Toward a Post-Quantum Zero-Knowledge Verifiable Credential System for Self-Sovereign Identity

Simone Dutto[1], Davide Margaria[2], Carlo Sanna[1], and Andrea Vesco[2]

[1] DISMA, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino (IT)
{simone.dutto,carlo.sanna}@polito.it
https://crypto.polito.it

[2] LINKS Foundation, Via Pier Carlo Boggio 61, 10138 Torino (IT)
{davide.margaria,andrea.vesco}@linksfoundation.com
https://linksfoundation.com

Abstract. The advent of quantum computers brought a large interest in post-quantum cryptography and in the migration to quantum-resistant systems. Protocols for Self-Sovereign Identity (SSI) are among the fundamental scenarios touched by this need. The core concept of SSI is to move the control of digital identity from third-party identity providers directly to individuals. This is achieved through Verifiable Credentials (VCs) supporting anonymity and selective disclosure. In turn, the implementation of VCs requires cryptographic signature schemes compatible with a proper Zero-Knowledge Proof (ZKP) framework.
We describe the two main ZKP VCs schemes based on classical cryptographic assumptions, that is, the *signature scheme with efficient protocols* of Camenisch and Lysyanskaya, which is based on the strong RSA assumption, and the BBS+ scheme of Boneh, Boyen and Shacham, which is based on the strong Diffie–Hellman assumption. Since these schemes are not quantum-resistant, we select as one of the possible post-quantum alternatives a lattice-based scheme proposed by Jeudy, Roux-Langlois, and Sander, and we try to identify the open problems for achieving VCs suitable for selective disclosure, non-interactive renewal mechanisms, and efficient revocation.

Keywords: Post-quantum Cryptography · Self-Sovereign Identity · Verifiable Credentials · Zero-Knowledge Proof

1 Introduction

Self-Sovereign Identity (SSI) [19] is a new model for digital identity on the internet and it promises to be one of the most important trends for the coming decades. The core concept of SSI is to move the control of digital identity from third-party "identity providers" directly to individuals, meaning that the information regarding the identity of each user must be controlled by the

user itself. The SSI framework leverages the decentralized identity paradigm and it builds upon two major standards of the World Wide Web Consortium (W3C): the Decentralized IDentifiers (DIDs) [23] and the Verifiable Credentials (VCs) [24]. These W3C standards make the SSI framework highly interoperable and portable.

In 2003, Camenisch and Lysyanskaya [7] started a new trend in the field of anonymous credentials by proposing a secure protocol for signing a committed message with anonymity features and security based on the strong RSA assumption. Their proposal is a combination of:

- a commitment scheme, which allows the user to keep the message secret;
- a blind signature, that describes how the issuer can sign the message through its commitment;
- a zero-knowledge proof of knowledge, exploited by the user to prove to a verifier the validity of the received signature on the committed message.

One year later [8], they published an alternative version based on bilinear maps, with a reference to a possible usage of the contemporary work of Boneh, Boyen, and Shacham [3], which obtained short group signatures based on the strong Diffie–Hellman assumption using bilinear group. This was the beginning of the BBS+ signature scheme [2] and its applications to verifiable anonymous credentials.

However, both the scheme of Camenisch and Lysyanskaya and of Boneh, Boyen, and Shacham are based on cryptographic assumptions that are not quantum-resistant. Therefore, the identification and assessment of novel solutions leveraging Post-Quantum Cryptography (PQC) is an active research topic. The NIST PQC standardization process [1] focuses on two basic protocols, namely Key Encapsulation Mechanisms (KEM) and digital signatures; and how to implement VCs from these building blocks is a highly nontrivial problem.

Thus, the urge for a migration to new post-quantum primitives. Among the proposal at the NIST PQC standardization process or the consequent independent works from the scientific community, the main theoretical advances have been achieved from the European Project PROMETHEUS [22]. This project aims to provide post-quantum signature schemes, encryption schemes and privacy-preserving protocols relying on lattices.

In 2022, Jeudy, Roux-Langlois and Sanders [13], inspired by the previous work of Camenisch and Lysyanskaya, proposed a new version of the anonymous signature scheme based on the learning with error problem, which is also at the base of the security for different finalist KEMs in the NIST PQC standardization process, namely CRYSTALS-KYBER and SABER, as well as for the alternate candidate FrodoKEM, and also for the FALCON digital signature. This is only a starting point, for the frameworks can be more complex and have different requirements. Some of the addressable improvements are the construction of anonymous and selective disclosure credentials, non-interactive renewal mechanisms and efficient revocation.

Given this background, the contribution of this paper is threefold:

1. we review the SSI framework and two classical zero-knowledge schemes suitable to a VC system for SSI (*i.e.* a RSA-based scheme and a DH-based scheme), highlighting their main peculiarities and cryptographic primitives;
2. we select and describe the solution of Jeudy, Roux-Langlois, and Sanders [13] as a promising post-quantum alternative with respect to the previous schemes;
3. we present some solutions for the efficient revocation of VCs and for counteracting the misuse of classical VCs, including some practical examples of credentials based on the previous schemes.

The paper is organized as follows: Section 2 describes the SSI framework and the different layers of its stack; Section 3 motivates the adoption of Zero-Knowledge Proofs in the SSI framework; Section 4 provide the mathematical details related to two classical zero-knowledge VCs schemes; Section 5 identifies and describes a quantum-resistant zero-knowledge VC scheme; Section 6 presents an efficient credential revocation mechanism suitable to previous schemes; Section 7 proposes the Holder Binding as an effective solution against the misuse of classical VCs, providing some practical examples of credentials; finally, Section 8 concludes the paper with the final remarks and future research directions.

2 Self-Sovereign Identity Framework

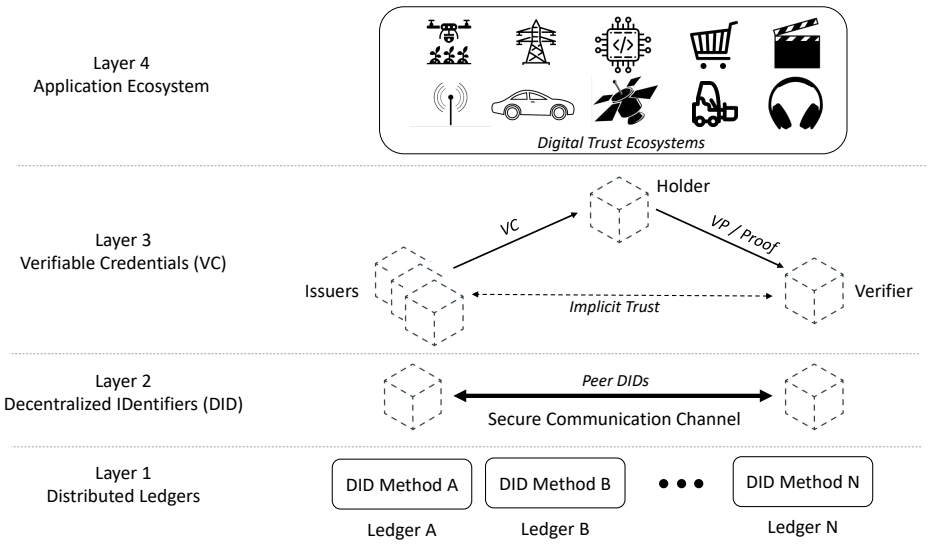

Fig. 1. Self-Sovereign Identity (SSI) framework stack.

The overall SSI stack is depicted in Fig. 1. Layer 1 is made of a Distributed Ledger Technology (DLT) acting as the Root-of-Trust (RoT) of the overall frame-

work. The DLT [14] works as a distributed storage of identity data where trust in data and their immutability is ensured by the underlying consensus protocol that the DLT adopts. Many of such protocols exist today but, when categorizing them by the structure of the ledger, two main classes emerge: blockchains (*e.g.* Bitcoin [16], Ethereum [4]) and directed acyclic graphs (*e.g.* IOTA Tangle [17]).

A Decentralized IDentifier (DID) [23] is a new type of globally unique identity designed to verify a subject (*i.e.* human beings and things). DIDs are designed to enable a subject to have control over its own identity in accordance with the Self-Sovereign Model. The DIDs are in the form of URI and they associate a DID subject with a DID Document [23] allowing trustable interactions associated with that subject.

Layer 1 also includes the DID Method [23], a software implementation to interact with the specific ledger. A DID method must provide the primitives to create the identity (*i.e.* generate two pairs of keys and the DID Document, and store it to the ledger at the DID address), to resolve a DID (*i.e.* retrieve a DID Document from the ledger address pointed to by the DID and verify the correct format of the DID Document), to Update the DID (*i.e.* generate a new DID and DID Document) and to revoke a DID (*i.e.* provide an evidence on the ledger that a DID and related DID Document has been revoked by the controller). The DID Method implementation is indeed ledger-specific and it makes the upper layers independent from the ledger adopted.

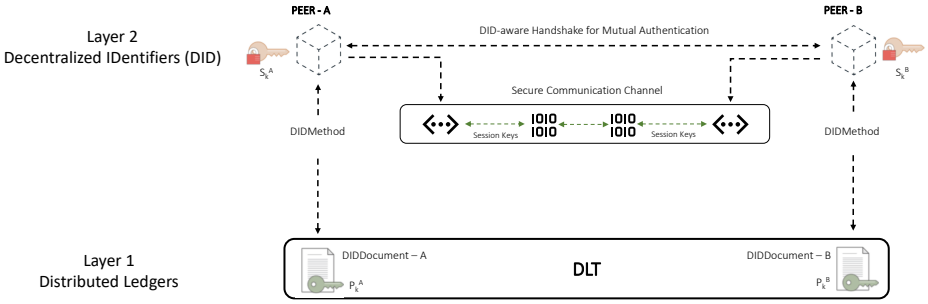

Fig. 2. DID-based DPKI for mutual-authentication and secure channel establishment.

Layer 2 makes use of DIDs and DID Documents to authenticate a peer and establish a secure communication channel (*i.e.* mutual-authentication, confidentiality and integrity) in accordance with a Decentralized Public-Key Infrastructure (DPKI) paradigm, as depicted in Fig. 2, where the DLT is the RoT. The DID Document contains the cryptographic public material and the verification methods to prove control of the DID (*i.e.* control of the cryptographic private material) at the core of secure channel establishment. The DID Document data model is encoded in JSON format by mapping property values to JSON types.

For instance, Fig 3 presents an example of DID Document that contains two public keys for authentication and assertion purposes, respectively.

Fig. 3. Example of DID Document.

Layer 2 leverages DID technology (*i.e.* the security foundation of the SSI framework) to start the authentication procedure.

As shown in Fig. 1, Layer 3 finalizes authentication and deals with authorization to services and resource access through Verifiable Credentials (VCs) [24]. VCs provide a mechanism to express digital credentials in a way that is cryptographically secure and machine verifiable. A VC is an unforgeable digital document that contains any further characteristic of the digital identity than a simple key pair and a DID. It represents all of the same information that a physical credential represents, but the addition of digital signatures makes VCs more tamper-evident and more trustworthy than their physical counterparts. The combination of key pairs, a DID, a DID Document and at least one VC forms the digital identity in the SSI framework.

Layer 3 works around the classical Triangle-of-Trust depicted in Fig. 1. Three different roles are expected to coexist in the classical setup:

- Holder: is the element of the system that possesses one or more VCs and that generates appropriate Verifiable Presentations (VPs) to a Verifier to request a service or a resource;
- Issuer: is the element of the system that asserts claims about one or more subjects, creating a VC from these claims, and transmitting the VC to a Holder. The subject is the entity about which claims are made (*e.g.* a human being or a thing). Very often the Holder and the subject coincide, but in some

cases the Holder can also possess a credential asserting claims on a different subject;
– Verifier: is the element of the system that receives one or more VPs, for processing (*i.e.* verification of cryptographic proof on the VCs generated by Issuers and Holders).

A VC contains some metadata to describe properties of the credential such as the context, ID, type, Issuer of the VC, issuance, and expiration date. Most importantly, a VC contains in the `credentialSubject` field one or more claims made by the same Issuer, then used by the Verifier to grant access to a service or a resource. A claim is a statement about a subject (any kind of statement is possible) and is expressed using subject-property-value relationships. The cryptographic proof is made by the Issuer to make the VC an unforgeable and verifiable digital document to state properties/characteristics/features. The VC data model is encoded in JSON format by mapping property values to JSON types as depicted on the left side of Fig. 4.

The Holder requests access to services and/or resources to the Verifier by presenting a VP as in the right side of Fig. 4. A VP is built as an envelop of the VC issued by an Issuer where proof is made by the Holder.

Classical Verifiable Credential (VC)

```
VerifiableCredential = {
    "@context": [
        "https://www.w3.org/2018/credentials/v1",
        "https://www.w3.org/2018/credentials/examples/v1"
    ],
    "id": "http://example/credentials/1",
    "type": ["VerifiableCredential"],
    "issuer": {
        "id": "did:didMethodName:address-Issuer"
    },
    "issuanceDate": "2022-18-07T15:10:24Z",
    "expirationDate": "2023-18-07T15:10:24Z",
    "credentialSubject": {
        "id": "did:didMethodName:address-Holder",
        "name": "John",
        "surname": "Doe",
        "company": "Company Name"
    },
    "proof": {
        "type": "RsaSignature2018",
        "created": "2022-18-07T15:10:24Z",
        "proofPurpose": "assertionMethod",
        "verificationMethod": "did:didMethodName:address-Issuer#keys-1",
        "jws": "signature of the Issuer"
    }
}
```

Classical Verifiable Presentation (VP)

```
VerifiablePresentation = {
    "@context": [
        "https://www.w3.org/2018/credentials/v1",
        "https://www.w3.org/2018/credentials/examples/v1"
    ],
    "type": ["VerifiablePresentation"],

    "VerifiableCredential": [{
        "@context": [
            "https://www.w3.org/2018/credentials/v1",
            "https://www.w3.org/2018/credentials/examples/v1"
        ],
        "id": "http://example/credentials/1",
        "type": ["VerifiableCredential"],
        "issuer": {
            "id": "did:didMethodName:address-Issuer"
        },
        "issuanceDate": "2022-18-07T15:10:24Z",
        "expirationDate": "2023-18-07T15:10:24Z",
        "credentialSubject": {
            "id": "did:didMethodName:address-Holder",
            "name": "John",
            "surname": "Doe",
            "company": "Company Name"
        },
        "proof": {
            "type": "RsaSignature2018",
            "created": "2022-18-07T15:10:24Z",
            "proofPurpose": "assertionMethod",
            "verificationMethod": "did:didMethodName:address-Issuer#keys-1",
            "jws": "signature of the Issuer"
        }
    }],

    "proof": {
        "type": "RsaSignature2018",
        "created": "2022-18-07T16:10:24Z",
        "proofPurpose": "assertionMethod",
        "verificationMethod": "did:didMethodName:address-Holder#keys-1",
        "jws": "signature of the Holder"
    }
}
```

Fig. 4. Example of Verifiable Credential (left) and Verifiable Presentation (right) as by the W3C Recommendation [24].

It must be noted that the Issuer can also publish and store on the ledger a data schema, that corresponds to a template for the credential and is useful to enforce a specific structure and data format for the VC (as specified in [24]). In this way the Verifier can use the `credentialSchema` property of the VC to

retrieve such data schema and, then, he can use it to verify if the structure and contents of a VC are well formed (*i.e.*, they conform to the published schema).

Issuers, in accordance with [24], are also responsible for VCs revocation for cryptographic integrity and for status change purposes. Thus, a Verifier, while checking for authenticity of VCs and VPs, must check the VC status and reject any request in case the VC is revoked. Issuers are instructed to announce revocations through a revocation list the Verifiers can access during verification process. The W3C Recommendation [24] urges Issuers to use mechanisms that mitigate potential privacy violations, for example by using a globally-unique identifier as the subject for any given VC and never re-use that VC.

On top of the first three layers, it is possible to build any ecosystem of trustable interactions among human beings and things.

3 The Reason Behind Zero-Knowledge Proof Adoption

Authorization and authentication are different but interrelated information security processes. In SSI frameworks, the authentication procedure starts at Layer 2 by proving control of a DID and it continues at Layer 3 where Verifiers check the authenticity and integrity of the VC. The only information persisting at both layers is the DID of the peer (*i.e.*, the `id` field) into the DID Document and into the `credentialSubject` field of the VC, as shown in Fig. 3 and Fig. 4. The other properties in the claim of the VC provide further (high level) information about the identity of the peer, and they are used by the Verifier to grant or to deny access to services and/or resources.

The simplest SSI implementation to conclude authentication at Layer 3 makes use of a DID value in the `credentialSubject` field of the VC. This choice works but it introduces privacy issues. Two Verifiers can collude, share their access logs, and trace/link any peer in the ecosystem. In principle, this bad practice may take place also at Layer 2. For this reason, the use of a different DID is encouraged for every peer relationship, but this precludes the use of DID into the VC to terminate authentication procedure at Layer 3.

This leads to the main point: the solution for a privacy-preserving VCs system is to leverage DIDs to build secure communication channel with server-side authentication only[3] and to proceed with client full authentication at Layer 3. In addition, it is necessary to adopt VCs that enable the Holder to manage their privacy by choosing the level of information disclosure. This objective is achieved by means of Anonymous and Selective Disclosure VCs. Anonymous VCs allow a peer to prove that their identity satisfies certain properties in an uncorrelated way without revealing any identity details. Selective Disclosure VCs allow a peer to select the identity properties to reveal on purpose to maintain the desired level of privacy.

[3] The main purpose of such a secure communication channel is server authentication, confidentiality, and integrity of application data exchanged by peers upon mutual-authentication and authorization completion.

The authorization procedure is based on disclosed claim properties in the VC or based on the possession of an Anonymous VC issued by a specific Issuer. The implementation of Anonymous and Selective Disclosure VCs is made possible by some cryptographic signature schemes compatible with a proper Zero-Knowledge Proof (ZKP) framework.

The following sections provide the mathematical foundations from existing literature to build such a VCs system compatible with SSI framework.

4 Classical Zero-Knowledge Verifiable Credentials

A VCs system is anonymous if it allows the Holder to demonstrate such credentials without revealing any additional information about their identity. In addition, it is required that the system allows the Holder to obtain the credentials anonymously.

For the latter requirement, the main solution adopted in the literature is to exploit a *commitment scheme*, which allows the Holder to communicate the secret properties in a credential to the Issuer without revealing them, together with a *blind signature scheme*, so that the Issuer can provide a digital signature on the commitment of the secret properties.

The first requirement is generally obtained through a protocol for the *ZKP of knowledge of a signature*, that any Verifier can adopt in order to check the validity of the digital signature and of the secret properties in the credential, which are known only by the Holder.

In the following subsections, two of the main zero-knowledge VCs schemes based on classical cryptographic problems are described. In particular, their security relies on the strong RSA assumption (that can be reduced to the Integer Factorization Problem, IFP) and on the strong Diffie–Hellman assumption (that can be reduced to the Discrete Logarithm Problem, DLP), respectively.

4.1 RSA-based Verifiable Credentials

In this section we describe one of the first ideas for a zero-knowledge VCs scheme, which was proposed by Camenisch and Lysyanskaya in 2003 [7]. They provided what they called *a signature scheme with efficient protocols*, where the efficient protocols were actually a commitment scheme with blind signature and a ZKP of knowledge of a signature. This solution is often referred as *CL signature scheme and protocols* and it is also included as a relevant example of ZKP scheme in the W3C VC data model [24].

Digital signature scheme. This is a basic digital signature scheme for a block of messages. This scheme is already compatible with a commitment scheme and ready to be adapted for achieving blindness.

Parameters: k and ℓ_m, which are integers representing the bit-length of the primes and the messages, respectively, and a security parameter ℓ.

Key Generation: the Issuer chooses:

- p, q safe primes[4] such that $n = pq$ is $\ell_n = 2k$ bits long;
- $a_1, \ldots, a_L, b, c \in \mathbb{Z}_n^\times$ quadratic residues;

return $\mathsf{pk}_I = (n, a_1, \ldots, a_L, b, c)$ and $\mathsf{sk}_I = p$.

Signing: given $m_1, \ldots, m_L$ of ℓ_m bits, pk_I and sk_I, choose:

- e prime of $\ell_e \geq \ell_m + 2$ bits, with $e > 2^{\ell_m+1}$;
- s integer $\ell_s = \ell_n + \ell_m + \ell$ bits long.

Compute $v \in \mathbb{N}$ such that $v^e \equiv \prod_{i=1}^{L} a_i^{m_i} b^s c \pmod{n}$ (this can be done efficiently because the signer knows p and q) and return the signature $\sigma = (e, s, v)$.

Verification: given $m_1, \ldots, m_L$, σ and pk, check that:

$$v^e \equiv \prod_{i=1}^{L} a_i^{m_i} b^s c \pmod{n},$$
$$2^{\ell_e - 1} < e < 2^{\ell_e}.$$

Security: the described scheme relies on the strong RSA assumption, which states that it is hard, on input n and $u \in \mathbb{Z}_n^\times$, to compute values $e > 1$ and v such that $v^e \equiv u \pmod{n}$.

Commitment scheme. The following protocol adopts the commitment scheme developed by Fujisaki and Okamoto [11] and further elaborated by Damgård and Fujisaki [9]. It is again based on the product of two large primes and on quadratic residues and it is generalized to work with a block of messages (*i.e.*, $m_1, \ldots, m_L$) instead of a single message.

Parameters: k and ℓ_m integers representing the bit-length of the primes and the messages, respectively.

Key Generation: after choosing:

- p_C, q_C safe primes such that $n_C = p_C q_C$ is $\ell_n = 2k$ bits long;
- $h_C \in \mathbb{Z}_n^\times$ quadratic residue;
- $g_{C_i} = {h_C}^{f_i} \pmod{n_C}$ for some random integer f_i, with i from 1 to L;

return $\mathsf{pk}_C = (n_C, g_{C_1}, \ldots, g_{C_L}, h_C)$.

Commitment: given $m_1, \ldots, m_L$ of ℓ_m bits and pk_C, take $r_C \in \mathbb{Z}_n$ random and return the commitment

$$C = \prod_{i=1}^{L} g_{C_i}^{m_i} {h_C}^{r_C} \pmod{n_C}.$$

[4] A *safe prime* is a prime number p such that $p = 2p' + 1$ for some prime number p'.

Blind signature scheme. The Holder queries the Issuer to send a valid signature on $m_1, \ldots, m_L$, using a commitment C, so that the messages remain unknown to the Issuer.

Inputs: both the parties share:

- $\mathsf{pk}_I = (n, a_1, \ldots, a_L, b, c)$ Issuer public key for the signature scheme;
- $\mathsf{pk}_C = (n_C, g_{C_1}, \ldots, g_{C_L}, h_C)$ commitment public key, where $n_C = p_C q_C$ is different from $n = pq$ in pk_I;
- C commitment to the messages m_i's;
- the parameters ℓ_m, ℓ_e, ℓ_s.

In addition, the Issuer consists in two separate software services:

1. a service for the generation of a digital identity (*i.e.* the claim properties in the VCs), capable to generate and to securely transmit to the Holder the messages m_i's of ℓ_m bits, the randomness $r_C \in \mathbb{Z}_n$, and the commitment $C = \prod_{i=1}^{L} g_{C_i}^{m_i} {h_C}^{r_C} \pmod{n_C}$;
2. a signature service, independent from the previous software logic and capable to blindly sign a commitment on secret messages.

In this way, both the Holder and the Issuer knows the commitment C. Only the Holder knows the messages m_i's and the randomness r_C. The signature service of the Issuer knows the factorization of $n = pq$, corresponding to the Issuer secret key $\mathsf{sk}_I = p$.

Protocol: the Holder queries the Issuer for a blind signature on the messages by:

- generating a new randomness $r \in \mathbb{Z}_n$;
- forming a new commitment $C_m = \prod_{i=1}^{L} a_i^{m_i} b^r \pmod{n}$;
- proving that C_m commits the same m_i's as C;
- proving knowledge of $m_1, \ldots, m_L$ and r;
- proving that $0 \leq m_i < 2^{\ell_m}$ and $0 \leq r < 2^{\ell_n}$.

Then, the Issuer signs the commitment C_m by:

- choosing a random prime e of ℓ_e bits;
- choosing a random integer r' of ℓ_s bits;
- computing the value $v = (C_m b^{r'} c)^{e^{-1} \pmod{(p-1)(q-1)}} \pmod{n}$.

Finally, the Holder evaluates $s = r + r'$ and outputs $\sigma = (e, s, v)$, that is a valid signature on $m_1, \ldots, m_L$.

ZKP of knowledge of a signature. The last required scheme allows the Holder in possession of a valid signature σ on messages m_i's to prove its validity to a Verifier. The Holder uses auxiliary commitments related to the values (e, s, v), so that the m_i's and σ remains unknown to the Verifier.

Inputs: both the parties share:

- $\mathsf{pk}_I = (n, a_1, \ldots, a_L, b, c)$ Issuer public key for the signature scheme;
- $\mathsf{pk}_V = (n, g, g_1, \ldots, g_L, h)$ Verifier public key, consisting in a new commitment public key with the same modulo $n = pq$ as in pk_I;
- the parameters ℓ_m, ℓ_e, ℓ_s.

In addition, the Holder knows the messages m_i's of ℓ_m bits, a new randomness $r_x \in \mathbb{Z}_n$, such that $C_x \equiv \prod_{i=1}^{L} g_i^{m_i} h^{r_x} \pmod{n}$, and $\sigma = (e, s, v)$ valid signature on the m_i's.

Protocol: the Holder must:

- choose randomly $w, r_w, r_e \in \mathbb{Z}_n$;
- compute $C_v = v g^w \pmod{n}$;
- compute $C_w = g^w h^{r_w} \pmod{n}$;
- compute $C_e = g^e h^{r_e} \pmod{n}$.

After sending the commitments C_x, C_v, C_w, C_e to the Verifier, the Holder carries out a ZKP of knowledge for $(m_1, \ldots, m_L, r_x, e, s, v, w, r_w, r_e)$ such that

$$
\begin{gathered}
c \equiv C_v^e \prod_{i=1}^{L} (1/a_i)^{m_i} (1/b)^s (1/g)^{ew} \pmod{n}, \\
1 \equiv C_w^e (1/g)^{ew} (1/h)^{er_w} \pmod{n}, \\
C_w \equiv g^w h^{r_w} \pmod{n}, \\
C_x \equiv \prod_{i=1}^{L} g_i^{m_i} h^{r_x} \pmod{n}, \\
C_e \equiv g^e h^{r_e} \pmod{n}, \\
2^{\ell_e - 1} < e < 2^{\ell_e}, \quad 0 \leq m_i < 2^{\ell_m}.
\end{gathered}
$$

Requirements: the blind signature scheme and the ZKP of knowledge of a signature use the following protocols which are secure under the strong RSA assumption [7]:

- ZKP of knowledge of discrete logarithm representation modulo a composite: given $n = pq$ and $g_1, \ldots, g_L, C \in \mathbb{Z}_n^\times$ quadratic residues, prove the knowledge of $m_1, \ldots, m_L$ such that
$$C \equiv \prod_{i=1}^{L} g_i^{m_i} \pmod{n};$$
- ZKP of knowledge of equality of representation modulo two different composite moduli: given two commitment keys $(n_1, g_1, \ldots, g_L)$ and $(n_2, h_1, \ldots, h_L)$, where $n_1 = p_1 q_1$ and $n_2 = p_2 q_2$, and $C_1, C_2 \in \mathbb{Z}_n^\times$ quadratic residues, prove the knowledge of $m_1, \ldots, m_L$ such that
$$C_1 \equiv \prod_{i=1}^{L} g_i^{m_i} \pmod{n_1} \qquad \text{and} \qquad C_2 \equiv \prod_{i=1}^{L} h_i^{m_i} \pmod{n_2};$$
- ZKP that a committed value lies in a given integer interval: given a commitment key (n, g, h), $C \in \mathbb{Z}_n^\times$ quadratic residue and two integers a, b, prove the knowledge of m, r such that
$$C \equiv g^m h^r \pmod{n}, \quad a \leq m \leq b.$$

Security parameters. For sake of clarity, the parameters considered in previous protocols can be listed as follows (with some examples for 128 bits of security):

- k length of the random primes p, q, p_C, q_C (*e.g.* $k = 1536$ bits);
- ℓ_n length of the special RSA moduli n and n_C (*e.g.* $\ell_n = 2k = 3072$ bits);
- ℓ_m length of each message m_i, with i from 1 to L (*e.g.* $\ell_m = 256$ bits is appropriate if each m_i is obtained as the digest of a secret claim in the VC, computed with a SHA-256 function, instead of SHA-1 adopted in [7]);
- ℓ additional security parameter (*e.g.* $\ell = \ell_m = 256$ bits);
- ℓ_e length of the prime e, part of the signature (*e.g.* $\ell_e = \ell_m + 2 = 258$ bits);
- ℓ_s length of the integer s, part of the signature (*e.g.* $\ell_s = \ell_n + \ell_m + \ell = 3584$ bits).

Achieving selective disclosure. The introduced scheme allows to obtain directly Anonymous VCs. In addition, since the scheme produces a single signature for a whole block of messages, it is easy to obtain Selective Disclosure VCs by simply allowing the Holder to share with the Verifier some of the claims in the credential.

Specifically, the previous protocol between the Holder and the Issuer (*i.e.* the blind signature scheme) remains unchanged, while the second protocol between the Holder and the Verifier (*i.e.* the ZKP of knowledge of a signature) can be easily adapted as follows. After sending the commitments C_x, C_v, C_w, C_e to the Verifier, the Holder:

- arbitrarily chooses a subset D from the full set of secret messages $m_1, \ldots, m_L$;
- selectively discloses $\{m_i\}_{i \in D}$, with $D \subset \{1, \ldots, L\}$, to the Verifier;
- carries out the ZKP of knowledge on:
  - the set of undisclosed messages $\{m_j\}_{j \in U}$ with $U = \{1, ..., L\} \setminus D$,
  - the signature $\sigma = (e, s, v)$,
  - the randomnesses r_x, w, r_w, and r_e.

4.2 DH-based Verifiable Credentials

One of the main drawbacks of the schemes in Section 4.1 are the multiple range constraints to be proven in the ZKP: they need to be separately executed for each undisclosed message, resulting in an heavy computational burden in the case of a VC with several claims/attributes.

An alternative and efficient scheme based on DH instead of RSA can be obtained by adapting the work of Boneh, Boyen and Shacham [3] as suggested by Camenisch and Lysyanskaya in the same year [8]. The resulting scheme is publicly known as BBS+ [2,5] and is one of the most promising solutions in the context of ZKP VCs. This scheme is also recognized as a ZKP scheme compatible to the W3C VC data model [24], in addition to the RSA-based scheme described in previous section.

In the following, firstly two new schemes related to DH taken from the former paper are introduced. Then, the results from the latter reference concerning bilinear maps and the LRSW assumption [15] are described. Finally the schemes composing BBS+ are obtained.

ZKP of knowledge for strong Diffie–Hellman. The first novelty introduced by Boneh, Boyen and Shacham [3] is a ZKP for proving the possession of a solution to an instance of the strong Diffie–Hellman problem.

Inputs: the public data are:

- $h, u, v, \in G_1$ cyclic group generated by a public g_1;
- $w \in G_2$ cyclic group generated by a public g_2;
- $\psi : G_2 \to G_1$ group isomorphism with $\psi(g_2) = g_1$;
- $e : G_1 \times G_2 \to G_T$ bilinear map.

The prover knows $A, k \in G_1$ and $x, y, \gamma \in \mathbb{Z}_p$ such that $A^{x+\gamma} k^y = g_1$ and $w = g_2^{\gamma}$, so that $e(Ak^{y/(\gamma+x)}, wg_2^x) = e(g_1, g_2)$.

Protocol: the prover generates:

- $\alpha, \beta \in \mathbb{Z}_p$ randomly;
- $T_1 = u^{\alpha}$, $T_2 = v^{\beta}$, $T_3 = Ak^{y/(\gamma+x)} h^{\alpha+\beta} \in G_1$;
- $\delta_1 = x\alpha$, $\delta_2 = x\beta \in \mathbb{Z}_p$,

and proves to the verifier the knowledge of the values $(x, \alpha, \beta, \delta_1, \delta_2)$ such that

$$u^{\alpha} = T_1, \quad v^{\beta} = T_2,$$
$$e(T_3, g_2)^x \cdot e(h, w)^{-\alpha-\beta} \cdot e(h, g_2)^{-\delta_1-\delta_2} = e(g_1, g_2)/e(T_3, w),$$
$$T_1^x u^{-\delta_1} = 1, \quad T_2^x v^{-\delta_2} = 1.$$

In particular, the prover takes randomly $r_x, r_\alpha, r_\beta, r_{\delta_1}, r_{\delta_2} \in \mathbb{Z}_p$ and computes:

- $R_1 = u^{r_\alpha}$, $R_2 = v^{r_\beta} \in G_1$;
- $R_3 = e(T_3, g_2)^{r_x} \cdot e(h, w)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\delta_1} - r_{\delta_2}} \in G_T$;
- $R_4 = T_1^{r_x} u^{-r_{\delta_1}}$, $R_5 = T_2^{x} v^{-r_{\delta_2}}$,

and sends the values $(T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$ to the verifier, which replies with a random challenge $c \in \mathbb{Z}_p$. Now, the prover sends back $s_i = r_i + ci$ for $i \in \{x, \alpha, \beta, \delta_1, \delta_2\}$, so that the verifier can check if

$$u^{s_\alpha} = T_1^c R_1, \quad v^{s_\beta} = T_2^c R_2,$$
$$e(T_3, g_2)^{s_x} \cdot e(h, w)^{-s_\alpha - s_\beta} \cdot e(h, g_2)^{-s_{\delta_1} - s_{\delta_2}} = \big(e(g_1, g_2)/e(T_3, w)\big)^c R_3,$$
$$T_1^{s_x} u^{-s_{\delta_1}} = R_4, \quad T_2^{s_x} v^{-s_{\delta_2}} = R_5.$$

Short group signature. The following is a group signature that requires a manager for the generation of the common parameters and allows each user of the group to anonymously sign a message on behalf of the group. The security is based on the strong DH problem and is compatible with the previous ZKP.

Parameters: k bit-length of the prime p.

Key Generation: given G_1, G_2 cyclic groups of prime order p, ψ and e as before, the manager chooses:

- the generators $g_2 \in G_2$ and $g_1 = \psi(g_2) \in G_1$;
- $h \in G_1$ and $\xi_1, \xi_2, \gamma \in \mathbb{Z}_p^\times$ randomly.

After setting $u, v \in G_1$ such that $u^{\xi_1} = v^{\xi_2} = h$ and $w = g_2^\gamma$, the group public key is $\mathsf{gpk} = (g_1, g_2, h, u, v, w)$ and the manager secret key is $\mathsf{msk} = (\xi_1, \xi_2, \gamma)$. In order to generate the user key, $k \in G_1$ is published so that the user chooses $y \in \mathbb{Z}_p$ and sends k^y to the manager, which takes $A \in G_1$ such that $A^{\gamma+x} k^y = g_1$ using a random $x \in \mathbb{Z}_p^\times$ and sends (A, x) to the user. Finally, the user has the secret key $\mathsf{usk} = (A, x, y)$ such that $A^{\gamma+x} k^y = g_1$.

Signing: given a message m, the public key gpk and the user secret key usk, compute:

- $T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5$ as in the previous protocol;
- the challenge $c = \mathrm{hash}(T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5) \in \mathbb{Z}_p$;
- $s_x, s_\alpha, s_\beta, s_{\delta_1}, s_{\delta_2}$ as in the previous protocol.

Output the signature $\sigma = (T_1, T_2, T_3, c, s_x, s_\alpha, s_\beta, s_{\delta_1}, s_{\delta_2})$.

Verification: given m, σ and gpk, evaluate:

$$\tilde{R}_1 = u^{s_\alpha} T_1^{-c}, \quad \tilde{R}_2 = v^{s_\beta} T_2^{-c},$$
$$\tilde{R}_3 = e(T_3, g_2)^{s_x} \cdot e(h, w)^{-s_\alpha - s_\beta} \cdot e(h, g_2)^{-s_{\delta_1} - s_{\delta_2}} \cdot \big(e(T_3, w)/e(g_1, g_2)\big)^c,$$
$$\tilde{R}_4 = T_1^{s_x} u^{-s_{\delta_1}}, \quad \tilde{R}_5 = T_2^{s_x} v^{-s_{\delta_2}},$$

and check if

$$c = \mathrm{hash}(T_1, T_2, T_3, \tilde{R}_1, \tilde{R}_2, \tilde{R}_3, \tilde{R}_4, \tilde{R}_5).$$

Security: the scheme relies on the assumption that the strong Diffie–Hellman problem is hard. Specifically, the statement is that, given G_1, G_2 of order p and the tuple $(g_1, g_2, g_2^\gamma, \ldots, g_2^{\gamma^q})$, is hard to find a pair $\big(g_1^{1/(\gamma+x)}, x\big)$ where $x \in \mathbb{Z}_p^\times$.

Digital signature scheme from LRSW. Camenischand Lysyanskaya [8] constructed a digital signature based on the assumption from a previous work of Lysyanskaya, Rivest, Sahai, and Wolf [15].

Parameters: k bit-length of the prime p.

Key Generation: given the cyclic groups G, G_T of prime order p with public generators g, g_T, respectively, and a bilinear, non-degenerate and efficient map $e : G \times G \to G_T$, then:

- sk contains $x, y, z \in \mathbb{Z}_p$;
- pk contains $X = g^x, Y = g^y, Z = g^z, W = Y^z \in G$.

Signing: given (m, r), pk and sk, obtain:

- $A = a^z$ for $a \in G$;
- $b = a^y$ and $B = A^y$;
- $c = a^{x+xym} A^{xyr}$.

The output is the signature $\sigma = (a, A, b, B, c)$.

Verification: given (m, r), σ and pk, check that:

$$e(a, Z) = e(g, A), \quad e(a, Y) = e(g, b), \quad e(A, Y) = e(g, B),$$
$$e(X, a) \cdot e(X, b)^m \cdot e(X, B)^r = e(g, c).$$

Security: the described scheme relies on the assumption that the LRSW problem is hard. Specifically, given G generated by g, $X = g^x$, $Y = g^y$ and the triplet (a, a^y, a^{x+mxy}) for some $a \in G$, the problem consists in finding the value of $m \in \mathbb{Z}_q$ satisfying the given conditions.

Blind signature scheme from LRSW. The second scheme taken from [8] is an adaptation of the previous one that allows to exploit the LRSW assumption to sign a committed message in a context of VCs between an Holder and an Issuer.

Inputs: both the parties share:

- $\mathsf{pk} = (p, g, G, g_T, G_T, e, X, Y, Z, W)$ for signature and commitment;
- $M = g^m Z^r$ commitment for (m, r).

Only the Holder knows the committed (m, r) while only the Issuer knows the secret key $\mathsf{sk} = (x, y, z)$ for the signature.

Protocol: firstly the Holder gives a ZKP of knowledge of the committed (m, r). Then the Issuer computes:

- $A = a^z$ for $a = g^\alpha$ with $\alpha \in \mathbb{Z}_p$;
- $b = a^y = Y^\alpha$ and $B = A^y = W^\alpha$;
- $c = a^x M^{\alpha xy} = a^{x+xym} A^{xyr}$.

The output is the signature $\sigma = (a, A, b, B, c)$.

ZKP of knowledge of a signature from LRSW. The last scheme required for the construction of Anonymous VCs allows the Holder in possession of a valid signature σ on the committed message (m, r) to prove its validity to a Verifier, without revealing σ.

Inputs: both the parties share:

- $\mathsf{pk} = (p, g, G, g_T, G_T, e, X, Y, Z, W)$ for signature and commitment;
- $M = g^m Z^r$ commitment for (m, r).

The Holder knows the committed (m, r) and a valid signature $\sigma = (a, A, b, B, c)$.

Protocol: the Holder obtains a blinded signature by:

- choosing randomly $s, s' \in \mathbb{Z}_p$;
- computing $\tilde{a} = a^{s'}$ and $\tilde{A} = A^{s'}$;
- computing $\tilde{b} = b^{s'}$ and $\tilde{B} = B^{s'}$;
- computing $\tilde{c} = c^{s'/s}$.

Thus, the Holder sends $\tilde{\sigma} = (\tilde{a}, \tilde{A}, \tilde{b}, \tilde{B}, \tilde{c})$ to the Verifier, and they carry out a ZKP of knowledge for (m, r, s) such that

$$e(g, \tilde{c})^s = e(X, \tilde{a}) \cdot e(X, \tilde{b})^m \cdot e(X, \tilde{B})^r.$$

The Verifier accepts if the proof works and

$$e(\tilde{a}, Z) = e(g, \tilde{A}), \quad e(\tilde{a}, Y) = e(g, \tilde{b}), \quad e(\tilde{A}, Y) = e(g, \tilde{B}).$$

BBS+ digital signature. Finally this section describes the BBS+ signature firstly introduced by Camenischand Lysyanskaya [8], as a suggestion of merging their results with the previous work of Boneh, Boyen, and Shacham [3] (see also [2,5]).

Parameters: k bit-length of the prime p.

Key Generation: given G_1, G_2 of prime order p, $\psi : G_2 \to G_1$, h generator of G_2, $g = \psi(h), g_0, g_1, \ldots, g_L$ generators of G_1, and $e : G_1 \times G_2 \to G_T$, then:

- take as secret key $\gamma \in \mathbb{Z}_p^\times$;
- evaluate the public key $w = h^\gamma$.

Signing: given $m_1, \ldots, m_L \in \mathbb{Z}_p^\times$, γ and w, then:

- choose random $e, s \in \mathbb{Z}_p^\times$;
- compute $A = (g\, g_0^s\, g_1^{m_1} \cdots g_L^{m_L})^{1/(\gamma+e)}$.

The signature is $\sigma = (A, e, s)$.

Verification: given $m_1, \ldots, m_L$, σ and w, check that

$$e(A, wh^e) = e(g\, g_0^s\, g_1^{m_1} \cdots g_L^{m_L}, h).$$

Security: as for the group signature from [3] the scheme relies on the strong Diffie–Hellman assumption. In order to increase the security level, the best choice for the cyclic groups are clearly Elliptic Curves (EC), so that the hard problem on which the scheme relies is ECDH.

Blind signature from BBS+. As for the schemes based on the LRSW assumption, it is possible to adapt the previous signature to be blind.

Inputs: both the Holder and the Issuer share:

- $\mathsf{pk} = (p, g, g_0, g_1, \ldots, g_L, h, w)$ for signature and commitment;
- $C = g_0^r\, g_1^{m_1} \cdots g_L^{m_L}$ commitment for $m_1, \ldots, m_L$ with $r \in \mathbb{Z}_p^\times$ random.

Only the Holder knows the committed $(m_1, \ldots, m_L, r)$ while only the Issuer knows γ.

Protocol: in order to obtain a signature on the committed messages:

- the Holder gives a ZKP of knowledge of the committed $(m_1, \ldots, m_L, r)$;
- the Issuer chooses $e, r' \in \mathbb{Z}_p^\times$, computes $A = (g g_0^{r'} C)^{1/(\gamma+e)}$ and sends (A, e, r');
- the Holder computes $s = r + r'$ so that $\sigma = (A, e, s)$ is valid.

ZKP of knowledge of a signature from BBS+. Finally, the last required scheme is the ZKP of the possession of a valid blind signature on a committed message.

Inputs: both the Holder and the Verifier share:

- $\mathsf{pk} = (p, g, g_0, g_1, \ldots, g_L, h, w)$ for signature and commitment;
- $C = g_0^r\, g_1^{m_1} \cdots g_L^{m_L}$ commitment for $m_1, \ldots, m_L$ with $r \in \mathbb{Z}_p^\times$ random.

Only the Holder knows the committed $(m_1, \ldots, m_L, r)$ and a valid signature $\sigma = (A, e, s)$.

Protocol: the Holder obtains a blinded signature by:

- choosing randomly $r_0, r_1 \in \mathbb{Z}_p^\times$;
- computing $A_0 = g_0^{r_0}\, g_1^{r_1}$ and $A_1 = A\, g_1^{r_0}$.

Thus, the Holder sends A_0, A_1 to the Verifier, and they carry out a ZKP of knowledge for the values $(r_0, r_1, \delta_0, \delta_1, m_1, \ldots, m_L, e, s)$ such that

$$A_0 = g_0^{r_0}\, g_1^{r_1}, \quad A_0^e = g_0^{\delta_0}\, g_1^{\delta_1},$$
$$\frac{e(A_1, w)}{e(g, h)} = \frac{e(g_1, w)^{r_0} \cdot e(g_1, h)^{\delta_0} \cdot e(g_0, h)^s \cdot e(g_1, h)^{m_1} \cdots e(g_L, h)^{m_L}}{e(A_1, h)^e}.$$

Security parameters. For sake of clarity, BBS+ adopts the pairing friendly curve BLS12-381 where $k = |p| = 381$, so that:

- $|\mathsf{pk}| = k(L + 5)$;
- $|C| = 381$;
- $|\sigma| = 3 \cdot 381 = 1143$.

It must be noted that BLS12-381 was intended to offer a 128-bit security level. However, a recent report [18] states that this curve achieves an actual security level between 117 and 120 bits at most.

Achieving selective disclosure. As for the scheme based on RSA, the introduced scheme allows to obtain directly Anonymous VCs. In addition, since the scheme produces a single signature for a whole block of messages, it is easy to obtain Selective Disclosure VCs by simply allowing the Holder to share with the Verifier some of their credentials.

5 Quantum-Resistant Zero-Knowledge Verifiable Credentials

The security of the mentioned schemes relies on the assumption that the classical RSA and (EC)DH are hard to solve. Specifically, these problems can be reduced to the IFP and the DLP, respectively. Actually, with standard computers, the solving algorithms have exponential time. In particular, the best algorithm for:

- solving the IFP, *i.e.*, finding the prime factors of n, is the general number field sieve, which has runtime exponential in $\sqrt[3]{\ell_n}$, where $\ell_n = \lceil \log_2 n \rceil$;
- solving the DLP is the index calculus, which has runtime exponential in $\sqrt[3]{p}$, where p is the order of the adopted cyclic group.

However, the idea of quantum computers that was introduced by Feynman [10] and others in the 1960s-80s, brought to new powerful algorithms. In particular, the Grover algorithm [12] that allows to obtain a quadratic speed-up on brute-force attacks and, more importantly, the Shor algorithm [21] that allows to solve the IFP and the DLP in polynomial runtime.

Until quantum computers were only a theoretical concept, modern cryptography was not in danger, but in 2000s the first working quantum computers were constructed and the cryptographic community started to worry about the risks. Today, the largest working quantum computers have about 60 qubits and are not yet a real threat to cryptographic schemes, but the NIST has already selected the standard Post-Quantum (PQ) schemes [1].

A secondary challenge, in time and absolutely not in importance, is the migration to quantum resistant schemes. Since SSI relies on VCs that are based on classical assumptions like the hardness of IFP and DLP, new post-quantum alternatives for the described schemes are required.

In the following, a proposal introduced by Jeudy, Roux-Langlois, and Sanders [13] resulting from the European project PROMETHEUS [20] is described.

5.1 Lattice-based Verifiable Credentials

The schemes proposed by Jeudy, Roux-Langlois, and Sanders [13] rely on the hardness of the Short Integer Solution (SIS) problem and of the Learning With Errors (LWE) problem, which are both related to lattices. This family of cryptosystems is one of the most reliable among the PQ alternatives and has the smallest data-size and best performance. Moreover, two of the three digital signature schemes (as well as the only Key Encapsulation Mechanism) selected

by the NIST PQ standardization process are based on those problems on lattices. The authors kept the formulation of the following schemes generic, so that different settings can be adopted depending on the wanted characteristics.

Digital signature scheme with lattices. As in the previous section, the starting point is a digital signature compatible with a ZKP.

Parameters: q prime, m, n, s_1, s_2 positive integers, $\sigma, \sigma_2 \in \mathbb{R}$ and $\sigma_1 = \sqrt{\sigma^2 + \sigma_2^2}$ sampling widths, $\mathbf{g} = [2^0, \ldots, 2^{\lceil \log_2 q \rceil - 1}] \in \mathbb{Z}_q^{1 \times \lceil \log_2 q \rceil}$ and a random $\mathbf{D} \in \mathbb{Z}_q^{n \times m}$.

Key Generation: knowing the parameters above:

- pick randomly $\mathbf{A} \in \mathbb{Z}_q^{n \times s_1}$;
- pick randomly $\mathbf{R} \in \{-1, 0, 1\}^{s_1 \times s_2}$;
- evaluate $\mathbf{B} = \mathbf{AR} \pmod q \in \mathbb{Z}_q^{n \times s_2}$;
- pick randomly $\mathbf{u} \in \mathbb{Z}_q^{n \times s_1}$.

Return $\mathsf{pk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$ and $\mathsf{sk} = \mathbf{R}$.

Signing: given $\mathbf{m} \in \{0,1\}^m$, pk and sk, then:

- pick $\mathbf{r}$ from a discrete Gaussian distribution $\mathcal{D}_\sigma(\mathbb{Z}^{s_1})$;
- compute the commitment $\mathbf{c} = (\mathbf{Ar} + \mathbf{Dm}) \pmod q$;
- pick randomly $\tau \in \mathbb{Z}_{q'} \setminus \{0\}$ for $q' < q$;
- obtain $\mathbf{v} \in \mathbb{Z}^{s_1+s_2}$ such that $\mathbf{v} +^t [{}^t\mathbf{r} \,|\, \mathbf{0}_{s_2}] = \mathtt{SampleD}(\mathbf{R}, \mathbf{A}, \tau\mathbf{I}_n, \mathbf{u} + \mathbf{c}, \sigma_2)$, which outputs an array $\mathbf{v}'$ that is statistically close to $\mathcal{D}_{\sigma_2}(\mathbb{Z}^{s_1+s_2})$ and such that $[\mathbf{A} \,|\, \tau(\mathbf{I}_n \otimes \mathbf{g}) - \mathbf{B}]\mathbf{v}' \equiv (\mathbf{u} + \mathbf{c}) \pmod q$.

The signature is $\mathsf{sign} = (\tau, \mathbf{v})$.

Verification: given $\mathbf{m} \in \{0,1\}^m$, pk and sign, then:

- evaluate $\mathbf{A}_\tau = [\mathbf{A} \,|\, \tau(\mathbf{I}_n \otimes \mathbf{g}) - \mathbf{B}] \in \mathbb{Z}_q^{n \times (s_1+s_2)}$;
- split $\mathbf{v} = {}^t[{}^t\mathbf{v}_1 \,|\, {}^t\mathbf{v}_2]$ with $\mathbf{v}_1 \in \mathbb{Z}^{s_1}, \mathbf{v}_2 \in \mathbb{Z}^{s_2}$;
- check that

$$\mathbf{A}_\tau \mathbf{v} \equiv (\mathbf{u} + \mathbf{Dm}) \pmod q,$$
$$\|\mathbf{v}_1\|_\infty \leq \sigma_1 \log_2 s_1, \quad \|\mathbf{v}_2\|_\infty \leq \sigma_2 \log_2 s_2, \quad \tau \in \mathbb{Z}_{q'} \setminus \{0\}.$$

Security: the described scheme relies on the SIS problem. Specifically, given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and the lattice $\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m \,|\, \mathbf{Ae} \equiv \mathbf{0} \pmod q\}$, the problem consists in finding an array $\mathbf{x} \in \Lambda_q^\perp(\mathbf{A})$ such that $\|\mathbf{v}\|_\infty$ and $\|\mathbf{x}\|_2$ are small.

Blind signature scheme with lattices. The first privacy-preserving scheme proposed [13] is an oblivious signing protocol, *i.e.* a blind signature.

Inputs: both the Holder and the Issuer share:

- the parameters required for the signature scheme, slightly modified as follows. After taking two widths $\sigma', \sigma'' \in \mathbb{R}$, evaluate $\sigma = \sqrt{\sigma'^2 + \sigma''^2}$ and, as before, $\sigma_1 = \sqrt{\sigma^2 + \sigma_2^2}$;
- $\mathsf{pk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$ for signature and commitment, with $\mathbf{A}$ obtained as hash of a public value.

Only the Holder knows the message $\mathbf{m} \in \{0,1\}^m$ to be committed, while only the Issuer knows the secret key $\mathbf{R}$.

Protocol: firstly, the holder creates a commitment of $\mathbf{m}$ by

- picking $\mathbf{r}'$ from a discrete Gaussian distribution $\mathcal{D}_{\sigma'}(\mathbb{Z}^{s_1})$;
- computing the commitment $\mathbf{c} = (\mathbf{A}\mathbf{r}' + \mathbf{D}\mathbf{m}) \pmod q$.

Thus, $\mathbf{c}$ is sent to the Issuer and the parties carry out a ZKP of knowledge for the committed $\mathbf{m}$. This is achieved by transforming the relation in one fitting the framework from [25]. After defining $\alpha = \lceil \sigma' \log_2 s_1 \rceil$ and $\mathbf{a} = \alpha \mathbf{1}_{s_1}$ so that

$$\mathbf{r}' \in [-\alpha, \alpha]^{s_1} \quad \text{and} \quad \mathbf{r}' + \mathbf{a} \in [0, 2\alpha]^{s_1},$$

let $k = \lfloor \sigma' \log_2 2\alpha \rfloor + 1$ and

$$\mathbf{g} = \big[\lfloor (2\alpha+1)/2 \rfloor, \lfloor (2\alpha+2)/2^2 \rfloor, \ldots, \lfloor (2\alpha+2^{k-1})/2^k \rfloor\big].$$

When denoting by $\overline{\mathbf{r}'} \in \{0,1\}^{s_1 k}$ the binary decomposition of $\mathbf{r}' + \mathbf{a}$ along $\mathbf{g}$, *i.e.* such that $(\mathbf{I}_{s_1} \otimes \mathbf{g})\overline{\mathbf{r}'} = \mathbf{r}' + \mathbf{a}$, the conditions to be proven become

$$\mathbf{A}(\mathbf{I}_{s_1} \otimes \mathbf{g})\overline{\mathbf{r}'} + \mathbf{D}\mathbf{m} \equiv \mathbf{c} + \mathbf{A}\mathbf{a} \pmod q, \quad \overline{\mathbf{r}'} \in \{0,1\}^{s_1 k}, \quad \mathbf{m} \in \{0,1\}^m,$$

that can be formulated as $\overline{\mathbf{A}}\mathbf{x} = \mathbf{y}$, where

$$\overline{\mathbf{A}} = \big[\mathbf{A}(\mathbf{I}_{s_1} \otimes \mathbf{g}) \,|\, \mathbf{D}\big], \quad \mathbf{x} = \begin{bmatrix} \overline{\mathbf{r}'} \\ \mathbf{m} \end{bmatrix}, \quad \mathbf{y} = \mathbf{c} + \mathbf{A}\mathbf{a},$$

with constraints for the binary coefficients given by

$$\mathcal{M} = \big\{(i,i,i) \,|\, i \in [1, s_1 k + m]\big\}.$$

If satisfied, the Issuer can:

- pick $\mathbf{r}''$ from a discrete Gaussian distribution $\mathcal{D}_{\sigma''}(\mathbb{Z}^{s_1})$;
- compute $\mathbf{c}' = (\mathbf{A}\mathbf{r}'' + \mathbf{c}) \pmod q$;
- pick randomly $\tau \in \mathbb{Z}_{q'} \setminus \{0\}$ for $q' < q$;
- obtain $\mathbf{v}' = \mathtt{SampleD}(\mathbf{R}, \mathbf{A}, \tau\mathbf{I}_n, \mathbf{u} + \mathbf{c}', \sigma_2) - {}^t[{}^t\mathbf{r}'' \,|\, \mathbf{0}_{s_2}] \in \mathbb{Z}^{s_1+s_2}$;
- send $(\tau, \mathbf{v}')$.

Now the Holder obtains $\mathbf{v} = \mathbf{v}' - {}^t[{}^t\mathbf{r}' \,|\, \mathbf{0}_{s_2}]$ and, if the signature $(\tau, \mathbf{v})$ is valid on $\mathbf{m}$, then this is the obtained blind signature.

ZKP of knowledge of a signature with lattices. Finally, the last required scheme is the ZKP of the possession of a valid blind signature on a committed message, which is called Prove [13].

Inputs: both the Holder and the Verifier know:

- the parameters required for the blind signature scheme;
- $\mathsf{pk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$ for blind signature and commitment.

Only the Holder knows the message $\mathbf{m} \in \{0,1\}^m$ and a valid blind signature $(\tau, \mathbf{v}) \in \mathbb{Z}_q \times \mathbb{Z}^{s_1+s_2}$.

Protocol: the Holder proves to the Verifier that $\mathbf{m}$ and $(\tau, \mathbf{v})$ satisfy

$$\mathbf{A}\mathbf{v}_1 - \mathbf{B}\mathbf{v}_2 + \tau(\mathbf{I}_n \otimes \mathbf{g})\mathbf{v}_2 - \mathbf{D}\mathbf{m} \equiv \mathbf{u} \pmod{q},$$

where $\mathbf{v} = {}^t[{}^t\mathbf{v}_1 \,|\, {}^t\mathbf{v}_2]$ such that $\mathbf{v}_1 \in \mathbb{Z}^{s_1}$ and $\mathbf{v}_2 \in \mathbb{Z}^{s_2}$, with

$$\|\mathbf{v}_1\|_\infty \leq \sigma_1 \log_2 s_1, \quad \|\mathbf{v}_2\|_\infty \leq \sigma_2 \log_2 s_2, \quad \tau \in \mathbb{Z}_{q'} \setminus \{0\}, \quad \mathbf{m} \in \{0,1\}^m.$$

Again, this can be transformed into a relation fitting the framework of Yang et al. [25]. Firstly, define

$$\begin{gathered}
\alpha_1 = \lceil \sigma_1 \log_2 s_1 \rceil, \quad \alpha_2 = \sigma_2 \log_2 s_2, \\
k_1 = \lfloor \log_2 2\alpha_1 \rfloor + 1, \quad k_2 = \lfloor \log_2 2\alpha_2 \rfloor + 1, \quad k' = \lfloor \log_2 q' \rfloor + 1, \\
\mathbf{g}_1 = \left[\lfloor (2\alpha_1+1)/2 \rfloor, \lfloor (2\alpha_1+2)/2^2 \rfloor, \ldots, \lfloor (2\alpha_1+2^{k_1-1})/2^{k_1} \rfloor\right], \\
\mathbf{g}_2 = \left[\lfloor (2\alpha_2+1)/2 \rfloor, \lfloor (2\alpha_2+2)/2^2 \rfloor, \ldots, \lfloor (2\alpha_2+2^{k_1-1})/2^{k_1} \rfloor\right], \\
\mathbf{g}' = \left[\lfloor (q'+1)/2 \rfloor, \lfloor (q'+2)/2^2 \rfloor, \ldots, \lfloor (q'+2^{k_1-1})/2^{k_1} \rfloor\right].
\end{gathered}$$

In addition, set $\mathbf{a}_1 = \alpha_1 \mathbf{1}_{s_1}$ and $\mathbf{a}_2 = \alpha_2 \mathbf{1}_{s_2}$ that allow to define $\mathbf{v}'_1 = \mathbf{v}_1 + \mathbf{a}_1$ and $\mathbf{v}'_2 = \mathbf{v}_2 + \mathbf{a}_2$.

Then, denote by $\overline{\mathbf{v}}_1, \overline{\mathbf{v}}_2$ the binary decomposition of $\mathbf{v}_1, \mathbf{v}_2$ along $\mathbf{g}_1, \mathbf{g}_2$, *i.e.* such that $(\mathbf{I}_{s_1} \otimes \mathbf{g}_1)\overline{\mathbf{v}}_1 = \mathbf{v}'_1$ and $(\mathbf{I}_{s_2} \otimes \mathbf{g}_2)\overline{\mathbf{v}}_2 = \mathbf{v}'_2$, respectively. Analogously, let $\overline{\tau}$ be the binary decomposition of τ along $\mathbf{g}'$, *i.e.* such that $\mathbf{g}'\overline{\tau} = \tau$. When considering $\mathbf{u}_2 = (\mathbf{I}_n \otimes \mathbf{g})\mathbf{v}_2$ and $\mathbf{u}'_2 = \tau\mathbf{u}_2$, there is additional linear relation but fewer decompositions, so that the Holder has to prove

$$\begin{cases}
\mathbf{A}(\mathbf{I}_{s_1} \otimes \mathbf{g}_1)\overline{\mathbf{v}}_1 - \mathbf{B}(\mathbf{I}_{s_2} \otimes \mathbf{g}_2)\overline{\mathbf{v}}_2 + \mathbf{u}'_2 - \mathbf{D}\mathbf{m} \equiv \mathbf{u} + \mathbf{A}\mathbf{a}_1 - \mathbf{B}\mathbf{a}_2 \pmod{q}, \\
(\mathbf{I}_n \otimes \mathbf{g})(\mathbf{I}_{s_2} \otimes \mathbf{g}_2)\overline{\mathbf{v}}_2 - \mathbf{u}_2 \equiv (\mathbf{I}_n \otimes \mathbf{g})\mathbf{a} \pmod{q}, \\
-\tau + \mathbf{g}'\overline{\tau} \equiv 0 \pmod{q},
\end{cases}$$

that can be formulated as $\overline{\mathbf{A}}\mathbf{x} = \mathbf{y}$, where

$$\overline{\mathbf{A}} = \begin{bmatrix} \mathbf{0}_{n\times 1} & \mathbf{0}_{n\times k'} & \mathbf{A}(\mathbf{I}_{s_1} \otimes \mathbf{g}_1) & -\mathbf{B}(\mathbf{I}_{s_2} \otimes \mathbf{g}_2) & -\mathbf{D} & \mathbf{0}_{n\times n} & \mathbf{I}_n \\ \mathbf{0}_{n\times 1} & \mathbf{0}_{n\times k'} & \mathbf{0}_{n\times s_1 k_1} & -(\mathbf{I}_n \otimes \mathbf{g})(\mathbf{I}_{s_2} \otimes \mathbf{g}_2) & \mathbf{0}_{n\times m} & -\mathbf{I}_n & \mathbf{0}_{n\times n} \\ -1 & \mathbf{g}' & \mathbf{0}_{1\times s_1 k_1} & \mathbf{0}_{1\times s_2 k_2} & \mathbf{0}_{1\times m} & \mathbf{0}_{1\times n} & \mathbf{0}_{1\times n} \end{bmatrix},$$

$$\mathbf{x} = \begin{bmatrix} \tau \\ \overline{\tau} \\ \overline{\mathbf{v}}_1 \\ \overline{\mathbf{v}}_2 \\ \mathbf{m} \\ \mathbf{u}_2 \\ \mathbf{u}'_2 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{u} + \mathbf{A}\mathbf{a}_1 - \mathbf{B}\mathbf{a}_2 \\ (\mathbf{I}_n \otimes \mathbf{g})\mathbf{a} \\ 0 \end{bmatrix},$$

with the set of constraints $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$ where

$$\mathcal{M}_1 = \big\{(i, i, i) \,|\, i \in [2, 1 + k' + m_1 k_1 + m_2 k_2 + m]\big\},$$

corresponds to the binary coefficients, while the relation $\mathbf{u}'_2 = \tau \mathbf{u}_2$ is added through

$$\mathcal{M}_2 = \big\{(1{+}k'{+}m_1k_1{+}m_2k_2{+}m{+}n{+}i, 1, 1{+}k'{+}m_1k_1{+}m_2k_2{+}m{+}i) \,|\, i \in [1, n]\big\}.$$

With this formulation, it is possible to exploit the efficient lattice-based ZKP of knowledge of Yang et al. [25].

Security parameters. For sake of clarity, the parameters considered in previous protocols can be listed as follows for 128 bits of security (for the exact proof case in [13]):

- the prime q has size $|q| = 128$, while $q' = 2^{61}$;
- the length of the message $\mathbf{m}$ is $m = 128$;
- the SIS problem dimension is $n = 495$;
- the trapdoor dimensions are $s_1 = 48732$ and $s_2 = 77220$;
- the sampling widths are $\sigma = 6026.03, \sigma_1 = 6026.05$ and $\sigma_2 = 12.73$.

With these values, the resulting data are:

- the public parameters that require 1.2 MB;
- the public key with $|\mathsf{pk}| = 1160$ MB;
- the private key with $|\mathsf{sk}| = 898$ MB;
- the signature with $|\mathsf{sign}| = 262$ MB.

Achieving selective disclosure. Both the classical schemes introduced before are directly suitable for Anonymous VCs and, with small modifications, for selective disclosure. The property that can be exploited to obtain such results is the possibility to produce a single signature for a whole block of messages. At the moment, the post-quantum schemes introduced above work only with a single message, so that their use for Anonymous VCs is possible but heavy for both data-size and performance. More importantly, the feasibility of the selective disclosure is still an open challenge.

6 Solving Revocation with an Efficient Credential Update

An efficient revocation mechanism represents a crucial functionality, needed in any credential system, independently of any privacy-enhancing features it offers (*e.g.*, anonymity or selective disclosure). The possible reasons why a credential needs to be revoked can be grouped in three categories:

1. *Natural expiration* - it is the most common case, where the credential reaches its expiration date, thus becoming outdated and unusable;
2. *Issuer-initiated revocation* - the Issuer revokes the credential before its expiration date, because for some reason the Holder has lost its permission to use the credential (*e.g.*, a driver's license revoked because of speeding);
3. *Holder-initiated revocation* - the Holder asks for a revocation of its credential, for instance in the case of credential theft or because its secret key has been compromised.

An efficient solution for the revocation and update of Anonymous VCs was introduced by Camenisch, Kohlweiss, and Soriente [6]. This solution overcomes the limitations of traditional credential revocation approaches, based on certificate revocation lists or online certification authorities, that are unsuitable to privacy-sensitive contexts.

The idea proposed by Camenisch, Kohlweiss, and Soriente consists in a non-interactive technique to update Issuer-controlled attributes of the VC. In this case, the VC contains a signature of the Issuer on a number of attributes, some of them secret and known only by the Holder and some of them chosen by the Issuer. Credential revocation is implemented by encoding a validity time property (*e.g.*, a simple *timestamp*) into one of the Issuer-controlled attributes. Thus, an Issuer can periodically update valid credentials off-line and publish a small per-credential update value, for instance on a public bulletin-board. For the Issuer, this process consists in:

- the update of the validity time property for each non-revoked credential;
- the re-computation of a small portion of the Issuer signature on the updated credential;
- the publication of such small signature update in a set of updates for all the non-revoked credentials, identified with unique serial numbers or, preferably, by means of pseudonyms.

A credential Holder can later download its update and re-validate its credential to prove possession of a valid credential for the current time period.

This solution has the advantage that the Verifier does not need to check any revocation lists, thus the showing and verification of credentials are as efficient as possible, without extra work or space incurred by enabling revocation. Moreover the costs for updating credentials are minimal for the Holder and are comparable to other solutions for the Issuer. Furthermore, this approach enables a *rich revocation semantic*, since the credential can be partially (*i.e.*, only some of the credential attributes) revoked and/or updated.

In principle, this solution is applicable to all the classical zero-knowledge VCs schemes previously presented in Section 4, including both their Anonymous VCs and Selective Disclosure VCs variants.

Concerning the RSA-based scheme presented in Section 4.1, the revocation mechanism is directly applicable as follows.

The *blind signature scheme* (see page 10) can be modified by splitting the final set of the messages m_i's signed by the Issuer in:

- $m_1, \ldots, m_K$ secret messages, known only by the Holder (with $K < L$);
- $m_{K+1}, \ldots, m_L$ additional attributes, chosen and controlled by the Issuer and known to the Holder. For instance, the last attribute m_L can be the validity time property.

The first steps of the protocol between Holder and Issuer remain unchanged, apart the definition of the commitments on the secret messages, that now are:

$$C = \prod_{i=1}^{K} g_{C_i}^{m_i} h_C{}^{r_C} \pmod{n_C} \qquad \text{and} \qquad C_m = \prod_{i=1}^{K} a_i^{m_i} b^r \pmod{n}.$$

Instead of directly signing C_m, the Issuer computes an extended commitment C'_m with the Issuer-controlled attributes $m_{K+1}, \ldots, m_L$

$$C'_m = C_m \prod_{i=K+1}^{L} a_i^{m_i}.$$

Then, the Issuer signs such extended commitment by:

- choosing a random prime e of ℓ_e bits;
- choosing a random integer r' of ℓ_s bits;
- computing the value $v = (C'_m b^{r'} c)^{e^{-1} \pmod{(p-1)(q-1)}} \pmod{n}$.

Finally, the Holder evaluates $s = r + r'$ and outputs $\sigma = (e, s, v)$, that is a valid signature on $m_1, \ldots, m_L$.

The protocol for the *ZKP of knowledge of a signature* (see page 11) remains basically unchanged, apart the fact that the Holder now must selectively disclose at least some of the Issuer-controlled attributes $m_{K+1}, \ldots, m_L$ to the Verifier (*i.e.*, at least the validity time property m_L). In this way, the Verifier can check if the credential is still valid (*i.e.*, not revoked or expired).

Finally, the Issuer implements the *credential update and revocation* process by:

- periodically updating the Issuer-controlled attributes $\hat{m}_{K+1}, \ldots, \hat{m}_L$ for each non-revoked credential;
- recomputing the extended commitment $\hat{C}'_m = C_m \prod_{i=K+1}^{L} a_i^{\hat{m}_i}$;
- maintaining the values of the signature e and r' unchanged;
- computing an updated value $\hat{v} = (\hat{C}'_m b^{r'} c)^{e^{-1} \pmod{(p-1)(q-1)}} \pmod{n}$.

Once receiving the updated attributes $\hat{m}_{K+1}, \ldots, \hat{m}_L$ and the updated value $\hat{v}$, the Holder simply replaces the previous value v with $\hat{v}$ and obtains an updated signature $\hat{\sigma} = (e, s, \hat{v})$. This process is repeated until the credential is revoked or expired.

In principle, this solution is also applicable to the DH-based VCs previously introduced in Section 4.2. As observed before, the lattice-based scheme introduced in Section 5.1 works only with a single message, and for now there are no options supporting multiple messages. Thus, exactly as the selective disclosure, new ideas are required in order to achieve an efficient revocation mechanism and this remains another big open issue for the post-quantum scheme.

7 Counteract the Misuse of Classical VCs with Holder Binding

The Holder Binding is proposed as a solution enabling any Holder to prove that he holds a valid credential and that such VC has been actually issued to him by a specific Issuer. This proof represents an additional functionality with respect to the common verification of the integrity and the validity of the VC. The objective is to protect the credential system against the potential misuse of a classical VC (*e.g.*, credential theft and impersonation of the Holder).

It can be implemented in the form of a *Linked Blinded Secret* (LBS), as follows:

1. the concept starts from a *secret* integer x, generated and known only by the Holder;
2. the Holder computes a commitment C_x on the secret x that, in this way, is *blinded* and can be safely shared;
3. the Issuer receives C_x from the Holder and, after proper verification on the knowledge of x, the Issuer embeds C_x into a VC and signs it together with other credential attributes, generated by the Issuer as plaintext. Thus, the secret results *linked* to the VC;
4. finally, the Issuer sends the resulting VC to the Holder. This VC includes the LBS, since it contains a blind signature on the secret x (*i.e.*, on the committed secret C_x) whose knowledge can be proved to a Verifier.

This idea is directly applicable to the previous RSA-based scheme presented in Section 4.1. In detail, the *blind signature scheme* (see page 10) and the interactions between the Holder and the Issuer can be modified as follows:

- among the messages $m_1, \ldots, m_L$ signed by the Issuer, only one message is actually a secret, known by the Holder (*e.g.*, the last message $m_L = x$);
- the other messages (*i.e.* $m_1, \ldots, m_{L-1}$) correspond to credential attributes chosen by the Issuer and known to the Holder;
- the Holder computes the commitment $C_x = a_L^x b^r \pmod{n}$, where the randomness $r \in \mathbb{Z}_n$ is also known only by the Holder;
- the commitment C_m is now defined as $C_m = C_x \prod_{i=1}^{L-1} a_i^{m_i} \pmod{n}$;

– the Holder proves the knowledge of x (not $m_1, \ldots, m_{L-1}$) and r to the Issuer;
– the Issuer blindly signs C_m, that embeds C_x (and, then, x);
– finally, as in the original scheme, the Holder evaluates $s = r + r'$ and outputs $\sigma = (e, s, v)$. Such values are a valid signature on $m_1, \ldots, m_L$ and, then, represent an unblinded signature on x.

At this point, the Holder can take advantage of the previous protocol for the ZKP of knowledge of a signature (see page 11). This protocol between the Holder and the Verifier remains basically the same, apart an important simplification: now the Holder must prove the knowledge of the secret x, while the other attributes $m_1, \ldots, m_{L-1}$ can be disclosed as plaintext to the Verifier.

After this analytical description, Figure 5 provides a concrete example of credential with the proposed solution for the Holder Binding (*i.e.*, the LBS). This example includes both a VC and a VP that take advantage of the RSA-based scheme (*i.e.*, the CL signature scheme and protocols) adapted for selective disclosure, as previously presented in Section 4.1.

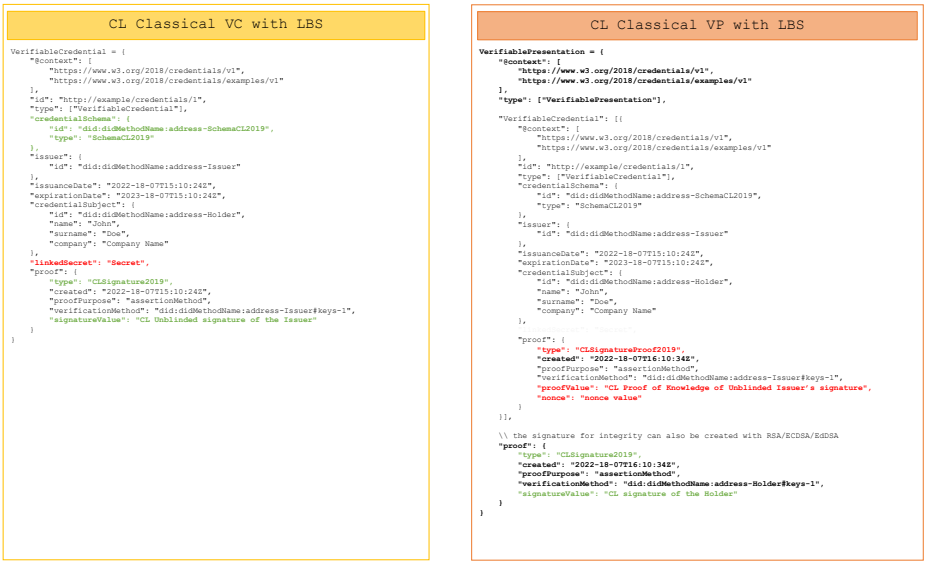

Fig. 5. Proposed CL-based VC (left) and VP (right) with LBS.

In detail, Figure 5 is similar to the previous example in Figure 4 (*i.e.*, a classical VC and VP for John Doe, an employee of a specific company). Nonetheless, the VC on the left side of Figure 5 is characterized by the following differences, highlighted with green and red colours:

1. it includes a `credentialSchema` property that is specified by the Issuer. This property points to a template, enforcing a specific structure and data

format for the VC. It can be used by a Verifier to determine if the structure and contents of the VC conform to the published schema and if the proof provided within the VC is valid (as specified in [24]);
2. it embeds the secret (*i.e.*, x, known only by the Holder) in a new dedicated property (*i.e.*, `linkedSecret`);
3. instead of a simple RSA signature, it adopts the CL signature scheme in the proof (*i.e.*, with `type` equal to `CLSignature2019`, in this example).

Note that the Holder obtains this VC at the end of the initial interactions with the Issuer (*i.e.*, after the protocol related to the *blind signature scheme*). The Holder never reveals the `linkedSecret` to the Issuer, since he discloses just a blinded version of x (*i.e.*, the commitment C_x). After proper verification, the Issuer blindly signs x, together with other credential attributes. Thus, the `signatureValue` field contains the CL *unblinded* signature $\sigma = (e, s, v)$, computed by the Holder after receiving the blind signature from the Issuer.

On the right side of Figure 5, the VP shows the following peculiarities:

1. the `linkedSecret` property is hidden, since the Holder never discloses the secret x to the Verifier;
2. the `proof` at the end of the `VerifiableCredential` does not directly contains the CL unblinded signature of the Issuer, but it contains a proof of knowledge of it according to the previous protocol for the ZKP of knowledge of a signature. Moreover, the `type` is equal to `CLSignatureProof2019` (instead of `CLSignature2019`) and the proof also includes a `nonce` for anti-replay protection;
3. finally, the second `proof` at the end of the VP includes another CL signature. The Holder computes such additional signature for integrity and, in principle, other signature schemes can be adopted (*e.g.*, RSA, ECDSA, or EdDSA).

For the sake of completeness, Figure 6 provides another practical example of implementation of the LBS concept based on BBS+.

This example adopts the protocols for the BBS+ scheme previously presented in Section 4.2 instead of the CL signature and protocols. For this reason, it can be noticed that the proofs in the VC and VP now use a `type` equal to `BbsBlsSignature2020` and `BbsBlsSignatureProof2020`.

8 Conclusions and Future Research

We described in details three zero-knowledge VCs schemes, of which the first two are based on classical cryptographic assumptions, and the third one is designed to be quantum-resistant.

As a topic of future research, it might be interesting to study how to extend the lattice-based scheme of Section 5.1 in order to support multiple messages, selective disclosure, and revocation mechanisms. Furthermore, it also seems worth investigating how VCs could be implemented using post-quantum primitives that are not lattice-based, such as hash-based schemes. Finally, both the implementation issues and the cryptographic agility of these solutions should be explored.

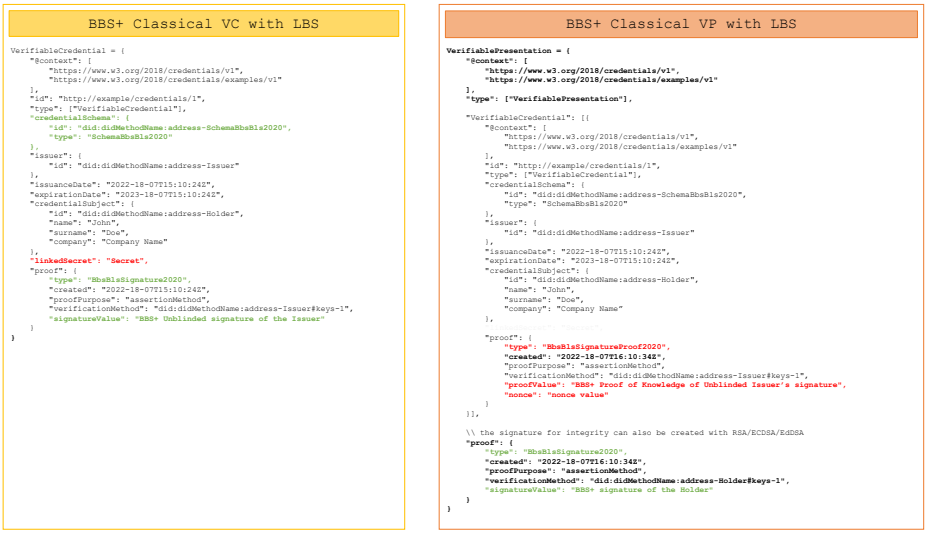

Fig. 6. Proposed BBS+-based VC (left) and VP (right) with LBS.

References

1. Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., Smith-Tone, D.: NIST IR 8413. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. Tech. rep., National Institute of Standards & Technology (2022)
2. Au, M.H., Susilo, W., Mu, Y.: Constant-size dynamic k-TAA. In: Security and Cryptography for Networks. pp. 111–125. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
3. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Advances in Cryptology – CRYPTO 2004. pp. 41–55. Springer Berlin Heidelberg (2004)
4. Buterin, V.: Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform (2014), https://ethereum.org/whitepaper
5. Camenisch, J., Drijvers, M., Lehmann, A.: Anonymous attestation using the strong diffie hellman assumption revisited. In: Franz, M., Papadimitratos, P. (eds.) Trust and Trustworthy Computing. pp. 1–20. Springer International Publishing, Cham (2016)
6. Camenisch, J., Kohlweiss, M., Soriente, C.: Solving revocation with efficient update of anonymous credentials. In: Garay, J.A., De Prisco, R. (eds.) Security and Cryptography for Networks. pp. 454–471. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
7. Camenisch, J., Lysyanskaya, A.: A signature scheme with efficient protocols. In: Security in Communication Networks. pp. 268–289. Springer Berlin Heidelberg (2003)
8. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Advances in Cryptology – CRYPTO 2004. pp. 56–72. Springer Berlin Heidelberg (2004)

9. Damgård, I., Fujisaki, E.: An integer commitment scheme based on groups with hidden order. Cryptology ePrint Archive, Report 2001/064 (2001), https://eprint.iacr.org/2001/064
10. Feynman, R.P.: Simulating physics with computers. International Journal of Theoretical Physics **21**(6), 467–488 (1982)
11. Fujisaki, E., Okamoto, T.: A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In: Nyberg, K. (ed.) Advances in Cryptology — EUROCRYPT'98. pp. 32–46. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)
12. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing. pp. 212–219. ACM (1996)
13. Jeudy, C., Roux-Langlois, A., Sanders, O.: Lattice-based signature with efficient protocols, revisited. Cryptology ePrint Archive, Paper 2022/509 (2022), https://eprint.iacr.org/2022/509
14. Kannengießer, N., Lins, S., Dehling, T., Sunyaev, A.: Trade-offs between distributed ledger technology characteristics. ACM Computing Surveys **53**(2), 1–37 (2020)
15. Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S.: Pseudonym systems. In: Selected Areas in Cryptography. pp. 184–199. Springer Berlin Heidelberg (2000)
16. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008), https://bitcoin.org/bitcoin.pdf
17. Popov, S.: The Tangle (2018), https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk-0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf
18. Pornin, T., Kircanski, A., Hemmel, M., Wong, D., Ghazizadeh, J., Hall-Andersen, M., Samuel, J.: Zcash overwinter consensus and sapling cryptography review. Tech. rep., NCC Group, Version 1.3 (2019), https://research.nccgroup.com/wp-content/uploads/2020/07/NCC_Group_Zcash2018_Public_Report_2019-01-30_v1.3.pdf
19. Preukschat, A., Reed, D.: Self-Sovereign Identity – Decentralized digital identity and verifiable credentials. Manning, Shelter Island, NY (2021), https://www.manning.com/books/self-sovereign-identity
20. PROMETHEUS: https://www.h2020prometheus.eu
21. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science. pp. 124–134. IEEE Computer Society (1994)
22. Veugen, T., Ricosset, T., Sanders, O., Herranz, J.: D5.2: Intermediate results on privacy-preserving cryptographic protocols. Tech. rep., Project PROMETHEUS (2019)
23. W3C: Decentralized Identifiers (DIDs) v1.0. Core architecture, data model, and representations. W3C Recommendation (2022), https://www.w3.org/TR/did-core/
24. W3C: Verifiable Credentials Data Model v1.1. W3C Recommendation (2022), https://www.w3.org/TR/vc-data-model/
25. Yang, R., Au, M.H., Zhang, Z., Xu, Q., Yu, Z., Whyte, W.: Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In: Advances in Cryptology – CRYPTO 2019. pp. 147–175. Springer International Publishing (2019)