials from structure-preserving signatures on equivalence classes. Cryptology ePrint Archive, Report 2021/1680, 2021. https://eprint.iacr.org/2021/1680.

[31] V. core development. Veramo. https://github.com/uport-project/veramo, 2023.

[32] O. Corporation. Oracle java card technology. 2023.

[33] A. de la Piedra, J. Hoepman, and P. Vullers. Towards a full-featured implementation of attribute based credentials on smart cards. volume 8813 of *Lecture Notes in Computer Science*, pages 270–289. Springer, 2014.

[34] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

[35] J. Doerner, Y. Kondi, E. Lee, abhi shelat, and L. Tyner. Threshold bbs+ signatures for distributed anonymous credential issuance. Cryptology ePrint Archive, Paper 2023/602, 2023. https://eprint.iacr.org/2023/602.

[36] eHealth Network. Interoperability of health certificates: Trust framework. 2021.

[37] M. Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In V. Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages

152–168, Santa Barbara, CA, USA, Aug. 14–18, 2005. Springer, Heidelberg, Germany.

[38] H. Foundation. Hyperledger indy. https://github.com/hyperledger/indy-node, 2023.

[39] H. Foundation. Hyperledger ursa. https://github.com/hyperledger/ursa, 2023.

[40] H. Gunasinghe and E. Bertino. Privbiomtauth: Privacy preserving biometrics-based and user centric protocol for user authentication from mobile phones. *IEEE Trans. Inf. Forensics Secur.*, 13(4):1042–1057, 2018.

[41] C. Hanser and D. Slamanig. Structure-preserving signatures on equivalence classes and their application to anonymous credentials. In P. Sarkar and T. Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 491–511, Kaoshiung, Taiwan, R.O.C., Dec. 7–11, 2014. Springer, Heidelberg, Germany.

[42] L. Hanzlik and D. Slamanig. With a little help from my friends: Constructing practical anonymous credentials. In G. Vigna and E. Shi, editors, *ACM CCS 2021: 28th Conference on Computer and Communications Security*, pages 2004–2023, Virtual Event, Republic of Korea, Nov. 15–19, 2021. ACM Press.

[43] J. Hesse, N. Singh, and A. Sorniotti. How to bind anonymous credentials to humans. Cryptology ePrint Archive, Paper 2023/853, 2023. https://eprint.iacr.org/2023/853.

[44] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In J. Motiwalla and G. Tsudik, editors, *ACM CCS 99: 6th Conference on Computer and Communications Security*, pages 28–36, Singapore, Nov. 1–4, 1999. ACM Press.

[45] M. Khalili, D. Slamanig, and M. Dakhilalian. Structure-preserving signatures on equivalence classes from standard assumptions. Cryptology ePrint Archive, Report 2019/1120, 2019. https://eprint.iacr.org/2019/1120.

[46] T. Looker, V. Kalos, A. Whitehead, and M. Lodder. The BBS Signature Scheme. Internet-Draft draft-irtf-cfrg-bbs-signatures-01, Internet Engineering Task Force, Oct. 2022. Work in Progress.

[47] W. Lueks, B. Hampiholi, G. Alpár, and C. Troncoso. Tandem: Securing keys by using a central server while preserving privacy. *Proc. Priv. Enhancing Technol.*, 2020(3):327–355, 2020.

[48] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. In H. M. Heys and C. M. Adams, editors, *SAC 1999: 6th Annual International Workshop on Selected Areas in Cryptography*, volume 1758 of *Lecture Notes in Computer Science*, pages 184–199, Kingston, Ontario, Canada, Aug. 9–10, 1999. Springer, Heidelberg, Germany.

[49] MATTR. bbs-signatures. *GitHub repository*, 2023.

[50] W. Mostowski and P. Vullers. Efficient u-prove implementation for anonymous credentials on smart cards. In M. Rajarajan, F. Piper, H. Wang, and G. Kesidis, editors, *Security and Privacy in Communication Networks - 7th International ICST Conference, SecureComm 2011, London, UK, September 7-9, 2011, Revised Selected Papers*, volume 96 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 243–260. Springer, 2011.

[51] W. Mostowski and P. Vullers. Efficient u-prove implementation for anonymous credentials on smart cards. In *Security and Privacy in Communication Networks*, pages 243–260. Springer Berlin Heidelberg, 2012.

[52] C. Paquin and G. Zaverucha. U-prove cryptographic specification v1.1 (revision 3), December 2013.

[53] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000.

[54] L. Rila and C. J. Mitchell. Security protocols for biometrics-based cardholder authentication in smartcards. In J. Zhou, M. Yung, and Y. Han, editors, *ACNS 03: 1st International Conference on Applied Cryptography and Network Security*, volume 2846 of *Lecture Notes in Computer Science*, pages 254–264, Kunming, China, Oct. 16–19, 2003. Springer, Heidelberg, Germany.

[55] A. Sonnino, M. Al-Bassam, S. Bano, S. Meiklejohn, and G. Danezis. Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*.

[56] Trinsic. Okapi. https://github.com/trinsic-id/okapi, 2023.

[57] P. Vullers and G. Alpár. Efficient selective disclosure on smart cards using idemix. In S. Fischer-Hübner, E. de Leeuw, and C. J. Mitchell, editors, *Policies and Research in Identity Management - Third IFIP WG 11.6 Working Conference, IDMAN 2013, London, UK, April 8-9, 2013. Proceedings*, volume 396 of *IFIP Advances in Information and Communication Technology*, pages 53–67. Springer, 2013.

A Preliminaries and Notation

A.1 Bilinear groups

An asymmetric bilinear type-3 group generator is a PPT algorithm BGen that takes as input the security parameter λ and outputs a tuple $\mathsf{BG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, g_T, e, p)$, where

- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are cyclic groups of order p, where p is an λ-bit prime.
- $\mathbb{G}_1 = \langle g_1 \rangle, \mathbb{G}_2 = \langle g_2 \rangle, \mathbb{G}_T = \langle g_T \rangle$.
- $e : \mathbb{G}_1 \times G_2 \to \mathbb{G}_T$ is an efficiently computable non-degenerate bilinear map.
- There is no efficiently computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$.

A.2 Signature of Knowledge

We define signatures of knowledge, and later use their instantiations for discrete-log like relations in the random oracle model [6, 37].

Definition A.1 (Signature of Knowledge)**.** *Let $\mathcal{U}_\lambda$ denote the set of functions from $\{0,1\}^*$ to $\{0,1\}^\lambda$. A pair of PPT algorithms $\mathsf{SoK} = (\mathsf{Prove}, \mathsf{Verify})$ are called a* signature of knowledge *for NP relation $\mathcal{R}$ if all of the following hold, where A^X means that algorithm A has oracle access to function or algorithm X.*

- **Completeness***: For all $(x,w) \in \mathcal{R}$, $m \in \{0,1\}^*$, $\mathsf{H} \in \mathcal{U}_\lambda$, and $\pi \leftarrow \mathsf{SoK.Prove}^{\mathsf{H}}(x,m,w)$ we have $\mathsf{SoK.Verify}^{\mathsf{H}}(x,m,\pi) = 1$.*

- **Zero-Knowledge [7]***: There exists a PPT simulator $\mathsf{SoK.Sim}$ which emulates a random-oracle $\mathsf{SoK.Sim.H}$ such that the below holds for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.*

$$\Pr\left[\begin{array}{c|r} \mathcal{A}_2^{\mathsf{H}_b}(\mathsf{state}, \pi_b) = b \wedge & (x,m,w,\mathsf{state}) \leftarrow \mathcal{A}_1(1^\lambda) \\ \mathcal{R}(x,w) = 1 & \pi_0 \leftarrow \mathsf{SoK.Prove}^{\mathsf{H}}(x,m,w) \\ & \pi_1 \leftarrow \mathsf{SoK.Sim}(x,m) \\ & b \leftarrow \{0,1\} \end{array}\right]$$

is negligible in λ, where $\mathsf{H}_0() = \mathsf{H}()$ for $\mathsf{H} \leftarrow \mathcal{U}_\lambda$ and $\mathsf{H}_1() = \mathsf{SoK.Sim.H}()$.

- **Argument-of-Knowledge***: There exists a PPT extractor $\mathsf{SoK}.\mathcal{E}$ such that for any PPT adversary $\mathcal{A}$ the following holds with overwhelming probability (over random choices of PPT algorithms and random oracle H): $(x,m,\pi) \leftarrow \mathcal{A}^{\mathsf{H}}$, $w \leftarrow \mathsf{SoK}.\mathcal{E}^{\mathcal{A},\mathsf{H}}(x,m,\pi)$, $(x,w) \in \mathcal{R} \vee \neg\mathsf{SoK.Verify}^{\mathsf{H}}(x,m,\pi)$. Here the notation $\mathsf{SoK}.\mathcal{E}^{\mathcal{A},\mathsf{H}}$ denotes that $\mathsf{SoK}.\mathcal{E}$ can access queries (and answers) made by $\mathcal{A}$ to H as well as query H itself. Additionally $\mathsf{SoK}.\mathcal{E}$ has access to copies of initial state of $\mathcal{A}$ to which it can simulate its own random oracle.*

A.2.1 Signatures of Knowledge for discrete log

In this paper, relations of interest to us are over cyclic groups of prime order. Let $\mathbb{G}$ be a cyclic group of prime order p. For integers $s,k,n \geq 1$, we consider relations $\mathcal{R}_{s,k,n}$ consisting of pairs (x,w) with $x = (y_1,\ldots,y_s,g_1,\ldots,g_k) \in \mathbb{G}^{s+k}$, $w = (\alpha_1,\ldots,\alpha_n) \in \mathbb{Z}_p^n$ such that:

$$(x,w) \in \mathcal{R}_{s,k,n} \Leftrightarrow \bigwedge_{i=1}^{s} \mathcal{L}_i(x,w) \tag{1}$$

where each $\mathcal{L}_i(x,w)$ is of the form $\prod_{j=1}^{t_i} g_{ij}^{\alpha_{ij}} = y_i$ with $\{g_{i1},\ldots,g_{it_i}\} \subseteq \{g_1,\ldots,g_k\}$ and $\{\alpha_{i1},\ldots,\alpha_{it_i}\} \subseteq \{\alpha_1,\ldots,\alpha_n\}$. We call the elements $(y_1,\ldots,y_\ell)$ in the statement x as "commitments", while $(g_1,\ldots,g_k)$ are referred to as "generators". Constructions of signatures of knowledge for relations $\mathcal{R}_{s,k,n}$ as defined above are presented in [25] and we recap it below.

Lemma A.1 ([25])**.** *Let $s,k,n \in \mathbb{N}$ and $\mathbb{G}$ denote a cyclic group. There exist a signature of knowledge $\mathsf{SDL} = (\mathsf{Prove}, \mathsf{Verify})$ for relation $\mathcal{R}_{s,k,n}$ in the random oracle model, assuming the hardness of computing discrete logarithms in $\mathbb{G}$.*

These signatures SDL are also proven to be *existentially unforgeable* [53] under the same assumptions. Following the notation introduced in [25], we use

$$\pi \leftarrow \mathsf{SDL}\{(\alpha_1,\ldots,\alpha_n) : \bigwedge_{i=1}^{s} \mathcal{L}_i(x,w)\}(m)$$

to denote the output of $\mathsf{SDL.Prove}^{\mathsf{H}}(x,m,w)$ where $(x,w) \in \mathcal{R}_{s,k,n}$ and $m \in \{0,1\}^*$. When m is the empty string $\perp$, we omit (m) in the above notation, and call π a *proof of knowledge*.
Notation: All constructions of signatures of knowledge as defined in Definition A.1 use a concrete hash function $\mathcal{H}$, which models the access to random-oracle H in the definition. Thus, notation for algorithms $\mathsf{SDL.Prove}$, $\mathsf{SDL.Verify}$ will not specify oracle access.

A.3 BBS+ Signature Scheme

BBS+ signatures were introduced in [1], building upon the BBS signatures introduced by [11]. Subsequently, the construction in [1] was adapted to asymmetric bilinear groups by Camenisch et al. [20].

Definition A.2 (BBS+ Signatures [20])**.** *Let $\mathsf{BG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, g_T, e, p) \leftarrow \mathsf{BGen}(1^\lambda)$ denote a bilinear group and $\ell \in \mathbb{N}$. Then the BBS+ signature scheme over BG with dimension ℓ is described by the algorithms $\mathsf{BBS}^+ := (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ as below:*

– KeyGen*: Sample $h_0,\ldots,h_\ell \leftarrow \mathbb{G}_1^{\ell+1}$, $x \leftarrow \mathbb{Z}_p$, $w \leftarrow g_2^x$, $\bar{g}_1 \leftarrow \mathbb{G}_1$, $\bar{g}_2 \leftarrow \bar{g}_1^x$. Set $sk = x$ and $vk = (w, \bar{g}_1, \bar{g}_2, h_0,\ldots,h_L)$.*

– Sign: *On input message* $\mathbf{m} = (m_1, \ldots, m_\ell) \in \mathbb{Z}_p^\ell$ *and secret key* x*, sample* $e \leftarrow \mathbb{Z}_p \backslash \{x\}$*,* $s \leftarrow \mathbb{Z}_p$*, compute* $A = \left(g_1 h_0^s \prod_{i=1}^{\ell} h_i^{m_i}\right)^{1/(e+x)}$*. Output* (A, e, s) *as the signature on* $\mathbf{m}$.

– Verify: *On input a public key* $(w, h_0, \ldots, h_\ell)$*, message* $\mathbf{m} = (m_1, \ldots, m_\ell)$ *and signature* $\sigma = (A, e, s)$*, output* $e(A, w g_2^e) \stackrel{?}{=} e(g_1 h_0^s \prod_{i=1}^{\ell} h_i^{m_i}, g_2)$.

The above signature scheme is proven to be *existentially unforgeable under chosen message attack* (EUF-CMA) under the q-Strong Diffie-Hellman Assumption (qSDH) in BG [10], which demands that no efficient adversary given the $q+3$ tuple $(g_1, g_1^x, g_1^{x^2}, \ldots, g_1^{x^q}, g_2, g_2^x) \in \mathbb{G}_1^{q+1} \times \mathbb{G}_2^2$ can output $(c, g_1^{1/(x+c)}) \in \mathbb{Z}_p \setminus \{-x\} \times \mathbb{G}_1$, except with negligible probability.

A.4 Proof of Knowledge for BBS+ Signatures

We describe the proof of knowledge of BBS+ signature as presented in [20]. The prover in possession of a BBS+ signature (A, e, s) on attributes $(m_1, \ldots, m_\ell)$, selectively discloses the attributes $\mathbf{a}_V = \{(i, m_i) : i \in V\}$ to a verifier $\mathcal{V}$ as follows: The prover chooses $r_1 \leftarrow \mathbb{Z}_p^*$, $r_2 \leftarrow \mathbb{Z}_p$ and computes $r_3 = 1/r_1$. Next it computes $A' = A^{r_1}$, $b = g_1 h_0^s \prod_{i=1}^{\ell} h_i^{m_i}$, $\overline{A} = A'^{-e} b^{r_1}$, $d = b^{r_1} h_0^{-r_2}$. Finally, the prover computes proof π as:

$$\pi \leftarrow \mathsf{SDL}\{(e, s, r_2, r_3, \{m_i\}_{i \in H}) : A'^{-e} h_0^{r_2} = \overline{A}/d \wedge d^{-r_3} h_0^s \prod_{i \in H} h_i^{m_i} = g_1^{-1} \prod_{i \in V} h_i^{-m_i}\}$$

In the above, the set $H = L \setminus V$ corresponds to undisclosed attributes $\mathbf{a}_H = \{(i, m_i) : i \in H\}$. The prover sends $(A', \overline{A}, d, \pi)$ to the verifier, who checks $e(A', w) = e(\overline{A}, g_2)$ and verifies the proof π against the statement computed from $A', \overline{A}, d, V$. For proof of completeness, zero knowledge and argument-of-knowledge we refer the reader to Section 4 in [20].

A.5 Signatures over committed messages

The BBS+ signature scheme outlined above allows an issuer to sign attributes while only knowing a commitment over them. This feature of the BBS+ signature scheme is used for realising the version of our scheme featuring *blind issuance* property. The protocol below due to Au et al. [1] allows a holder to obtain signature over message vector $(m_1, \ldots, m_\ell)$ where $\{m_i : i \in H\}$ are hidden from the issuer for some publicly known sets H, L with $H \subseteq L$.

- Holder computes commitment $C = h_0^{s'} \prod_{i \in H} h_i^{m_i}$ and the proof $\pi \leftarrow \mathsf{SDL}\{(s', \{m_i\}_{i \in H}) : h_0^{s'} \prod_{i \in H} h_i^{m_i} = C\}$. It sends (C, π) to the issuer.
- The issuer verifies $b \leftarrow \mathsf{SDL.Verify}((C, \{h_i\}_{i \in H}), \pi)$. The issuer aborts if the verification fails. Otherwise, it computes the signature as: $e \leftarrow \mathbb{Z}_p^* \backslash \{x\}$, $s \leftarrow \mathbb{Z}_p$, $A = (g_1 h_0^s \cdot C \cdot \prod_{i \in L \setminus H} h_i^{m_i})^{1/(e+x)}$. It sends $\mathbf{a}_I, (A, e, s)$ to the holder.
- The holder sets $\sigma = (A, e, s + s')$ as the signature over the message vector $(m_1, \ldots, m_\ell)$.

Signatures over committed messages have applications in scenarios where users must get their cryptographic secrets (such as signing keys) certified by an issuer without revealing the secrets to the issuer.