IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 November 2024, accepted 8 December 2024, date of publication 16 December 2024,
date of current version 26 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3518597

RESEARCH ARTICLE

BLS-MT-ZKP: A Novel Approach to Selective Disclosure of Claims From Digital Credentials

ŠEILA BEĆIROVIĆ RAMIĆ, (Graduate Student Member, IEEE), IRFAN PRAZINA, DAMIR POZDERAC, RAZIJA TURČINHODŽIĆ MULAHASANOVIĆ, (Member, IEEE), AND SAŠA MRDOVIĆ

Faculty of Electrical Engineering, University of Sarajevo, 71000 Sarajevo, Bosnia and Herzegovina

Corresponding author: Šeila Bećirović Ramić (sbecirovic1@etf.unsa.ba)

This work was supported by the Federal Ministry of Education and Sciences of Bosnia and Herzegovina under Grant 0101-10880-11/23.

ABSTRACT Digital credentials represent crucial elements of digital identity on the Internet. Credentials should have specific properties that allow them to achieve privacy-preserving capabilities. One of these properties is selective disclosure, which allows users to disclose only the claims or attributes they must. This paper presents a novel approach to selective disclosure BLS-MT-ZKP that combines existing cryptographic primitives: Boneh-Lynn-Shacham (BLS) signatures, Merkle hash trees (MT) and zero-knowledge proof (ZKP) method called Bulletproofs. Combining these methods, we achieve selective disclosure of claims while conforming to selective disclosure requirements. New requirements are defined based on the definition of selective disclosure and privacy spectrum. Besides selective disclosure, specific use cases for equating digital credentials with paper credentials are achieved. The proposed approach was compared to the existing solutions, and its security, threat, performance and limitation analysis was done. For validation, a proof-of-concept was implemented, and the execution time was measured to demonstrate the practicality and efficiency of the approach.

INDEX TERMS BLS signatures, bulletproofs, digital credentials, Merkle hash trees, selective disclosure.

I. INTRODUCTION

Digital credentials represent attestations, i.e. evidence of an individual's qualifications, claims, or achievements [1]. They are the digital equivalent of ''paper'' credentials that people carry to prove their identity or qualification, e.g., identity card, driver's licence, passport and diploma. This paper focuses on this definition of digital credentials. The definition of digital credential should be distinct from the term used in other fields of computer science, where it can represent a simple password [2].

Since the term ''digital credential'' was introduced, an ongoing effort has been made to standardize it. David Chaum's theoretical construction of anonymous credentials from 1985 [3] represents the first significant development of digital credentials. These types of credentials enable users to prove that they possess credentials and can disclose information from them while maintaining anonymity [4]. Further advancements followed, including the first practical implementation by Camenish-Lysanskaya [5], to the development of attribute-based credentials in IBM's Idemix [6], [7] and Microsoft's U-Prove [8]. More recently, the focus shifted toward standardizing and developing verifiable credentials, which are tamper-evident and cryptographically verifiable. As an open standard, they are being increasingly developed to incorporate properties that enable preservation of privacy, such as selective disclosure and data minimization [9], [10]. Users share information they must or want with other parties using selective disclosure. Verifiers require information from users for authentication and to provide them with services. With selective disclosure, users can disclose only what is necessary and other information if they choose to. Selective disclosure represents a privacy-enhancing mechanism that has been extensively studied in recent years [1]. Currently, the selective disclosure research area is expanding, with different approaches introduced regularly. There is no clear winner

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Ni.

or the "best" universal solution for selective disclosure in digital credentials. It is therefore necessary to find a solution that satisfies all the requirements for selective disclosure schemes and legal regulations if needed.

This paper introduces a new approach to selective disclosure of claims in digital credentials. This approach represents a combination of different elements with a concrete implementation using the hash-based method of Merkle trees, the signature-based method using Boneh-Lynn-Shacham signatures and the zero-knowledge proof (ZKP) method Bulletproofs. The aim of this work is to create an approach to selective disclosure that fulfills the requirements defined by ETSI (The European Telecommunications Standards Institute) [11]. With this approach, it is possible to selectively disclose claims from multiple digital credentials, combining them into one presentation with the ability to prove that claims belong to the user. This approach also enables multiple issuers to issue one credential and to have a user sign their credential, which complies with specific real-world examples. In addition, it enables proving that a claim value belongs to a specified range without revealing it. This is the first solution with all of these features. As far as we know there is no other approach that can achieve all selective disclosure requirements as defined in [11].

II. PRELIMINARIES

A. DIGITAL CREDENTIAL SYSTEM AND ROLES

Digital credentials are electronic certificates or documents issued by an entity to verify an individual's qualifications, claims, or accomplishments. In a digital credentials system, users rely on these certificates because they are easy to manage and verify. The following explanation of credential system is based on a verifiable credentials system because it is the current standard. The credential system has three roles: Issuer, Holder, and Verifier. Figure 1 shows the process of issuance and verification alongside the user roles. Issuer issues and signs the credentials provided to the holder. The issuer saves the issuance record in a publicly available registry. Holder keeps their credentials and sends them as a presentation to verifier when needed. Verifier checks the validity of the issued credential against the information in the publicly available registry [12].

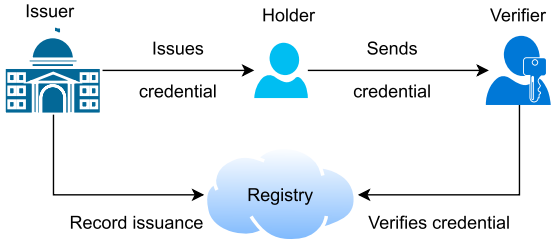


FIGURE 1. Process of digital credential issuance and user roles.

In the context of non-selective verification, the validity of the credential is checked based on the digital signature of the entire credential. An open issue with credential verification is how to trust the credential when some claims or parts necessary for signature verification are missing during the selective disclosure.

B. SELECTIVE DISCLOSURE

Selective disclosure is a mechanism that allows holders to reveal only information that they must and to whom they must. It also allows them to reveal only the information they want and to whom they want. Selective disclosure is a mechanism designed to preserve the privacy of individuals and organizations. It serves as a privacy design pattern that enhances users' trust, privacy and security by enabling data minimization in terms of collected data, compliance with data regulation (General Data Protection Regulation (GDPR) [13] and the California Consumer Privacy Act (CCPA) [14]) and determining who can access user data and under what conditions [15], [16].

The development of digital identity has led to the establishment of regulatory and legal frameworks. In 2014, the European Union Council introduced a regulation, eIDAS (Electronic Identification, Authentication and Trust Services), which supports secure cross-border transactions by establishing a digital identity and authentication framework. In 2023, a provisional agreement on updating and modifying eIDAS was reached called eIDAS2. This revised regulation outlines requirements for selective disclosure to ensure convenience and personal data protection, including minimizing its processing [17]. Due to the necessity of selective disclosure, several standardization drafts are currently developed by ETSI [11] and the European Blockchain Services Infrastructure (EBSI) [18]. ETSI defines the following requirements for selective disclosure [11]:

1) The possibility that the user selectively discloses attributes so that these attributes appear to be part of an attestation/credential other than the one they were originally part of;
2) The possibility to selectively disclose attributes from at least two separately issued attestations issued by the same issuer;
3) The possibility to selectively disclose attributes from at least two separately issued attestations issued by different issuers;
4) The possibility to selectively disclose attributes from a single attestation;
5) The selectively disclosed attributes are unlinkable by means other than the information shared in the attribute over multiple sharing sessions of the disclosed attributes to at least two different verifiers who can collude and compare the attribute disclosures they have received;
6) The selectively disclosed attributes are unlinkable by means other than the information shared in the attribute over multiple sharing sessions of the disclosed attributes to the same verifier;

7) The selectively disclosed attributes are unlinkable by means other than the information shared in the attribute over multiple sharing sessions of the disclosed attributes to at least one verifier who can collude with the credential issuer and show the attribute disclosures they have received;
8) Whether or not the verifier, upon receipt of selectively disclosed attributes, can confirm that the attributes were issued to the same identity subject that is presenting the attributes (or to an authorized representative thereof) and to no one else;
9) Whether or not the verifier, upon receipt of selectively disclosed attributes, can confirm that the attributes describe the same identity subject that is presenting the attributes (or to an authorized representative thereof) and to no one else.

These requirements define that the approach to selective disclosure must:

- support multi-show unlinkability (5, 6 and 7);
- support combined presentations (2 and 3);
- be resilient against colluding parties (5 and 7);
- assure holder binding and proper pairing of presented disclosures (1, 4, 8 and 9).

Additional requirements can arise from observing the spectrum of privacy for digital credentials, which ranges from pseudonymous to strongly identified. Individuals have varying comfort levels regarding the information they are willing to share and the insights that can be inferred from it. This is especially true when dealing with sensitive data such as medical records [19]. A key selective disclosure element is understanding that sometimes only proving a value without revealing it is needed, e.g., proving that one's age is over 21 [20].

From the definition of selective disclosure and the privacy spectrum, we explicitly define following additional requirements:

10) The disclosed attributes are part of a valid credential, and by using them, it is possible to validate the entire credential;
11) The disclosed attributes are part of a valid credential issued by an issuer whose origin (issuer and issuance) can be verified;
12) It is possible to prove the properties/values of attributes (range proofs, set membership) without disclosing them to protect privacy.

The memory and time requirements for selective disclosure key components (issuing credentials, generating and verifying the presentation) are currently not defined. However, since selective disclosure must be performed on different agents, which means on different devices (computers, servers, mobile devices), it is necessary to find an approach to selective disclosure that can be performed on all of them. In addition, the approach's scalability is crucial for real-world use cases. Standardization of approaches must also consider potential changes in the post-quantum period [11].

We recognized three use case scenarios to evaluate the usability of the selective disclosure approach.

Figure 2 shows the main use case scenario of selective disclosure. For example, a university student receives their diploma as a digital credential. They send their diploma to potential employers when they apply for a job. Certain employers do not need all of the data in the diploma. Therefore, the student shares the university name and degree name with employer A while they share their grade and domain of study with employer B.

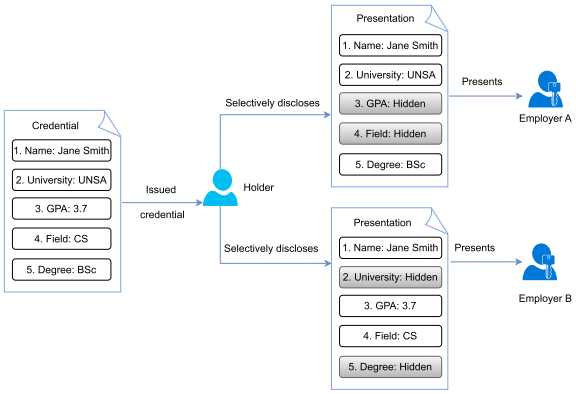


FIGURE 2. Selective disclosure use case - single credential.

Another use case of selective disclosure is combining information from two credentials to send to an employer. The use case is shown in Figure 3. The user receives digital credentials for a driver's licence from the government and a diploma from the university. They combine these into one presentation and send it to employers. The claim about validity and category from the first credential and school and GPA from the other are added to one combined presentation.

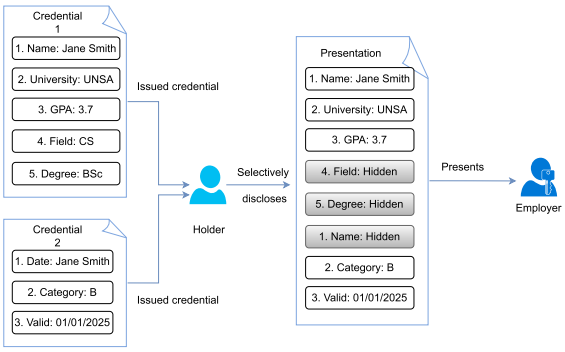


FIGURE 3. Selective disclosure use case - multiple credentials.

The third use case requires proving something has a value within required limits or in a required set without revealing that value. To achieve that, zero-knowledge proof is commonly used. For example, a student doesn't want to reveal their GPA, but they can derive evidence showing that the GPA is in the required range. The use case is shown in Figure 4.

This use case is crucial in privacy-preserving contexts, i.e., when somebody needs to prove their age without revealing it, their salary range, or that they belong to a particular group.

In all use cases, employers should be able to confirm the validity of the presentations they receive using the public registry.

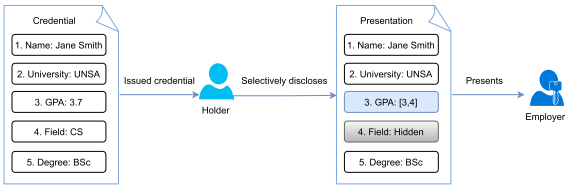


FIGURE 4. Selective disclosure use case - zero-knowledge proof.

Different methods are used to achieve selective disclosure. Authors of [21] define the following methods:

- Atomic Credentials;
- Selective disclosure signatures;
- Hashed values.

Atomic credentials consist of a single claim, allowing users only to reveal the necessary claims/credentials during selective disclosure. Even though they are the most straightforward approach, atomic credentials are hard to manage. The number of credentials and overhead are increased. Additionally, there is no assurance that two claims can be correctly paired, which could result in presenting credentials with two unrelated or incorrect claims about the same subject. For example, two credentials about car type and car mileage from two different cars can be combined into one presentation.

Atomic credentials option is generally discarded as a viable and applicable one for selective disclosure. They define an additional method, zero-knowledge proof, for selective disclosure, besides hash- and signature-based methods. They also show that a combination of methods can achieve selective disclosure.

Hash-based methods hash claims about the subject in the credential. The user shares the presentation; they provide the hashed claims alongside the actual values of claims they wish to disclose. The verifier then hashes the provided values to verify that they match the original hashes. Merkle hash trees [22], [23], [24], hidden commitment schemes [25], [26], [27], and hashed message authentication codes [28] are the most commonly used. Signature methods use particular types of signatures that allow the disclosure of specific claims. Most commonly used are CL-signatures [5], [29], [30], [31], [32] and BBS+ [12] signatures. Zero-knowledge proof methods commonly implement zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) for achieving selective disclosure of claims [33], [34].

Analyzing the existing approaches, we can see that the three methods and their combinations have certain advantages and disadvantages. Hash- and signature-based methods require showing the claims, while ZKP methods hide them/prove them. Both showing and hiding/proving the claims are necessary for an all-encompassing digital credential system. Hash methods are the least computationally intensive and least complex of the three, while ZKP methods are the most intensive and complex ones. Hash methods and ZKP methods alone cannot verify the issuer or holder, which is one of the key elements in digital credentials. Methods that use digital signatures usually add another layer of operations due to the canonicalization algorithm needed for claims. By combining specific methods, we can achieve what is lacking using only one method, but it can also add high computational costs and complexity. The approach should focus on fulfilling the requirements and enabling standard features, such as those in physical credentials, to achieve widespread adoption.

The approach presented in this paper combines all three methods for selective disclosure to fulfill the requirements and scenarios. It combines Merkle trees, BLS signatures, and Bulletproofs to achieve selective disclosure. Below is a short description of each element to increase understanding.

C. BLS SIGNATURES

The BLS (Boneh-Lynn-Shacham) signature scheme uses bilinear pairings for verification, with signatures represented as elements on an elliptic curve [35]. The scheme has the following characteristics [36]:

- Uniqueness and determinism: There is only one valid signature for any key or message:
- Signature Aggregation: Multiple signatures created with different public keys for various messages can be combined into a single aggregated signature;
- Threshold signatures: The scheme allows threshold issuing where multiple signers collaborate in producing a single signature.

The BLS signature scheme is provably secure in the random Oracle model and unforgeable under adaptive chosen message attacks. This security relies on the intractability of the computational Diffie-Hellman problem in a gap Diffie-Hellman group.

This scheme consists of three primary functions for generation, signing and verifying signatures [35]:

- generate - Algorithm for key generation selects a random integer x such that $0 < x < r$. The private key is x, and the corresponding public key is g^x, where g is a generator of the group;
- sign - Given the private key x and a message m, the signature is computed by hashing the message m, into a point on the elliptic curve $h = H(m)$ and then computing the signature as $\sigma = h^x$;
- verify - To verify a signature, it is necessary to check whether the equation $e(\sigma, g) = e(H(m), g^x)$ holds where e is the bilinear pairing function.

D. MERKLE TREE

A Merkle tree, or hash tree, is a data structure that ensures data integrity and consistency for verification and

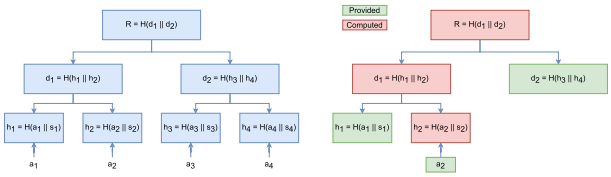


FIGURE 5. Merkle tree generation and Merkle tree verification.

synchronization. Named after Ralph Merkle, who patented it in 1979 [37], the Merkle tree is a structure where each leaf node is labeled with the hash of data. Branch nodes are labeled with the hash of their child nodes. Merkle tree allows for efficient and secure verification of the contents of large datasets [22]. Figure 5 shows the Merkle tree of elements and the verification method that an element belongs to the tree. Hashes for each attribute are calculated $h_i = H(a_i||s_i)$, where s_i represents salt and is stored in the leaf nodes. The leaves root is determined by hashing pairs of nodes together $d_i = H(h_i||h_j)$. This process continues until the Merkle tree's root hash is obtained. To prove that element a_2 is part of the tree, the root is recreated using the element and its inclusion path $[a_2, h_1, d_2]$. If the reconstructed root matches the original root, the verification is complete.

E. BULLETPROOFS AND PEDERSEN COMMITMENT

A ZKP method is necessary for the third scenario, where Bulletproofs are utilized for range proofs. Bulletproofs are a non-interactive zero-knowledge (NIZK) protocol designed for general arithmetic circuits, which produce compact proofs without a trusted setup. Bulletproofs are based on the discrete logarithm assumption. Using the Fiat-Shamir heuristic they are made non-interactive. The name comes from a description of its properties by one of the authors: "Short like a bullet with bulletproof security assumptions" [38].

The core of the protocol is in its inner-product argument, initially presented by Groth [39] and later refined by Bootle et al. [40] This refinement provided proof for two unrelated binding vectors, Pedersen Commitments, that satisfy the inner-product relation. Building on this, Bulletproofs offer communication-efficient, zero-knowledge proofs, including range proofs derived from inner-product calculations on Pedersen commitments [41].

A commitment scheme is a cryptographic primitive that allows a prover to commit to a chosen value without revealing it to the verifier (hiding) while ensuring the value cannot be altered (binding). It is widely used for creating blinded, non-interactive commitments [42]. Bulletproofs are an efficient implementation that uses elliptic curve cryptography (ECC), called the Elliptic Curve Pedersen Commitment. For a value x, a random blinding factor r, a generator point G and a point H such that $H = x_H G$ (where x_H cannot be determined without solving the elliptic curve discrete logarithm problem), the commitment is calculated as $C(x, r) = xH + rG$. This commitment is homomorphic such that for messages x, x_0 and x_1, blinding factors r, r_0 and r_1 and scalar k, the following relation holds: $C(x_0, r_0)+C(x_1, r_1) = C(x_0 + x_1, r_0 + r_1)$ and $C(k \cdot x, k \cdot r) = k \cdot C(x, r)$. This commitment is computationally binding and perfectly hiding [41].

In the Bulletproof protocol, the prover needs to convince the verifier that a Pedersen commitment $C(x, r) = xH + rG$ contains a value x within the range $x \in [0, 2^n - 1]$. To achieve this, the prover represents x as a vector of its binary bits $a = (a_1, \ldots, a_n)$ where each a_i is either 0 or 1. The core idea is to conceal all the bits in a single vector Pedersen Commitment. The proof involves showing that each bit ω satisfies $\omega(\omega - 1) = 0$, confirming that ω is either 0 or 1 and that their sum equals x. Throughout the protocol, the verifier sends random linear combinations of constraints and challenges $\in \mathbf{Z_p}$ to the prover. In response, the prover constructs a vectorized inner product relation that contains vector a, the constraints and challenges $\in \mathbf{Z_p}$, and appropriate blinding vectors $\in \mathbf{Z_p^n}$. Because Pedersen commitments allow for a vector to be split and compressed, the number of rounds for the inner product is reduced to a logarithmic number of rounds [41]. These elements are necessary for creating an approach to selective disclosure that fulfills the requirements.

III. RELATED WORKS

Merkle hash tree has been used to implement credentials with selective disclosure as demonstrated in several studies [22], [23], [24]. In [22], authors use the Merkle tree for decentralized identity and store the root on the blockchain. Authors of [23] use encryption on each attribute in order to prevent attack, while the authors of [24] present a credential as a Merkle B+ tree where two nodes are for the same attribute (control node and data node). Besides the Merkle tree, other hashing techniques such as keyed-hash message authentication codes [28], [43] or hidden commitment schemes are used [25], [26], [27]. These methods rely on hashing individual attributes. The attribute is revealed alongside salt during disclosure, while others are sent in their hashed form. On the other hand, Merkle tree do not require sending all of the attributes, but rather the proof.

BLS signatures are a method used for selective disclosure, where each claim is individually signed, and an aggregated signature is created that represents the credential signature [44]. Besides BLS signatures, the most commonly used signatures for selective disclosure are those based on special functions from elliptic curve cryptography [45], [46], CL signatures [32], and BBS+ signatures [12]. A key element of these signatures is that they require a canonicalization algorithm to be used for all attributes. This adds complexity to the solution and, in most cases, increases the size of the signature (depending on the number of attributes).

The ZKP method used in papers [33], [34] is Groth-16, a zk-SNARKs method. This method requires a trusted setup, which adds additional complexity. It should be noted that other ZKP methods that do not require a trusted setup, such as Bulletproofs, are also used, while zk-STARKs methods are not commonly used in digital credentials [47].

To meet specific requirements for selective disclosure, different researchers have combined various approaches. The following papers present combined approaches for selective disclosure. The authors of [48] propose the CredChain architecture. This architecture integrates a hash-based method with redactable signatures. In this approach, each credential is represented by a Merkle tree with hashed attributes, and the root is signed with a redactable signature that includes both the signature and auxiliary information. During disclosure, the auxiliary data is updated in the signature. This method allows the credential to be signed once, generating multiple claims while minimizing user interaction and preventing correlation.

In [49], authors propose combining BLS signatures with hashing. The credentials attributes are hashed alongside the user's digital identifier (DID), which serves as a salt. Each claim is signed, and the aggregated signature is the signature of the entire credential.

The Coconut scheme introduced in [50] represents a selective disclosure credential scheme that combines threshold issuance signature scheme with ZKP for attribute proving.

Pointhcheval-Sanders Multi-Signatures (PS-MS) pairing scheme is used in paper [51]. Using this approach, public key shares are aggregated into a single public key. This enables verification of a credential that includes ZKP of selectively disclosed values.

The authors of [52] expand the vector-commitments with zero-knowledge proof of knowledge (ZPKP) protocols to create a commitment, retrieve and update values. This method also involves blind signatures but has the drawback of being a one-time solution, meaning it requires re-issuance for each use.

Structure-preserving signatures on equivalence classes (SPS-EQ) with fully adaptive NIZK proofs (AND and NAND proofs) is used in paper [53] which allows signer-hiding as well.

Merkle tree combined with Poseidon hash for zk-SNARKs is introduced in paper [54]. This approach adds additional leaves to the tree, where the right half of tree contains attributes while the left represents metadata. Zk-SNARKs is used when needed. Otherwise, the Merkle tree is used in constrained devices.

These methods usually combine two approaches: hash- and signature-based, ZKP and signature-based, or ZKP and hash-based to achieve additional functionalities. Each approach combines only two of the three. The combinations do not consider the requirements for selective disclosure defined in II nor do they satisfy all of them (e.q. some approaches are focused only on unlinkability, while most of them do not consider combining credentials into one presentation). Combining all three approaches in a manner explained in the following section allows the fulfillment of all theoretical requirements and the implementation of practical solutions for selective disclosure without additional complexity.

IV. COMBINED APPROACH TO SELECTIVE DISCLOSURE

The primary purpose of the following approach is to present a solution for selective disclosure. Each cryptographic primitive used in this combined approach is chosen due to the specific requirement it can fulfill:

- A Merkle tree with different salts enables:
  - -- Proving that disclosed attributes are part of a valid credential (requirements: 1, 4, 8, 9 and 10);
  - -- Validating the entire credential (requirements: 10);
  - -- Preventing the linkage of attributes with the same values to a certain degree (requirements: 5, 6 and 7);
  - -- Preventing recording of each attribute hash in a publicly available registry;
  - -- Creation of credentials without a fixed-size attribute lists for practical usage;
- BLS signatures enable:
  - -- Verification of the credential's origin, i.e., verification of the issuer (requirements: 11);
  - -- Verification of the identity owner(requirements: 1, 4, 8, 9);
  - -- Verification of a presentation that consists of multiple credentials (requirements: 2 and 3);
- Pedersen commitments with Bulletproofs enable:
  - -- Proving attribute values (requirements: 12);
  - -- Achieving unlinkability through homomorphic values allows proving an attribute's existence in the tree without disclosing other information if required (requirements: 5, 6 and 7);
  - -- Smaller proof sizes that reduce communication costs;

All these methods were chosen as part of the approach due to their simplicity in implementation and the possibility of execution on computers or mobile devices.

The formalization of this approach is presented through the three main procedures: issuing credentials and generating and verifying the presentation. A formal definition of algorithms is given in 1, 2 and 3.

To understand these formal procedures, the building blocks of this approach are defined and evaluated through use cases.

A. CREATING A CREDENTIAL

Digital credentials have predefined formats and should be treated as regular credentials with predefined fields and information available. They can be issued in JSON, XML, or other formats and sent in the same formats. The existence of a digital credential and verification method should be recorded in a publicly available registry, database, or DLT such as blockchain.

In this approach, we propose presenting every digital credential as a Merkle hash tree. Depending on the format of the digital credential, each attribute has a corresponding leaf in the Merkle tree, as shown in Figure 6. The Merkle tree is generated using standard hashing functions. However, the first layer of leaves, created by hashing claims, uses a

Algorithm 1 Credential Issuance

Require: Set of attributes $\{a_i, v_i\}$, where a_i is attribute name, v_i attribute value, private key sk

Ensure: Issuance record (R, σ), credential $(\{a_i\}, \{v_i\}, \{s_i\}, R, \sigma)$

1: **for** each attribute a_i in the credential **do**
2: Generate a randomized salt s_i
3: Compute the Pedersen commitment $c_i = v_i \cdot G + s_i \cdot H$ on an elliptic curve, where G and H are base points
4: **end for**
5: Construct a Merkle tree $\mathcal{M}$ with commitments $\{c_i\}$ as leaves
6: Compute the Merkle root R of $\mathcal{M}$
7: Sign the root R with the BLS signature using the private key sk to obtain $\sigma = \text{BLS.Sign}(R)$
8: **Output:**
- **Issuance Record:** (R, σ)
- **Credential:** $(\{a_i\}, \{v_i\}, \{s_i\}, R, \sigma)$

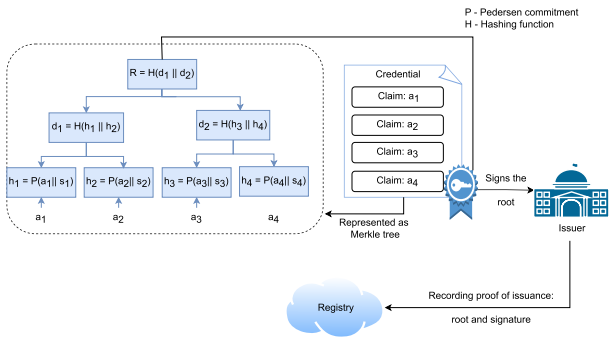


FIGURE 6. Issuance of credential.

Pedersen commitment hash in this solution. The Pedersen commitment and the appropriate salt value allow for various ZKP circuits. One ZKP method used in this solution is Bulletproofs. The root of the Merkle tree is signed using a BLS digital signature. The root and the signature are recorded in the previously mentioned public registry to have proof of issuance. Root hash represents a credential fingerprint, allowing a validity check of the credential while the signature authenticates the issuer. Due to the usage of signatures, public keys should also be publicly available and traceable to the issuer. Issuance is shown in Figure 6.

Holders can create verifiable presentations based on the credentials they have received. First, we will consider the regular use case of digital credential. This use case is shown in Figure 7. The holder creates a presentation based on a single credential. They sign the root of the credential using their private key. Afterward, they aggregate their signature with the issuer's signature into one single signature. This reduces the overall size of the signatures in the presentation and allows for verification of credential ownership. The aggregated signature is used to verify the issuer as the source of the credential and to whom it was issued. The root is used

Algorithm 2 Generating the Presentation

Require: Set of credentials $\{\text{cred}_j\}$, where each credential includes attribute names $\{a_i\}$, attribute values $\{v_i\}$, salts $\{s_i\}$, Merkle tree roots R_j, and signatures σ_j

Ensure: Presentation $(\{\text{proofs}_j\}, \sigma_{\text{agg}})$, where $\{\text{proofs}_j\}$ contains proofs and σ_{agg} is the aggregated signature

1: **for** each credential cred_j in $\{\text{cred}_j\}$ **do**
2: **for** each disclosed attribute a_i in cred_j **do**
3: **if** attribute a_i requires a range proof **then**
4: Generate range proof using Bulletproofs, denoted as $\text{proof}_{i-\text{range}}(c_i)$ on committed value
5: Generate Merkle tree membership proof for the commitment $\text{proof}_{i-\text{Merkle}}(c_i, R_j)$
6: **else**
7: Generate Merkle tree membership proof $\text{proof}_{i-\text{Merkle}}(v_i, s_i, R_j)$, using attribute value v_i and the salt s_i
8: **end if**
9: **end for**
10: **end for**
11: Aggregate signatures from each credential to obtain $\sigma_{\text{agg}} = \text{Aggregate}(\{\sigma_j\})$
12: **Output:**
- **Presentation:** $(\{\text{proofs}_j\}, \sigma_{\text{agg}})$, where $\{\text{proofs}_j\}$ includes disclosed attributes with their salts or committed attributes and their corresponding proof: Bulletproof range proofs $\text{proof}_{i-\text{range}}$ and Merkle tree membership proofs $\text{proof}_{i-\text{Merkle}}$, and σ_{agg} is the aggregated signature

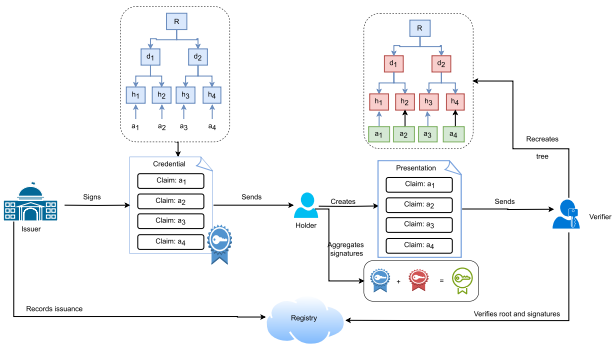


FIGURE 7. Regular use case of fully disclosing a single credential.

to verify the credential's validity. When the verifier receives the presentation, they recreate the Merkle tree to get the root. They check the root and the signatures against the public record of issuance and public keys.

B. MULTI-ISSUER CREDENTIALS

If credentials are created using the presented approach, creating different kinds of credentials is possible. One of the credentials that can be created is one signed by multiple issuers. One of the real-life examples where this is useful are university diplomas. Usually, a university diploma contains

Algorithm 3 Presentation Verification

Require: Presentation $(\{\text{proofs}_j\}, \sigma_{\text{agg}})$, where $\{\text{proofs}_j\}$ includes proofs (Bulletproofs, Merkle membership proofs) and σ_{agg} is the aggregated signature, Issuance records $\{(R_j, \sigma_j)\}$, public keys $\{pk_j\}$

Ensure: **True** if verification succeeds, **False** otherwise

for each credential proof proof_j in the presentation **do**
  for each disclosed attribute a_i in proof_j **do**
    if attribute a_i includes a range proof **then**
      Verify the range proof $\text{proof}_{i-\text{range}}(c_i)$ using Bulletproofs for commited attribute c_i
      Verify Merkle tree membership proof $\text{proof}_{i-\text{Merkle}}(c_i, R_j)$ for the commitment c_i
    else
      Verify Merkle tree membership proof $\text{proof}_{i-\text{Merkle}}(v_i, s_i, R_j)$ using attribute value v_i and the salt s_i
    end if
  end for
end for
Verify the aggregated signature σ_{agg} for the presentation using public keys $\{pk_j\}$
Output:
- **True** if all Bulletproofs, Merkle tree membership proofs, and the aggregated signature are valid
- **False** if any verification step fails

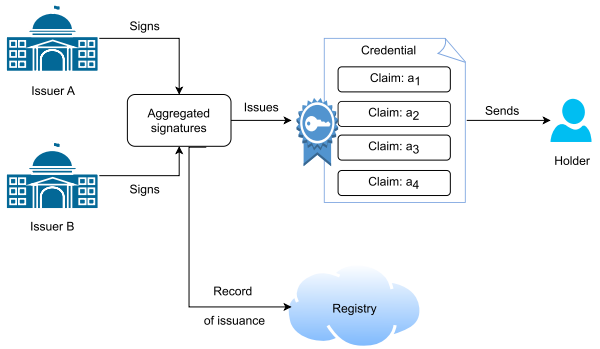


FIGURE 8. Issuing a certificate signed by multiple issuers.

the signatures of the faculty's dean and the university's rector. Using the ability to aggregate signatures, several issuers can sign one credential while producing one signature. This aggregated signature is the same size as the signature of a single issuer. Aggregated signatures can be verified using all the required public keys. Figure 8 shows an example of issuing this type of credential.

In addition to the possibility of multiple issuers of one credential, it is also possible to have a use case where the holder can sign a credential alongside the issuer (shown in Figure 9). There are specific situations where it is necessary to emphasize who the credential's owner is and if the appropriate holder received the credential, e.g., delegation

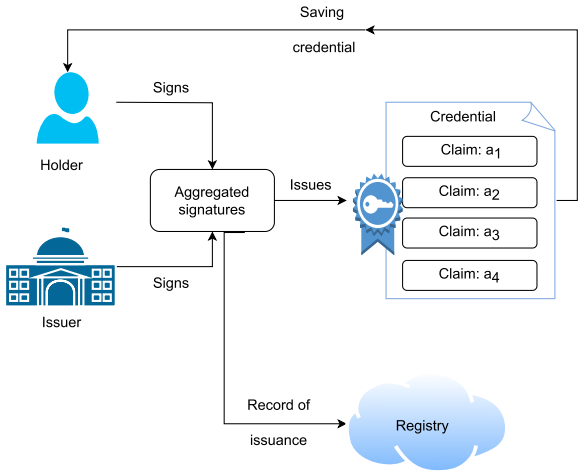


FIGURE 9. Issuing a certificate signed by the issuer and holder.

cases. Therefore, it is possible to aggregate the signatures of the holder and issuer into one credential signature. To verify the credentials, signatures are checked using recorded public keys.

C. SELECTIVE DISCLOSURE OF CLAIMS

1) USE CASE 1: SELECTIVE DISCLOSURE OF CLAIMS FROM A SINGLE CREDENTIAL

The first use case of selective disclosure of claims from a single credential is achieved primarily through Merkle trees. Figure 10 shows this use case, and the exact steps are given below:

1) Entities register their public keys in a public registry.
2) In addition to the public key, the issuer registers the credential format to enable the correct generation of the Merkle tree.
3) When issuing a credential, the issuer:
   a) Creates a digital credential and signs the Merkle tree root using BLS.
   b) Records the issuance proof, i.e. its root and signature.
4) The holder receives the credential and stores it in their digital wallet;
5) The verifier requests presentation with specified claims;
6) The holder sends their credential as a presentation to the verifier as follows:
   a) They reveal claim values that verifier requested along with their salts; for the other values, they prepare the corresponding proofs generated using the Merkle tree. The presentation now consists of all the elements needed to recreate the root of the Merkle tree;
   b) They sign the root and aggregate their signature with the issuer's signature to create a unique presentation signature;
   c) They send the presentation to the verifier.

7) When receiving a presentation, the verifier does the following:
   a) Recreates the Merkle tree using the registered format, disclosed values, salts and the resulting hashes, thus obtaining the tree's root;
   b) Verifies the root using the public registry (thus validating the credential issuance). They verify the signature of the credential, as well as the holder's signature, using recorded public keys.

2) USE CASE 2: SELECTIVE DISCLOSURE OF CLAIMS FROM MULTIPLE CREDENTIALS

The basis for this use case is BLS signatures that allow the proper merging of multiple credentials. It is shown in Figure 11, and its explanation is given below with the exact steps:

1) Entities register their public keys in a public registry;
2) In addition to the public key, the issuers register the credential format to enable the correct generation of the Merkle tree;
3) When issuing a credential, the issuers:
   a) Create digital credentials and sign the Merkle tree roots using BLS;
   b) Record the issuance proof, i.e. the roots and signatures.
4) The holder receives the credentials and stores them in their digital wallet;
5) The verifier requests presentation with specified claims;
6) The holder combines their credentials into a presentation and sends it to a verifier as follows:
   a) They reveal claim values that verifier requested along with their salts; for the other values, they prepare the corresponding proofs generated using the Merkle tree. The presentation now consists of all the elements needed to recreate the roots of the Merkle trees;
   b) The signatures of the roots of the trees of individual credentials are aggregated into one presentation signature, together with the holder's signature. This single signature is used to verify the validity of the credentials issuers and holders;
   c) Holder sends a presentation to the verifier.
7) When receiving a presentation, the verifier does the following:
   a) Recreates the Merkle trees using the registered formats, disclosed values, salts and the resulting hashes, thus obtaining the trees' roots;
   b) Verifies the roots using the public registry (thus validating the credentials' issuance). They verify the aggregated issuers' signature of the credentials and the holder's signature using recorded public keys.

In this use case, BLS signatures are crucial because they allow the aggregation of signatures; that is, they confirm that the presentation was created based on multiple credentials.

3) USE CASE 3: SELECTIVE DISCLOSURE OF CLAIMS/PROVING CLAIMS WITHOUT REVEALING THEM

The third use case extends the previous ones offering range proofs. This feature is crucial in preserving privacy and data minimization. It allows hiding and proving the values instead of revealing and showing them. When implemented, this use case allows users not to reveal their birth date or salary amount but to prove they are in the required range. As part of this solution, the key elements are the Pedersen commitments and the ZKP tool Bulletproofs. This tool allows proving that a value belongs to a range, that a certain arithmetic expression is valid, that a value belongs to a set, and so on. The steps of the third use case (shown in Figure 12) are as follows:

1) Entities register their public keys in a public registry;
2) In addition to the public key, the issuer registers the credential format to enable the correct generation of the Merkle tree;
3) When issuing a credential, the issuer:
   a) Creates a digital credential and signs the Merkle tree root using BLS;
   b) Records the issuance proof, i.e. its root and signature.
4) The holder receives the credential and stores it in their digital wallet;
5) The verifier requests presentation with specified claims;
6) The holder sends their credential as a presentation to the verifier as follows:
   a) They reveal claim values that verifier requested along with their salts; for the other values, they prepare the corresponding proofs generated using the Merkle tree. For values that the holder does not want to reveal but has to prove, they generate a Bulletproof range proof, where they can show that the value belongs to a required range. The presentation now consists of all the elements needed to recreate the root of the Merkle tree;
   b) They sign the root and aggregate their signature with the issuer's signature to create a unique presentation signature;
   c) They send the presentation to the verifier.
7) When receiving a presentation, the verifier does the following:
   a) Check the range proof validity. Recreates the Merkle tree using the registered format, disclosed values, salts and the resulting hashes, thus obtaining the tree's root;
   b) Verifies the root using the public registry (thus validating the credential issuance). They verify

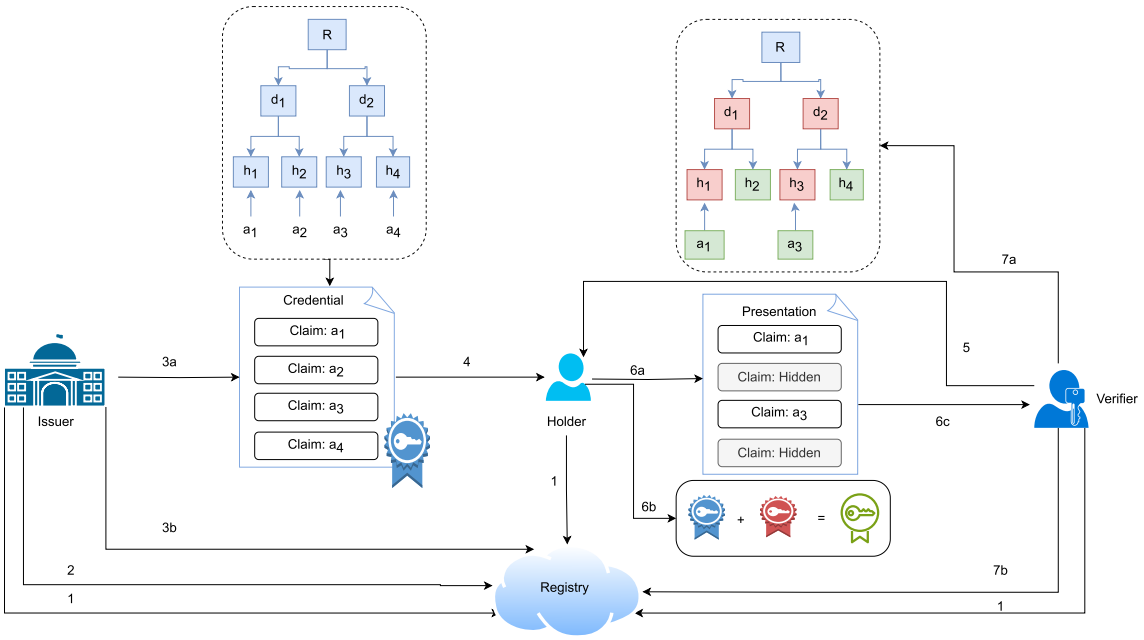


FIGURE 10. Use case 1: Selective disclosure of claims from a single credential.

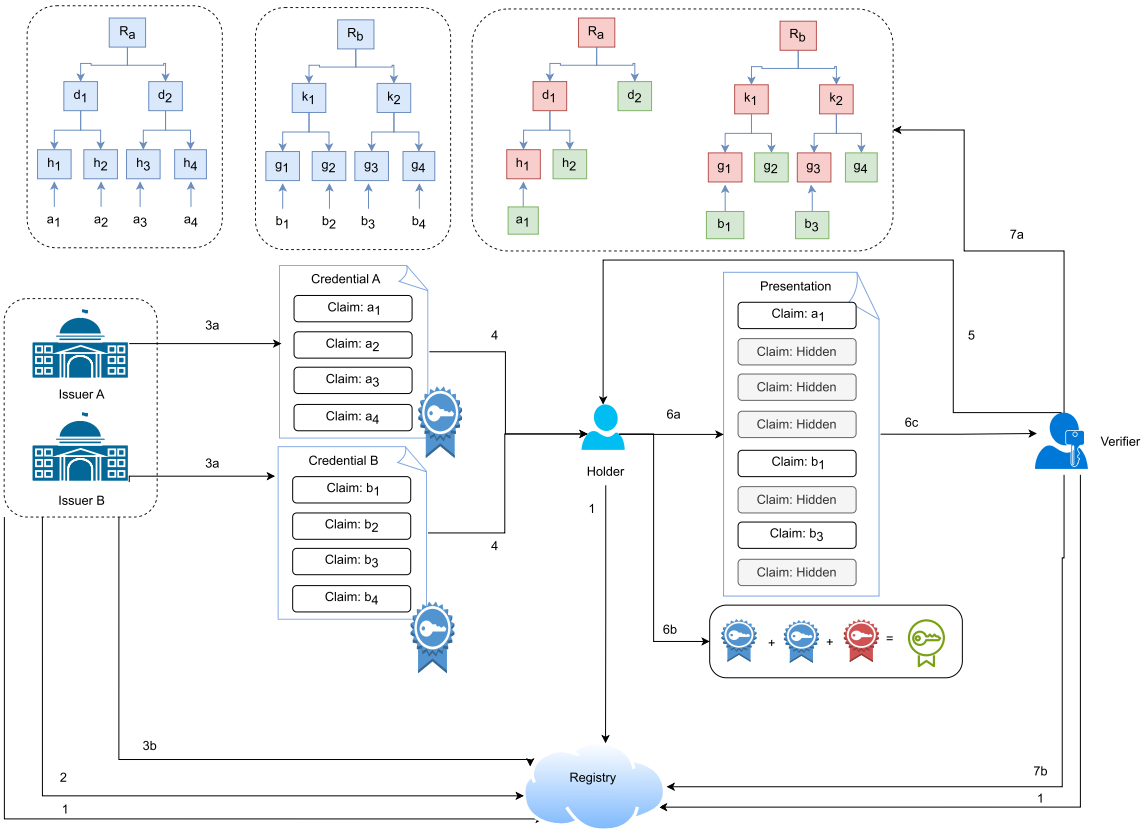


FIGURE 11. Use case 2: Selective disclosure of claims from multiple credentials.

the issuer's signature of the credential, as well as the holder's signature, using recorded public keys.

In this use case, the critical element is Pedersen's homomorphic commitment and Bulletproofs.

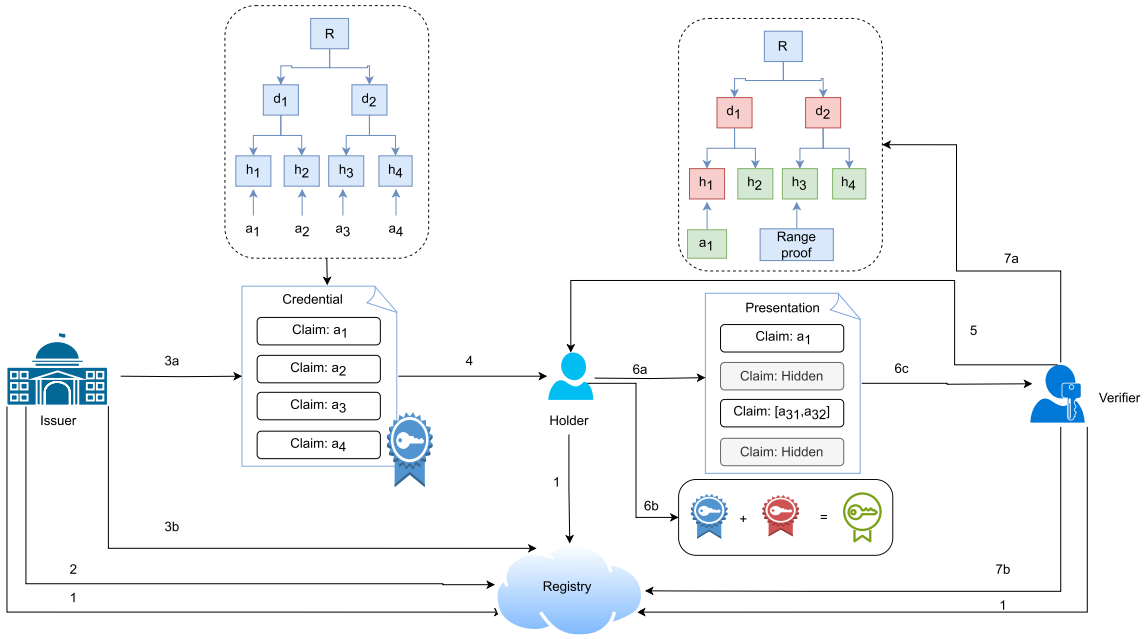


FIGURE 12. Use case 3: Selective disclosure of claims/proving claims without revealing them.

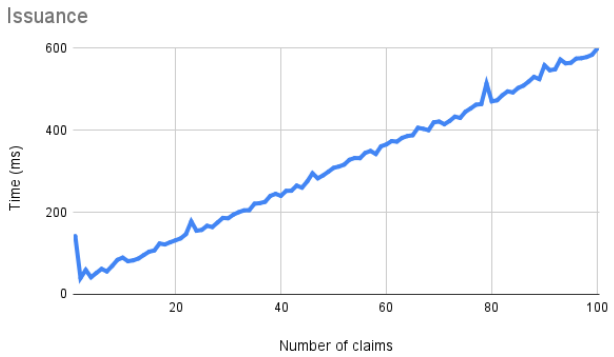


FIGURE 13. Credential issuance.

V. TIME MEASUREMENTS

The proof of concept code of this paper is publicly available on Github. The proof of concept is modularly implemented in JavaScript with separate functions that can be used in plug-and-play manner. To demonstrate how small are the device requirements for this approach, time measurement was conducted on Thinkpad T470 Laptop with Intel® Core™ i5-6300U CPU. The solution's efficiency in issuing many claims, ranging from 1 to 100 with randomized values, is demonstrated and quantified in Figure 13. This performance metric clearly explains the solution's capability to manage a large volume of data in short time.

More than 100 claims in a single credential are unmanageable by a holder and should be avoided. It is clear that as the number of claims grows, the time needed to issue them grows, but it is still unnoticeable from a user's perspective.

Proof generation is also measured using a credential containing 100 claims where 1 to 50 were disclosed. It is split into two categories: selective showing of textual claims and selective disclosure of numbers using range proof as shown in Figure 14. Selective disclosure using range proof is slower than a simple showing of the attribute, but it can still be expressed in seconds for a small number of range proofs.

Verifying is also split into the same two categories as shown in Figure 15, where the time needed to verify range proof is longer than for verification of showed claim. Aggregation of credentials into one and verifying them has only added an element of BLS aggregation, which doesn't affect the time in a significant way.

VI. DISCUSSION

The presented solution meets the set requirements and enables different usage scenarios for selective disclosure. It uses:

- **Merkle trees** that allow unlinkability (through hashing and salts), value hiding, and validation of credentials;
- **BLS signatures** that allow verification of issuance, verification of the credentials' holder, combining multiple credentials into one presentation, and the possibility of multiple signatures on one credential;
- **Pedersen Commitments and Bulletproofs** that allow proving values without revealing them in order to promote minimization and overall selective disclosure.

A. COMPARISON TO OTHER APPROACHES

This solution belongs to a new category, and considering that no solution belonging to the same category exists at the time of writing, comparisons from the aspect of work and

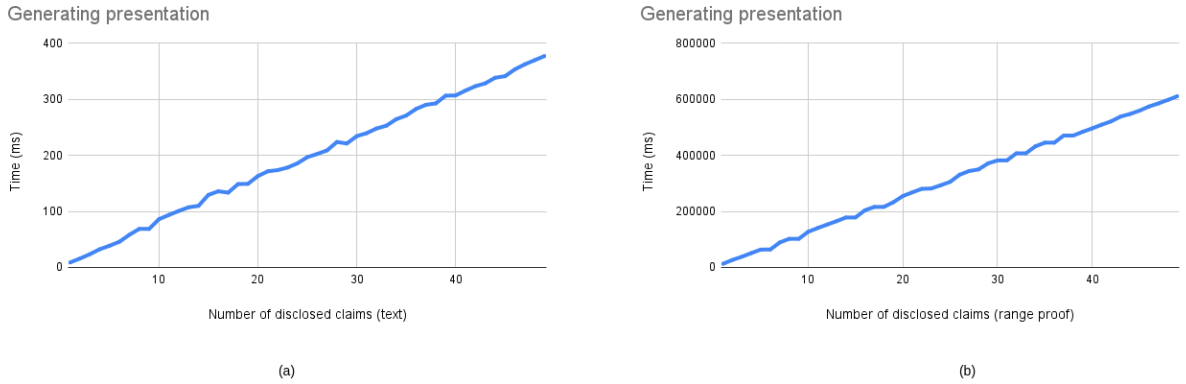


FIGURE 14. Generating presentation: (a) Text claims; (b) Range proofs.

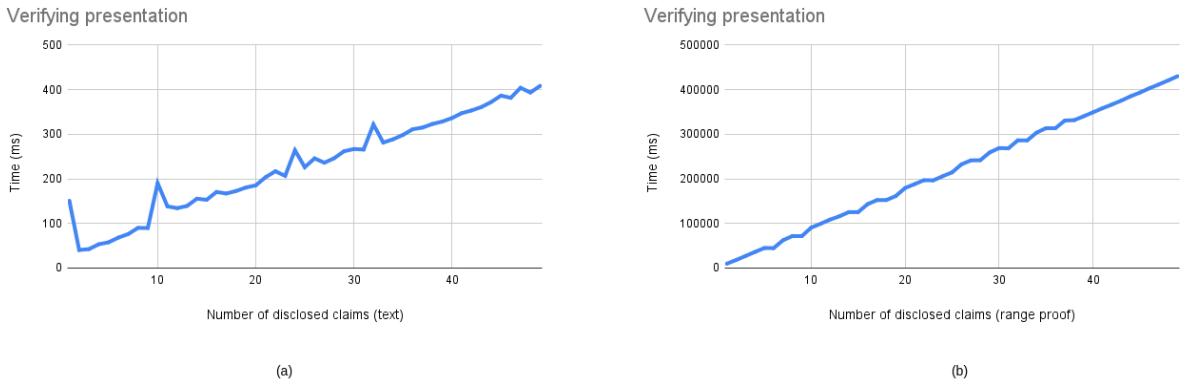


FIGURE 15. Verifying presentation: (a) Text claims; (b) Range proofs.

performance are difficult to make. Nonetheless, in order to present the benefits of this approach, a comparison to existing combined approaches needs to be made (comparing with a single method approach is unnecessary due to additional functionalities achieved through combination):

- Compared to the combination of hash- and signature-based methods, presented approach enables proving the value of an attribute without revealing it. It also does not require canonicalization algorithms because a single message is signed. Compared to:
  - -- [49], presented approach enables combining credentials into a single presentation (req. 2 and 3), proving the value of an attribute without revealing it (req. 12), proving that the identity owner is the subject of the credential (req. 8 and 9);
  - -- [48], presented approach enables combining credentials into a single presentation (req. 2 and 3), proving the value of an attribute without revealing it (req. 12), proving that the identity owner is the subject of the credential (req. 8 and 9), uses Merkle tree proofs for faster verification (approach in [48] reconstructs the tree using first layer of tree instead of parent nodes when required).
- In comparison to the hash- and ZKP-based methods [54], presented approach allows verification of the issuer and identity owner and does not require a trusted setup that is necessary for existing solutions based on zk-SNARKS and proof size is smaller. Compared to [54], presented approach does not require sending each credential as a separate presentation while using the same salts for attributes (req. 2 and 3).
- Compared to the signature and ZKP-based methods, it enables explicit credential validation and combining credentials and does not require canonicalization algorithms or a trusted setup. Compared to
  - -- [51], presented approach enables combining credentials from different issuers into a single presentation (req. 3), proving the property of an attribute without revealing it (req. 12);
  - -- [50], presented approach enables combining credentials from different issuers or issuers groups into a single presentation (req. 3), proving the property of an attribute without revealing it (req. 12);
  - -- [52], presented approach enables combining credentials from different issuers or issuers groups into a single presentation (req. 3), proving the property

of an attribute without revealing it (predicates) (req. 12). Paper [52] is more focused on access control attributes (rules for access control) compared to regular credentials;
-- [53], presented approach enables combining credentials into a single presentation (req. 2 and 3), proving the property of an attribute without revealing it (req. 12). The paper [53] focuses on issuer-hiding using ZKP, which contradicts the transparency requirements of digital identity systems in terms of widespread adoption (legal and regulatory).

B. PERFORMANCE ANALYSIS

A brief explanation of performance using $\mathcal{O}$ or defined times within the standard, for individual elements is as follows [41], [55], [56]:

- Merkle trees:
  - -- Space - $\mathcal{O}(n)$
  - -- Searching - $\mathcal{O}(\log n)$
  - -- Traversal - $\mathcal{O}(n)$
  - -- Insertion - $\mathcal{O}(\log n)$
  - -- Deletion - $\mathcal{O}(\log n)$
  - -- Synchronization - $\mathcal{O}(\log n)$
  - -- Proof - $\mathcal{O}(\log_2 n)$
- BLS signatures:
  - -- Private key - 32 bytes
  - -- Public key - 48 bytes
  - -- Signature - 96 bytes
  - -- Aggregated signature - 96 bytes
  - -- Signing - 370μ s
  - -- Verifying - 2700μ s
- Bulletproofs:
  - -- Proof generation - $\mathcal{O}(n)$
  - -- Proof verification - $\mathcal{O}(n)$
  - -- Proof size - $3 \log 2n + 9$ elements

Due to the sequential nature of the proposed approach, elements such as credential issuance, presentation generation, and presentation verification are complex as the most complex element in them, e.q. Merkle tree generation is the most complex for credential issuance, proof generation and verification from Bulletproofs are the most complex for presentation generation and verification. They are as follows for the three algorithms:

- Credential issuance - $\mathcal{O}(n \log k)$ - based on generating Merkle tree and Pedersen commitments of attributes that depend on the number of bits k of each attribute value;
- Presentation generation - $\mathcal{O}(m*n)$ - based on generating Bulletproofs range proofs for m attributes;
- Presentation verification - $\mathcal{O}(m*n)$ - based on verifying range proofs for m attributes.

In case of combining credentials, the last two complexities are multiplied by the number of credentials.

The implemented approach demonstrates not only its usability in practical scenarios and efficient time execution but also its practical application in any format of digital credentials, particularly verifiable credentials. It should be noted that the solution is written in JavaScript and that the time measured depends on the language and the device used. Still, when implemented in the language commonly used for web and mobile applications, the measured time shows that this solution is feasible and viable for practical implementation.

C. SECURITY AND THREAT ANALYSIS

As this approach uses three primitives in a sequential manner, the proof of security and threats directly depend on the security assumptions and threats of the primitives.

In their basic form, Merkle trees can be attacked in two ways. The first way is if salt is not used. This enables an attack by a rainbow table (tables of already known hashed values). Correlation attacks can also be prevented if different salts are used for different values. The root does not reveal the depth of the tree, so Merkle trees are vulnerable to second-preimage attacks (allowing the creation of a Merkle tree with the same root). In order to prevent this attack, it is recommended to enable certificate transparency (adding bits to hashed data and internal nodes) or to limit the tree size [57], [58].

The primary advantage of BLS signatures is the possibility of aggregation. In the original BLS scheme, aggregation was vulnerable to rogue public-key attacks. To avoid this, it is recommended to use proof of knowledge of the secret key (KOSK) or unique messages. Boneh, Drijvers, and Neven presented a modified implementation in their work [59], which is not susceptible to this attack.

Bulletproofs, as presented in the original paper, are susceptible to the Frozen Heart attack. In the case of an insecure protocol, this attack allows falsifying proofs that will still be successfully verified. The original paper uses the Fiat-Shamir transformation to make the proof fully non-interactive. However, this implementation omits a crucial component. To prevent this attack, adding a Pedersen commitment to the Fiat-Shamir transformation hash is sufficient. This prevents proof falsification [60].

Another threat to this approach is a man-in-the-middle attack on presentations. A malicious party could intercept and reuse a presentation, potentially spoofing the owner's identity. To mitigate this attack, it is necessary to have nonces, session-based presentation keys, or challenge-response protocols. All of this makes intercepted data unusable in subsequent sessions. Implementations of mentioned elements depends on the system where this approach will be used.

When designing and implementing a protocol that uses presented approach for selective disclosure, the proposed corrections must be applied to ensure adequate security.

D. LIMITATIONS

The presented selective disclosure approach satisfies the defined requirements but has certain limitations.

One requirement that represents a limitation of the system is unlinkability and collusion resistance. Unlinkability prevents linking presentations of the same user without their permission.The following unlinkability types exist [9]:

- Unlinkability of presentations: The verifier cannot link two presentations of the same credential;
- Unlinkability of verifiers: Two colluding verifiers should not be able to learn that they have received presentations of the same credential;
- Issuer and verifier unlinkability (honest verifier): The issuer must not be able to know that the credential was sent to a verifier;
- Issuer and verifier unlinkability (compromised verifier): The issuer must not know that the credential was sent to the verifier, even if the verifier tries colluding with the issuer.

In all these situations, unlinkability is limited to use cases when the credential does not contain information that directly or indirectly identifies the user, such as a unique ID number or tax number. This requirement contradicts the creation of an open and transparent digital identity system.

We will consider the mentioned types of unlinkability from the aspect of the proposed approach:

- Unlinkability of presentation and unlinkability of verifiers: By using Pedersen binding values, which are homomorphic, it is possible to prove that a value exists within a credential without revealing it. This does not allow the verifier to know the details of the Merkle tree. For credentials created by the identity owner, they can create a presentation with different salts when creating the tree. In addition, the proposed approach is: When issuing a credential, multiple versions are generated, which differ by salts and, thus, by resultant roots. When an identity owner sends a presentation, a different version is randomly sent each time. In this way, the verifier/s is/are never able to recreate the correct Merkle tree;
- Unlinkability of issuers and honest verifiers is not achieved through this approach. However, there is a potential solution to the problem. This solution is of a systemic and regulatory nature. There must be legal frameworks that regulate the honesty and correctness of issuers, as well as properly defined procedures;
- Unlinkability of issuers and compromised verifiers is difficult to achieve through the presented approach due to the use of a salt prepared by the issuer. A potential way to prevent collusion is to use the mentioned homomorphism. The identity owner can create appropriate commitments using their salts for attributes whose values can be proven to the issuer. By using ''out-of-the-box'' commitments, the issuer continues with the credential issuing process, and in case of collusion cannot reveal the attributes without knowing salts.

Another limitation of the approach is its use in existing or new identity systems. In order to use the approach correctly, it is necessary to:

- Fulfill the recommendations of the implementation of individual elements in order to ensure the security of the system;
- Fulfill the recommendations of the implementation of individual elements to ensure the performance of the system;
- Choose technology and individual elements to achieve the best possible performance.

This approach was created in an agnostic manner, meaning it can be used with different credential types: verifiable credentials, attribute-based credentials, and anonymous credentials. It can also be used with different identity models, from centralized and federated to decentralized and self-sovereign identity models. In the case of the centralized and federated identity model, the recording of the credential issuance in the public registry refers to the recording of the issuance in the corresponding centralized database. The step in which the verifier validates and verifies the presentation is also done through communication with the issuer (without reliance on a public data registry).

The performance and security of this approach depend on the implementation of the entire system, which means that this approach, implemented within the system, increases in complexity. Depending on the identity model used, key management, recording of credentials issuance, and verification processes can change. For example, in centralized or federated models, PKI (public key infrastructure) can be used, while in decentralized models, DIDs (decentralized identifiers) and DID documents can be used.

Integrating this approach with existing identity models is necessary for its widespread adoption. It is necessary to define how this approach can be layered onto existing identity infrastructures without disruption. Key elements that need to be considered are scalability, trust management and user experience. Optimizing each element of the presented approach will allow for usage in high-volume environments. Using standardized cryptographic primitives and implementation will allow regulatory compliance in centralized and federated models. Usability and user interaction for this approach are also key to the adaptability of the approach and one of the challenges in practical implementation. User experience managing the credentials and creating presentations must be considered when implementing the approach.

This approach requires standardizing certain elements. It is necessary to define the format of required claims, how the proofs will be added to verifiable presentations, and how the revocation of issued credentials can be done. Each of these elements can affect the widespread adoption of this approach. Even though the presented approach fulfills the requirements, standardization efforts may face different challenges. These challenges include interoperability across implementation, which depends on the exact implementation of each element

used, and various regulations, which are different depending on the jurisdiction and evolving landscape, where techniques and digital identity standards are rapidly evolving. Each implementation of this approach should be interoperable, and this can be achieved by defining precise specifications and reference and test implementations. This approach is modular and adaptable to address regulatory differences in case of a requirement for specific cryptographic primitives. To achieve proper standardization, it is necessary to update this approach to be compatible with emerging standards such as W3C VCs and DIDs.

E. FUTURE WORK AND POTENTIAL IMPROVEMENTS

Using this approach, it is possible to change some primitives to achieve adequate security. Moreover, one of the future challenges is to verify and compare the behaviour of similar algorithms in the approach or to improve the algorithms to those that can be executed in the post-quantum world. For example, it is possible to replace Bulletproofs with some zk-STARK algorithm. Both variants do not require a trusted setup. Although zk-STARK is not widely used in this field [47], current performance and the existence of post-quantum tools would allow new elements. In addition, Pedersen binding values can be replaced with the Poseidon hash [61], which has been proven adequate for zk-STARK tools. In contrast, BLS signatures can be replaced with other aggregateable signatures, such as PQScale [62] - FALCON [63]. In order to reduce proofs in presentation, it is also necessary to consider aggregating Merkle tree proofs.

VII. CONCLUSION

In this paper, we presented a solution for the selective disclosure of claims in digital credentials. By combining Merkle hash trees with BLS signatures and the ZKP method Bulletproof, we fulfilled the requirements of selective disclosure. This solution allows for:

- Generation of credential with multiple issuers, which has only one aggregated signature;
- Generation of credential that both issuer and holder signs, which has only one aggregated signature;
- Selectively disclosing claims from a single credential, while ensuring unlinkability and maintaining the verifiable pairing between the holder and the credential;
- Selectively disclosing claims from multiple credentials, combining them into one presentation, while ensuring unlinkability and maintaining the verifiable pairing between the holder and the credential;
- Generating small range proofs for values that should not be disclosed and verifying them.

The solution presented belongs to a category of combined approaches for selective disclosure: ZKP, hash- and signature-based. The solution is usable and practical in real-life scenarios. Future work will include creating verifiable credentials using this approach for a self-sovereign identity system. It should also consider other ZKP tools that can be used and the possibility of a post-quantum solution that uses this approach but different methods. New approaches to selective disclosure are needed as the rapid development of digital identity systems demands enhanced privacy and security mechanisms to keep pace with evolving technological and user needs.

REFERENCES

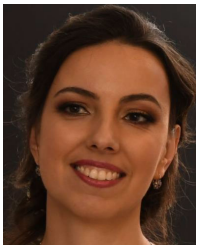
[1] Š. B. Ramic, E. Cogo, I. Prazina, E. Cogo, M. Turkanovic, R. T. Mulahasanovic, and S. Mrdovic, "Selective disclosure in digital credentials: A review," *ICT Exp.*, vol. 10, no. 4, pp. 916–934, Aug. 2024.
[2] J. Sedlmeir, R. Smethurst, A. Rieger, and G. Fridgen, "Digital identities and verifiable credentials," *Bus. Inf. Syst. Eng.*, vol. 63, pp. 603–613, Apr. 2021.
[3] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Commun. ACM*, vol. 28, no. 10, pp. 1030–1044, Oct. 1985.
[4] F. Baldimtsi and A. Lysyanskaya, "Anonymous credentials light," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. CCS*, 2013, pp. 1087–1098.
[5] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *Proc. Int. Conf. Theory Appl. Cryptograph*, Jan. 2001, pp. 93–118.
[6] J. Camenisch and E. Van Herreweghen, "Design and implementation of the idemix anonymous credential system," in *Proc. 9th ACM Conf. Comput. Commun. Secur.*, Nov. 2002, pp. 21–30.
[7] U. Kenig. (2011). *Identity Mixer*. Accessed: Aug. 2, 2024. [Online]. Available: http://primelife.ercim.eu/results/opensource/55-identity-mixer
[8] C. Paquin. *U-prove Technology Overview V1.1 (revision 2)*. Accessed: Aug. 2, 2024. [Online]. Available: https://www.microsoft.com/en-us/research/publication/u-prove-technology-overview-v1-1-revision-2/
[9] D. Fett, K. Yasuda, and B. Campbell. (2024). *Selective Disclosure for JWTs (SD-JWT)*. Accessed: Aug. 2, 2024. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt/08/
[10] (2019). *Hyperledger*. Accessed: Aug. 2, 2024. [Online]. Available: https://www.hyperledger.org/
[11] S. Elfors and A. Burckard. (2023). *Etsitr 119 476 V1.1.1: Electronic Signatures and Infrastructures (esi)-analysis of Selective Disclosure and Zero-knowledge Proofs Applied To Electronic Attestation of Attributes*. Accessed: Aug. 2, 2024. [Online]. Available: https://www.etsi.org/deliver/etsi
[12] D. Yamamoto, Y. Suga, and K. Sako, "Formalising linked-data based verifiable credentials for selective disclosure," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops*, Jun. 2022, pp. 52–65.
[13] P. Voigt and A. V. D. Bussche, "The EU general data protection regulation (GDPR)," in *A Practical Guide*, 1st ed., Cham, Switzerland: Springer, 2017.
[14] (2018). *California Consumer Privacy Act (CCPA)*. Accessed: Aug. 2, 2024. [Online]. Available: https://leginfo.legislature.ca.gov/faces/codes_displayText
[15] (2018). *[Support] Selective Disclosure*. Accessed: Aug. 2, 2024. [Online]. Available: https://www.privacypatterns.org/patterns/Support-Selective-Disclosure
[16] (2023). *Selective Disclosure Guide: Privacy Feature of Verifiable Credentials*. Accessed: Aug. 2, 2024. [Online]. Available: https://www.dock.io/post/selective-disclosure
[17] (2021). *Regulation of the European Parliament and of the Council Amending Regulation as Regards Establishing a Framework for a European Digital Identity*. Accessed: Aug. 2, 2024. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex
[18] (2023). *Selective Disclosure: An Ebsi Improvement Proposal*. Accessed: Aug. 2, 2024. [Online]. Available: https://ec.europa.eu/digital-building-blocks/sites/display/EBSI/Selective
[19] F. Ullah, C.-M. Pun, O. Kaiwartya, A. S. Sadiq, J. Lloret, and M. Ali, "HIDE-healthcare IoT data trust ManagEment: Attribute centric intelligent privacy approach," *Future Gener. Comput. Syst.*, vol. 148, pp. 326–341, Nov. 2023.
[20] *Verifiable Credentials Data Model V2.0*. Accessed: Aug. 2, 2024. [Online]. Available: https://www.w3.org/TR/vc-data-model-2.0/
[21] *Verifiable Credentials Implementation Guidelines 1.0*. Accessed: Aug. 2, 2024. [Online]. Available: https://www.w3.org/TR/vc-imp-guide/

[22] A. Tariq, H. B. Haq, and S. T. Ali, "Cerberus: A blockchain-based accreditation and degree verification system," *IEEE Trans. Computat. Social Syst.*, vol. 3, no. 2, pp. 1–10, May 2022.
[23] Z. Yang, H. Ma, M. Ai, M. Zhan, G. Wu, and Y. Zhang, "A minimal disclosure signature authentication scheme based on consortium blockchain," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Aug. 2022, pp. 516–521.
[24] R. Tian, L. Kong, B. Zhang, X. Li, and Q. Li, "Authenticated selective disclosure of credentials in hybrid-storage blockchain," in *Proc. IEEE 28th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Jan. 2023, pp. 330–337.
[25] A. Squicciarini, E. Bertino, E. Ferrari, F. Paci, and B. Thuraisingham, "PP-trust-X: A system for privacy preserving trust negotiations," *ACM Trans. Inf. Syst. Secur.*, vol. 10, no. 3, p. 12, Jul. 2007.
[26] A. C. Squicciarini, A. Trombetta, E. Bertino, and S. Braghin, "Identity-based long running negotiations," in *Proc. 4th ACM Workshop Digit. Identity Manage.*, Oct. 2008, pp. 97–106.
[27] A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in *16th Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Jan. 2010, pp. 177–194.
[28] D. W. Kravitz, "Exploration and impact of blockchain-enabled adaptive non-binary trust models," in *Proc. 20th Int. Conf. Distrib. Comput. Netw.*, Jan. 2019, pp. 124–133.
[29] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," in *Proc. 22nd Annu. Int. Cryptol. Conf.*, Jan. 2002, pp. 61–76.
[30] J. Camenisch and A. Lysyanskaya, "A signature scheme with efficient protocols," in *Proc. 3rd International Conference*, 2003, pp. 268–289.
[31] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Proc. Annu. Int. Cryptol. Conf.*, Jan. 2004, pp. 56–72.
[32] J. Camenisch and T. Groß, "Efficient attributes for anonymous credentials," *ACM Trans. Inf. Syst. Secur.*, vol. 15, no. 1, pp. 1–30, Mar. 2012.
[33] M. Schanzenbach, T. Kilian, J. Schutte, and C. Banse, "ZKlaims: Privacy-preserving attribute-based credentials using non-interactive zero-knowledge techniques," in *Proc. 16th Int. Joint Conf. e-Business Telecommun.*, 2019, pp. 325–332.
[34] J. Lee, J. Choi, H. Oh, and J. Kim, "Privacy-preserving identity management system," *IACR Cryptol. ePrint Arch.*, vol. 2021, p. 1459, Jan. 2021.
[35] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Jan. 2001, pp. 514–532.
[36] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Jan. 2003, pp. 416–432.
[37] R. C. Merkle, "Method of providing digital signatures," U.S. Patent 4 569 309, Jun. 1, 1982.
[38] B. Bunz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 315–334.
[39] J. Groth, "Linear algebra with sub-linear zero-knowledge arguments," in *Proc. Annu. Int. Cryptol. Conf.*, 2009, pp. 192–208.
[40] J. Bootle, A. Cerulli, P. Chaidos, J. Groth, and C. Petit, "Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting," in *Proc. 35th Annu. Int. Conf. Theory Appl. Cryptograph.*, Jan. 2016, pp. 327–357.
[41] *Bulletproofs and Mimblewimble*. Accessed: Aug. 2, 2024. [Online]. Available: https://tlu.tarilabs.com/cryptography/bulletproofs-and-mimblewimble
[42] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Proc. Annu. Int. Cryptol. Conf.*, 1991, pp. 129–140.
[43] A. De Salve, A. Lisi, P. Mori, and L. Ricci, "Selective disclosure in self-sovereign identity based on hashed values," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2022, pp. 1–8.
[44] A. Rondelet. (2020). *A Note on Anonymous Credentials Using BLS Signatures*. [Online]. Available: https://arxiv.org/pdf/2006.05201
[45] A. Yoganand Athavale, K. Singh, and S. Sood, "Design of a private credentials scheme based on elliptic curve cryptography," in *Proc. 1st Int. Conf. Comput. Intell., Commun. Syst. Netw.*, Jul. 2009, pp. 332–335.
[46] I. Sene, A. A. Ciss, and O. Niang, "I2PA: An efficient ABC for IoT," *Cryptography*, vol. 3, no. 2, p. 16, Jun. 2019.
[47] B. O. Roelink, M. El-Hajj, and D. K. Sarmah, "Systematic review: Comparing zk-SNARK, zk-STARK, and bulletproof protocols for privacy-preserving authentication," *Secur. Privacy*, vol. 7, no. 5, p. 401, Apr. 2024.
[48] R. Mukta, J. Martens, H.-y. Paik, Q. Lu, and S. S. Kanhere, "Blockchain-based verifiable credential sharing with selective disclosure," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 959–966.
[49] Z. Li, "A verifiable credentials system with privacy-preserving based on blockchain," *J. Inf. Secur.*, vol. 13, no. 2, pp. 43–65, 2022.
[50] A. Sonnino, M. Al-Bassam, S. Bano, S. Meiklejohn, and G. Danezis. (2020). *Coconut: Threshold Issuance Selective Disclosure Credentials With Applications To Distributed Ledgers*. [Online]. Available: https://arxiv.org/pdf/1802.07344
[51] J. García-Rodríguez, R. T. Moreno, J. B. Bernabe, and A. Skarmeta, "Implementation and evaluation of a privacy-preserving distributed ABC scheme based on multi-signatures," *J. Inf. Secur. Appl.*, vol. 62, Nov. 2021, Art. no. 102971.
[52] M. Kohlweiss and A. Rial, "Optimally private access control," in *Proc. 12th ACM Workshop Workshop Privacy Electron. Soc.*, Nov. 2013, pp. 37–48.
[53] A. Connolly, P. Lafourcade, and O. P. Kempner, "Improved constructions of anonymous credentials from structure-preserving signatures on equivalence classes," in *Public-Key Cryptography—PKC 2022: 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8–11, 2022, Proceedings, Part I*, 2022, pp. 409–438, doi: 10.1007/978-3-030-97121-2_15.
[54] M. Babel and J. Sedlmeir. (2023). *Bringing Data Minimization To Digital Wallets at Scale With General-Purpose Zero-Knowledge Proofs*. [Online]. Available: https://arxiv.org/pdf/2301.00823
[55] G. Becker. (2008). *Merkle Signature Schemes, Merkle Trees and Their Cryptanalysis*. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d7c3aa65bc5df32d94dcc8b29dceca240bdf8bef
[56] D. Boneh, S. Gorbunov, R. S. Wahby, H. Wee, C. A. Wood, and Z. Zhang. (2022). *Bls Signatures*. Accessed: Aug. 2, 2024. [Online]. Available: https://www.ietf.org/archive/id/draft-irtf-cfrg-bls-signature-05.html
[57] E. Andreeva, C. Bouillaguet, O. Dunkelman, and J. Kelsey, "Herding, second preimage and trojan message attacks beyond merkle-damg," in *Proc. 16th Annu. Int. Workshop*, 2009, pp. 393–414.
[58] B. Laurie, A. Langley, and E. Kasper, *Certificate Transparency*, document RFC 6962, 2013.
[59] D. Boneh, M. Drijvers, and G. Neven, "Compact multi-signatures for smaller blockchains," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Jan. 2018, pp. 435–464.
[60] (2022). *The Frozen Heart Vulnerability in Bulletproofs*. [Online]. Available: https://blog.trailofbits.com/2022/04/15/the-frozen-heart-vulnerability-in-bulletproofs/
[61] L. Grassi, D. Khovratovich, C. Rechberger, A. Roy, and M. Schofnegger, "Poseidon: A new hash function for zero-knowledge proof systems," in *Proc. 30th USENIX Secur. Symp. (USENIX Secur. 21)*, Jan. 2021, pp. 519–535.
[62] J.-H. Hsiang, S. Fu, P.-C. Kuo, and C.-M. Cheng. (2023). *Pqscale: A Post-Quantum Signature Aggregation Algorithm*. Accessed: Nov. 10, 2024. [Online]. Available: https://uploads-ssl.webflow.com/642374103c1677f8f335c581/64771752dbe6933ceb1d712b_PQScale.pdf
[63] G. Alagic, D. Cooper, Q. Dang, T. Dang, J. M. Kelsey, J. Lichtinger, Y.-K. Liu, C. A. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, D. Smith-Tone, and D. Apon. (2022). *Status Report on the Third Round of the Nist Post-Quantum Cryptography Standardization Process*. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413.pdf

ŠEILA BEĆIROVIĆ RAMIĆ (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees from the Department of Computer Science and Informatics, Faculty of Electrical Engineering, University of Sarajevo, Bosnia and Herzegovina, in 2017 and 2019, respectively, where she is currently pursuing the Ph.D. degree. She is currently a Senior Teaching Assistant with the Department of Computer Science and Informatics, Faculty of Electrical Engineering, University of Sarajevo. Her research interests include computer networks and security, self-sovereign identity, and privacy.

IRFAN PRAZINA received the B.Sc. and M.Sc. degrees from the Department of Computer Science and Informatics, Faculty of Electrical Engineering, University of Sarajevo, Bosnia and Herzegovina, in 2013 and 2015, respectively, where he is currently pursuing the Ph.D. degree. He is currently a Senior Teaching Assistant with the Department of Computer Science and Informatics, Faculty of Electrical Engineering, University of Sarajevo. His research interests include web technologies, software testing, and mobile application development.

DAMIR POZDERAC received the B.Sc. and M.Sc. degrees from the Department of Computer Science and Informatics, Faculty of Electrical Engineering, University of Sarajevo, Bosnia and Herzegovina, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Informatics, Faculty of Electrical Engineering. He is currently a Teaching Assistant with the Department of Computer Science and Informatics, Faculty of Electrical Engineering, University of Sarajevo. His research interests include web technologies, computer vision, and information retrieval.



RAZIJA TURČINHODŽIĆ MULAHASANOVIĆ (Member, IEEE) received the Ph.D. degree from the Department of Computing and Informatics, in 2015. She is currently an Associate Professor with the Faculty of Electrical Engineering, University of Sarajevo, Bosnia and Herzegovina. During her work at the faculty, she taught courses in the field of programming, applied mathematics in computing and informatics, databases, information systems, computer graphics, computer networks, and simulations. Her current research interests include computer simulations, computer networks and security, applied mathematics in computer science, cryptography, and automated timetabling problems.

SAŠA MRDOVIĆ received the Ph.D. degree in intrusion detection systems from the Department of Computing and Informatics, in 2009. He is currently a Professor with the Faculty of Electrical Engineering, University of Sarajevo. He teaches computer networks and security courses. He has published three books on networks and security and a number of papers in scientific journals and conference proceedings. He has been reviewing papers for various journals and conferences. He also works on projects with industry and government in the area of information security. His main research interests include digital information security, digital forensics, and the IoT. He holds a CISSP, security industry certificate.