User-Centric and Privacy-Preserving Attribute-Based Authentication in Healthcare Systems Leveraging zk-SNARKs and Soulbound Tokens

Biagio Boi
dept. of Computer Science
University of Salerno
Fisciano, Salerno, Italy
bboi@unisa.it

Franco Cirillo
dept. of Computer Science
University of Salerno
Fisciano, Salerno, Italy
fracirillo@unisa.it

Marco De Santis
dept. of Computer Science
University of Salerno
Fisciano, Salerno, Italy
mdesantis@unisa.it

Christian Esposito
dept. of Computer Science
University of Salerno
Fisciano, Salerno, Italy
esposito@unisa.it

***Abstract*—Digital health services for disease diagnosis, follow-up, and patient empowerment manage data that belongs to a special class of personal information, according to the General Data Protection Regulation (GDPR). For this reason, user authentication and access control are among the key security measures suggested for their protection. However, in the medical context, it is crucial to balance security and privacy support with timeliness and ease of access, which requires innovative solutions.**

This manuscript introduces an innovative approach leveraging Soulbound Tokens (SBTs) and Zero-Knowledge Proofs (ZKPs), particularly zk-SNARKs, to provide a privacy-aware mechanism for patient authentication in the medical domain. SBTs are utilized within an Attribute-Based Access Control (ABAC) model, ensuring that only eligible patients can access specific medical treatments. In a treatment-specific model, an SBT is issued for each diagnosis, allowing precise control but increasing management complexity. Alternatively, in a diagnosis-category-based model, SBTs are grouped by diagnostic categories. This reduces the number of tokens and optimizes the space in the patient's wallet but sacrifices some precision in the information. Results demonstrate the timeliness of the proposed approach, with an average time of 6.82s for the release of an SBT and a maximum on-chain verification time of 15.04ms, showcasing their future adoption in a real-time environment, such as the medical context.

***Index Terms*—SBT, ZKP, ABAC, zk-SNARKs, SSI**

I. Introduction

Worldwide data protection legislation recognizes authentication and authorization as critical means for ensuring the security and privacy of personal information [1]. Authentication verifies a user's identity, granting access to protected systems and resources, while authorization determines which resources can be accessed based on predefined permissions. Traditional passwords remain the most widely used authentication method, but they come with significant limitations, particularly their vulnerability to theft and cyberattacks. As a result, there is an increasing need for multi-factor authentication (MFA) systems that add various layers of security, or more complex, biometric-based features. However, implementing biometric systems can be costly, both in terms of performance and privacy [2], and their precision can vary according to environmental conditions or individual differences [3]. Moreover, MFA suffers from various weaknesses, such as the effort required to retrieve, remember, and enter information that collides with the need for timely and effective authentication for healthcare services and/or the lack of user education that makes MFA difficult to use properly by healthcare actors without proper CS skills and capabilities. Lastly, current authentication solutions have a centralization to a given server, dealing with the work of processing identity attributes and user authentication. Such a centralization may imply illegitimate profiling of user habits over the web and digital service of interest. Authorization, on the other hand, governs resource access based on the user's role and requirements. Common access control models include RBAC (Role-Based Access Control), ABAC (Attribute-Based Access Control), ReBAC (Relationship-Based Access Control), MAC (Mandatory Access Control), and DAC (Discretionary Access Control). Privacy has become a key issue, and some of these approaches are critical to maintaining when a privacy solution needs to be used since attributes may involve personal data. Moreover, privacy-related legal obligation needs solutions to ensure that personal data is not collected without explicit user consent or a valid legal basis to authorize specific data processing. This novelty causes a shift towards the need for user-centric authorization, which means having each subject manage and protect their personal information autonomously and securely by personally indicating the authorization policies.

The mentioned user-centricity and higher privacy presentation call for innovative solutions in the context of authentication and authorization services. On the one hand, Self-Sovereign Identity (SSI) offers a safer, more privacy-preserving approach to identity management [4], as it enables individuals to have complete control over their data, unlike centralized models where a single authority manages information. On the other hand, Zero-Knowledge Proofs (ZKP) emerge as a solution for privacy preservation [5]. ZKPs allow a party to prove the truth of a statement without revealing additional information. Advanced variants, like zk-SNARKs,

979-8-3315-9984-3/25/$31.00 ©2025 IEEE

enhance efficiency and security by validating transactions without real-time interaction, making them especially useful in blockchain environments where privacy and scalability are critical. In the current literature, some proposed solutions based on SSI or ZKP exist to have more effective and efficient authentication and authorization within the medical sector. This work presents a novel approach to improving authentication and authorization in the healthcare sector using Soulbound Tokens (SBTs) as a method to provide both authentication and authorization for implementing ABAC in the medical domain. These non-transferable tokens are uniquely linked to an individual, providing a secure way to verify patient identity and grant access based on an attribute, such as medical treatments. This work makes three key contributions:

- It introduces a novel, revocable, and privacy-aware ABAC-compliant authentication schema that leverages SSI and SBT. This schema enhances privacy without revealing the hold attributes while retaining the benefits of the ABAC model.
- It presents a concrete use case for applying the proposed framework within the medical domain, where patients are identified based on their disease. This disease, treated as an attribute, is a basis for granting access to the appropriate treatment.
- It evaluates the costs, performance, and security of the proposed protocol, considering the current Ethereum network. The evaluation also includes real-time application demonstrations, highlighting the protocol's practical viability while evaluating the protection against a Dolev-Yao schema.

This approach, which combines SSI, blockchain, and ZKP technologies, ensures secure, privacy-preserving, and efficient handling of authentication and authorization in the healthcare domain. The manuscript is structured into six sections: in Section II, we introduce the concepts of SSI, SBT, and ZKP. Section III discusses existing work in the context of decentralized authentication for the medical domain. Section IV discusses the proposed authentication protocol. Performance results are presented in Section V, along with cost analysis. Section VI concludes the manuscript, presenting limitations and future research directions.

II. Background

A. Self-Sovereign Identity (SSI)

Identity takes on new complexities in the digital sphere, raising concerns about verifying and authenticating the individuals we interact with online. Static identifiers like email and passwords, while standard in digital interactions, do not guarantee the authenticity of identity. A centralized identity system is where a single entity, such as a service provider, collects and stores users' identity-related information, including credentials and personal data. Centralized authentication systems store sensitive data, posing a risk of large-scale breaches due to system vulnerabilities. Furthermore, such systems raise privacy concerns, as organizations may collect and use user data for commercial purposes without explicit consent or sufficient transparency. The fragmentation of digital identities, requiring multiple accounts for different services, complicates credential management and undermines security and privacy. A decentralized identity model, enabled by technologies like blockchain, is known as Self-Sovereign Identity (SSI). Unlike traditional account-based systems, SSI allows users to interact directly with services and applications while retaining data control. This approach ensures data sharing occurs only when necessary and without intermediaries. In an SSI model, the user's identity is unique and managed through a digital wallet, eliminating the need for multiple credentials to access services like banks or public institutions. SSI addresses issues inherent in centralized models, such as poor privacy and data breach risks, by emphasizing decentralization and user control.

Decentralized Identifiers (DIDs) enable the SSI model. These are unique identifiers generated by users, independent of centralized registries. DIDs can represent various entities—people, organizations, objects—and ensure security and privacy through cryptography. DIDs rely on verifiable data registries, like blockchain or peer-to-peer networks, to ensure the authenticity, integrity, and accessibility of identifiers. Tools like DID resolvers and dereferences allow applications to retrieve or directly interact with resources linked to a DID. Verifiable Credentials (VCs) replicate traditional credentials, such as ID cards, but with enhanced security through digital signatures. A VC typically contains subject information, details about the issuing authority, and proofs of authenticity. Holders of VCs can generate Verifiable Presentations (VPs) to securely share credentials with verifiers. The main actors in the VC ecosystem are: (1) Holder / Prover which possesses the credentials, stored securely in a digital wallet; (2) Issuer which provides the credentials, such as governments or organizations (3) Verifier which validates the credentials for authenticity and integrity. VCs operate on trust models, requiring confidence in issuers, registries, and repositories to ensure data integrity and security.

B. Soulbound Token (SBT)

Soulbound Tokens (SBTs) are a groundbreaking innovation in the blockchain world. They are designed to represent personal credentials and digital identities in a permanent and non-transferable way. Unlike NFTs, which focus on owning and trading digital assets, SBTs are tied to individuals and used to verify credentials such as educational achievements, work experience, or community memberships. By recording this information on the blockchain, SBTs eliminate reliance on centralized systems, ensuring transparency, immutability, and user control.

These tokens cannot be transferred or sold, ensuring their integrity and preventing misuse. They also incorporate features like ”social recovery,” allowing trusted parties to help regain access if needed. SBTs can demonstrate proof of qualifications and reputation in a decentralized, trust-based environment, making them ideal for job applications, DAO memberships, or certifying skills. Ultimately, SBTs represent a shift toward empowering individuals with verifiable, decentralized identities, paving the way for a more equitable and transparent Web3. They can potentially revolutionize how we think about reputation, trust, and identity in the digital age.

C. Zero-Knowledge Proofs (ZKP)

In cryptography, a zero-knowledge proof (ZKP) is a protocol that allows a prover (P) to convince a verifier (V) that a given statement is true, without revealing any information beyond the validity of the statement itself. The fundamental principle of these proofs is that the verifier gains certainty that the statement is true but does not learn any additional details about how the proof was obtained or the content of the statement itself.

A zero-knowledge proof must satisfy three essential properties:

1) Completeness: If the statement is true, an honest prover can convince an honest verifier of its validity.
2) Soundness: If the statement is false, a dishonest prover cannot convince an honest verifier.
3) Zero-knowledge: If the statement is true, a dishonest verifier learns nothing beyond the truth of the statement. This is ensured by the existence of a simulator that can produce a proof indistinguishable from the one generated by the prover in the real interaction, without knowing the content of the statement.

In other words, the verifier acquires only one piece of information: the truth of the statement, without obtaining any additional details. Zero-knowledge proofs can be categorized into two main types: interactive proofs and non-interactive proofs. In interactive ZKPs, the prover and verifier engage in multiple rounds of interaction during the proof process. This interaction typically involves a series of challenges and responses. In each round, the verifier challenges the prover with a question or assertion, and the prover responds with evidence supporting the truth of their statement. This process continues until the verifier is convinced of the validity of the statement.

In non-interactive zero-knowledge proofs (NIZK), the prover provides a proof that the verifier can verify at any time without further interaction between the two. NIZKs do not require repeated message exchanges between the prover and verifier. Instead, the prover generates the proof once and sends it to the verifier, who can independently verify it.

These protocols are particularly well-suited for applications such as blockchain, where the verifier can be a smart contract. In such scenarios, the smart contract verifies the proof provided by the prover and takes actions, such as approving or rejecting a transaction on Ethereum, based on its validity.

Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge (zk-SNARKs) represent one of the most advanced forms of cryptographic zero-knowledge proofs. These systems enable a prover to demonstrate the validity of a statement without revealing any additional information or requiring further interaction with the verifier. The key features of zk-SNARKs are:

- Succinctness: zk-SNARK proofs are highly compact, even for complex statements, and can be verified in linear time relative to the size of the public input.
- non-interactivity: Unlike traditional interactive proofs, zk-SNARKs require only a single interaction between the prover and verifier, eliminating the need for multiple communications.
- Zero-knowledge: The verifier learns no information about the witness (*e.g.*, w) beyond the validity of the statement.
- Soundness: Only a prover who genuinely possesses the secret witness can generate a valid proof, preventing attempts at falsification.

zk-SNARKs rely on a polynomial-time computable relationship $P(x, w)$ to verify if $P(x, w) = 1$, where:

- x: The public value of the statement.
- w: The secret witness that validates x.

The construction of a zk-SNARK proof involves the following steps:

a) Definition of the Program:: A program P takes as input a public value x and a secret witness w, returning 1 if the witness is correct and 0 otherwise. The prover's goal is to demonstrate that there exists a w such that $P(x, w) = 1$.

b) Encoding the Program:: The program P is encoded as a computable relation that can be expressed in polynomial form. For instance, the polynomial $x^3 + x + 5$ is satisfied by $x = 3$, which serves as the witness w.

c) High-Level Code Representation:: The polynomial function can be implemented in a high-level programming language, such as Python. Below is an example:

```
def f(x):
    y = x**3
    return x + y + 5
```

d) Creation of an Algebraic Circuit:: The program P is represented as an algebraic circuit consisting of elementary operations (*e.g.*, additions and multiplications) applied to variables and constants. Each gate in the circuit corresponds to an algebraic operation linking the variables.

e) Conversion to a Rank-1 Constraint System (R1CS):: The algebraic circuit is transformed into a Rank-1 Constraint System (R1CS). These constraints are expressed as triplets of vectors (v, w, k) that satisfy the equation:

$$t \cdot v \cdot t \cdot w - t \cdot k = 0,$$

where t is a solution to the system, and $\cdot$ represents the dot product between vectors. This equation ensures that specific relationships between the vectors hold.

f) Conversion to a Quadratic Arithmetic Program (QAP):: The R1CS system is converted into a Quadratic Arithmetic Program (QAP), where constraints are represented by polynomials. These polynomials, derived using Lagrange interpolation, facilitate simultaneous verification of the constraints.

g) zk-SNARK Proof Generation:: Using the generated polynomials, such as H and Z, the prover creates an encrypted proof in the form of a pair of polynomials $[g_H, g_Z]$, demonstrating knowledge of the correct witness without revealing it. The prover employs the proving key in this step.

h) Proof Verification:: The verifier receives the proof $[g_H, g_Z]$ along with a verification polynomial g_T, representing the target polynomial T. Cryptographic operations are performed to compare $g_H \cdot g_Z$ with g_T. If the values match, the verifier is convinced that the prover has the correct witness. The verification key is used during this step.

III. Related works

The literature has recently highlighted a growing interest in self-sovereign and privacy-preserving digital identity solutions, focusing on using Soulbound Tokens (SBTs) and Zero-Knowledge Proofs (ZKPs). Traditional authentication mechanisms (*e.g.*, OAuth, OpenID Connect, or biometric authentication) are affected by the issue that verification of credentials and authorization decisions are centralized at a single entity, capable of potentially profiling user habits and representing a privacy issue [6]. Blockchain-based and SSI-modeled authentication schemes are unaffected by such an issue due to their decentralized nature. They have attracted increasing interest, as the new EIDAS regulation highlights decentralized authentication as a primary means of implementing digital identities in Europe [7].

A primary category of studies explores the implementation of SBTs in specific contexts, such as COVID-19 vaccination certification. In this domain, several studies have demonstrated that these tokens ensure the traceability and authenticity of health certifications while preventing the sale or transfer of such information, thus fostering trust in health certifications [8]. Another research area focuses on the adoption of SBTs in the education sector, where digital certificates issued as NFTs provide a decentralized and secure solution for validating students' acquired competencies. This innovation not only enhances the reliability of certifications but also reduces the costs and time associated with their issuance, paving the way for new practices in education and training [9].

An additional area of research involves user authentication in virtual environments and the metaverse. Here, integrating decentralized identities (DIDs) with SBTs has shown potential for facilitating Know Your Customer (KYC) processes while ensuring user privacy through the use of ZKPs [10]. A recent proposal by Kalbantner et al. [11] introduces a KYC system based on ZKPs and SBTs to ensure regulatory compliance without revealing sensitive data, adhering to self-sovereign identity principles and encouraging positive behavior through reputation management in decentralized marketplaces.

The protection of privacy in complex contexts, such as healthcare systems and IoT networks, is another area where SBTs and ZKPs have made significant strides. Recent studies have proposed blockchain-based protocols that combine these technologies to grant access to sensitive data only to authorized users while maintaining privacy, even in potentially vulnerable situations [12], [13]. These studies not only demonstrate the effectiveness of SBTs and ZKPs in digital identity management but also open new avenues for addressing data security challenges in an increasingly interconnected world.

Parallel to these advancements, several studies focus on using SBTs for managing privacy and security in decentralized environments. The DSMAC (Decentralized Self-Management of Data Access Control) system [14] proposes a blockchain-based model and self-sovereign identities for secure healthcare data management. This approach leverages smart contracts and verifiable credentials to address privacy and scalability challenges, providing a robust solution for emergency scenarios.

In the domain of wireless sensor networks, Anshul et al. have proposed a ZKP-based mechanism [15] to ensure the secure authentication of nodes. This solution mitigates the risks of attacks that could compromise network integrity by employing a modified Guillou-Quisquater identification system combined with the μTESLA protocol for secure authentication. Another application focuses on content protection in Content-Centric Networks (CCNs). Shashidhara et al. developed the NextGen Authentication protocol [16], which utilizes SBTs to ensure the integrity and authenticity of content, eliminating vulnerabilities of traditional systems and preventing sophisticated attacks, such as denial-of-service attacks. In this context, SBTs also help manage content access, reducing risks associated with unauthorized access.

An important contribution comes from Stokkink et al. [17], who explored integrating self-sovereign digital identities (DIDs) with network-level anonymization techniques. Their TCID (Truly Self-Sovereign Identity) system aims to balance privacy with the credibility of information in scenarios where digital identities must remain entirely private yet verifiable.

This work introduces key innovations to address the limitations of current healthcare data protection solutions. By integrating zk-SNARKs, it enables the verification of sensitive information, such as patient diagnoses, without revealing the underlying data, thus enhancing privacy and security beyond traditional methods. Additionally, the use of SBTs within a SSI framework, combined with Attribute-Based Access Control for managing diagnoses, allows for more efficient credential management. The novel approach of issuing SBTs for categories of diagnoses simplifies management, reduces costs, and optimizes wallet space, making the system more scalable and adaptable to growing healthcare demands.

IV. System Model

The proposed system aims to extend the typical approaches based on SSI and it is particularly aimed to medical context where patients need to request treatments either for a specific diagnosis or for an entire category of diagnoses. In the proposed architecture we leveraged ZKP for the verification of the validity of requests without exposing sensitive information, such as the patient's diagnosis. Only after successful verification through ZKP is the treatment associated with the diagnosis, ensuring the patient's privacy is not compromised. In particular, considering the powerful of ZKP, we will explore two different approaches, both relevant to the medical use-case:

- **Treatment for a Single Disease**: Patients can request a specific treatment t for an identified condition, such as a particular kind of condition (eg. breast cancer) without disclosing additional diagnostic details. The treatment t is exclusively linked to the identified disease d, reducing the risk of errors and ensuring a targeted and secure approach.
- **Treatment for a Disease Category**: Patients can request a treatment t_k applicable to all diseases $d_1, d_2, ..d_i \in C_k$ within a specific subcategory k, such as *lung cancer* or *melanoma*. In this scenario, the treatment applies to diseases within the same category, while specific details about individual diagnoses remain undisclosed. This approach optimizes the management of treatments while preserving patient confidentiality.

This framework leverages the privacy-preserving capabilities of SSI and ZKP to achieve a balance between precise treatment application and the protection of sensitive patient information. The entire framework is divided into four phases: initialization, enrollment, authentication and authorization, and revocation, where phases two and three are depicted in Fig. 1. The framework is based on two smart contracts responsible for the release of the SBTs and for their verification, represented in Fig. 1 at step 2 and 5.

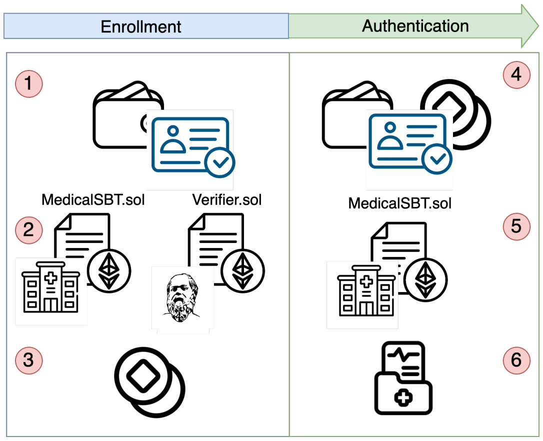

Fig. 1. System Model characterized by enrollment phase (left), and authentication phase (right).

In particular, these two contracts are responsible for:

- *MedicalSBT.sol*: The MedicalSBT.sol is divided into MedicalRecordSBT.sol and MedicalRecordSBTwCategory.sol, which are contracts designed for managing medical records using Soulbound Tokens (SBT) on the blockchain, leveraging the OpenZeppelin ERC721 framework. Both contracts allow a user to possess multiple SBTs, associated with different diagnoses or categories of diseases, through mappings that link each Ethereum address to multiple tokens. The primary difference between the two contracts lies in how they handle treatment permissions: one manages permissions for individual diagnoses, while the other handles them for categories of diseases, according to what we introduced at the beginning of this section.
- *Verifier.sol*: Facilitates the verification of cryptographic proofs using zk-SNARKs, and autogenerated using Zokrates. The contract is composed of two main parts. First, the Pairing library provides the building blocks for elliptic curve operations, which are essential for zk-SNARK proof validation. Second, the Verifier contract integrates these cryptographic operations into a coherent mechanism for proof validation. It defines a structure called VerifyingKey that stores the cryptographic keys needed for the verification process. The contract's primary function, verify, checks the validity of a zk-SNARK proof by ensuring it satisfies specific mathematical conditions.

In addition, the framework Veramo is responsible for the management of the communication between the cryptographic wallet and the smart contracts. This interaction is assisted by a graphical interface which helps the user to release the SBT.

A. Initialization

As depicted in Fig. 2 the ZoKrates compiler compiles the high-level circuit to create the algebraic circuit and the conversion to R1CS. Once the linear code is ready, it is processed to derive the Proving Key K_P and the Verification Key K_V. K_P is stored by the Prover which is any user of the system interested in participate to the authentication protocol. K_V instead, is exported and inserted in the *Verifier.sol* smart contract for the verification of ZKP produced by the provers in the enrollment phase. To simplify the analysis of the proposed architecture, we consider a schema typically used in the context of SSI within medical domain [18], where the User is equipped with a $vc = (d_i, sign(d_i, PK_I)$, composed of the disease d_i signed with the Public Key PK_I belonging to the Issuer.

B. Enrollment

We consider a system where the user is equipped with a set of VCs securely stored in a cryptographic wallet. These credentials are issued by the National Sanity Service (NSS) leveraging the Ethereum Decentralized Identifier (DID) Method (*did:eth*), allowing verification through the *Ethereum DID Registry* maintained on the Ethereum blockchain. Unlike traditional VCs, which may directly expose attributes, the NSS enhances security and privacy by providing the user with a Proving Key K_P. This key facilitates the creation of zero-knowledge proofs (ZKPs), encoded as polynomial representations $[g_H, g_Z]$, enabling users to prove possession of valid attributes without disclosing sensitive details.

Once the user saves the VCs in their cryptographic wallet, they can initiate the enrollment process. As depicted in the Fig. 3, the primary attribute used for authentication is the medical diagnosis d_i, while other fields within the VC are selectively disclosed. The diagnosis attribute undergoes a privacy-preserving transformation via a ZKP generation process. This involves two distinct steps: first, a witness w

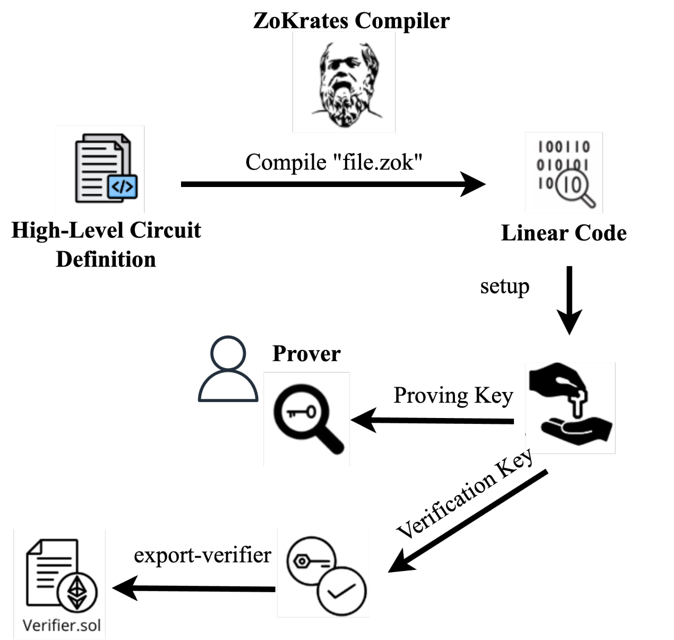

Fig. 2. ZoKrates Initialization steps, composed by the circuit compilation (top), key setup (middle), and key export (bottom).

is derived by hashing the diagnosis(s) d_i that the user wants to prove together with the circuit c; second, the witness w is combined with K_P to produce the cryptographic proof. This proof is encapsulated in a packet m_1 together with the ETH address of the Prover A_U and the Prover Key K_P composed of $[g_H, g_Z]$. The message m_1 is sent to the NHS for verification. The NHS invoke the method *requestSBT* contained in the *MedicalSBT.sol*.

To mint the SBT, the NSS interacts with the *MedicalSBT.sol* smart contract deployed on the Ethereum blockchain. The smart contract validates the ZKP by invoking a verifier circuit to ensure the proof's integrity. If the verification is successful, the NSS mints the SBT and delivers it to the user's cryptographic wallet. This SBT serves as a robust and privacy-preserving credential for future authentication processes.

With the SBT now in possession, the user no longer needs to share any identifiable information. The SBT inherently encapsulates the ZKP, which securely represents the diagnosis in a privacy-aware manner. This enables the user to access treatments or services without the need for repeated proof generation or data exchange. Furthermore, this approach significantly reduces the verification time, as the authentication process leverages the pre-validated SBT, streamlining interactions while maintaining robust privacy and security guarantees.

C. Authentication and Authorization

The authentication process adopts a challenge-response schema, widely used in decentralized environments. As depicted in Fig. 3, it begins with the Holder sending its address, denoted as A_U, which is bound to a public key $PK(A_U)$, through a self-containerization of cryptographic address [19]. The NHS generates a nonce n_2 and encrypts it using the public key $PK(A_U)$ of the Holder, then transmits the encrypted nonce $\{n_2\}_{PK(A_U)}$ to the Holder. Only the genuine owner of the wallet associated with A_U can decrypt this message and retrieve n_2. To complete the challenge, the Holder must send the decrypted nonce back to the NHS. If the decrypted nonce matches the original n_2 generated by the NHS, the Holder is considered eligible for authentication, with the address A_U, which does not imply any information about the identity of the user, but just assess that the communication is authenticated for that specific address.

Once authentication is established, an authorization check is performed using the *MedicalSBT.sol* contract. The *canUserReceiveTreatment* function is invoked with the previously authenticated address A_U and the requested treatment d_i. The function verifies whether:

1) A_U is included in the list of authorized SBT holders ($A_U \in A_{sbt}$).
2) The hash of the diagnosis shared by the Holder ($\text{hash}(d_i)$) matches one of the hashes stored in the SBT associated with A_U ($\text{hash}(d_i) \in sbt_{A_U}$).

If both conditions are satisfied, the Holder is authenticated and deemed can proceed to access the resources.

D. Revocation

The revocation protocol follows an offline interaction in which the Prover must physically verify their identity before a NHS authority. We assume that the NHS maintains a registry linking the physical identities of Provers to their respective cryptographic identifiers, denoted as A_U. This identifier A_U is also referenced by the `MedicalSBT.sol` contract to validate the Prover's authentication credentials, specifically their possession of P_w. Furthermore, we argue that this registry is embedded within the credentials issued to the Prover by an authorized entity. These credentials incorporate multiple identity attributes, including physical identity card numbers, ensuring consistency between the Prover's physical and digital identities. To revoke a Holder's authentication privileges, the NHS removes the identifier A_U from the authorized address list A_{sbt} within the system. As a result, when the Holder attempts to authenticate following the procedure outlined in Fig. 3, the NHS initially accepts A_U as a valid identifier. However, upon invoking the `MedicalSBT.sol` contract—specifically the function `canUserReceiveTreatment`(A_U, d_i)—the system denies authentication, as $A_U \notin A_{sbt}$ due to the revocation process applied. This revocation mechanism ensures that the Holder is permanently excluded from the authentication system, thereby preventing unauthorized access to medical services.

V. Results

This section describes the experiments conducted to evaluate the performance and results of blockchain operations needed for the entire lifecycle of the proposed authentication approach. The tests were designed and performed to measure three key metrics: execution times, gas consumption, and the associated economic costs of the operations. The cost in euros

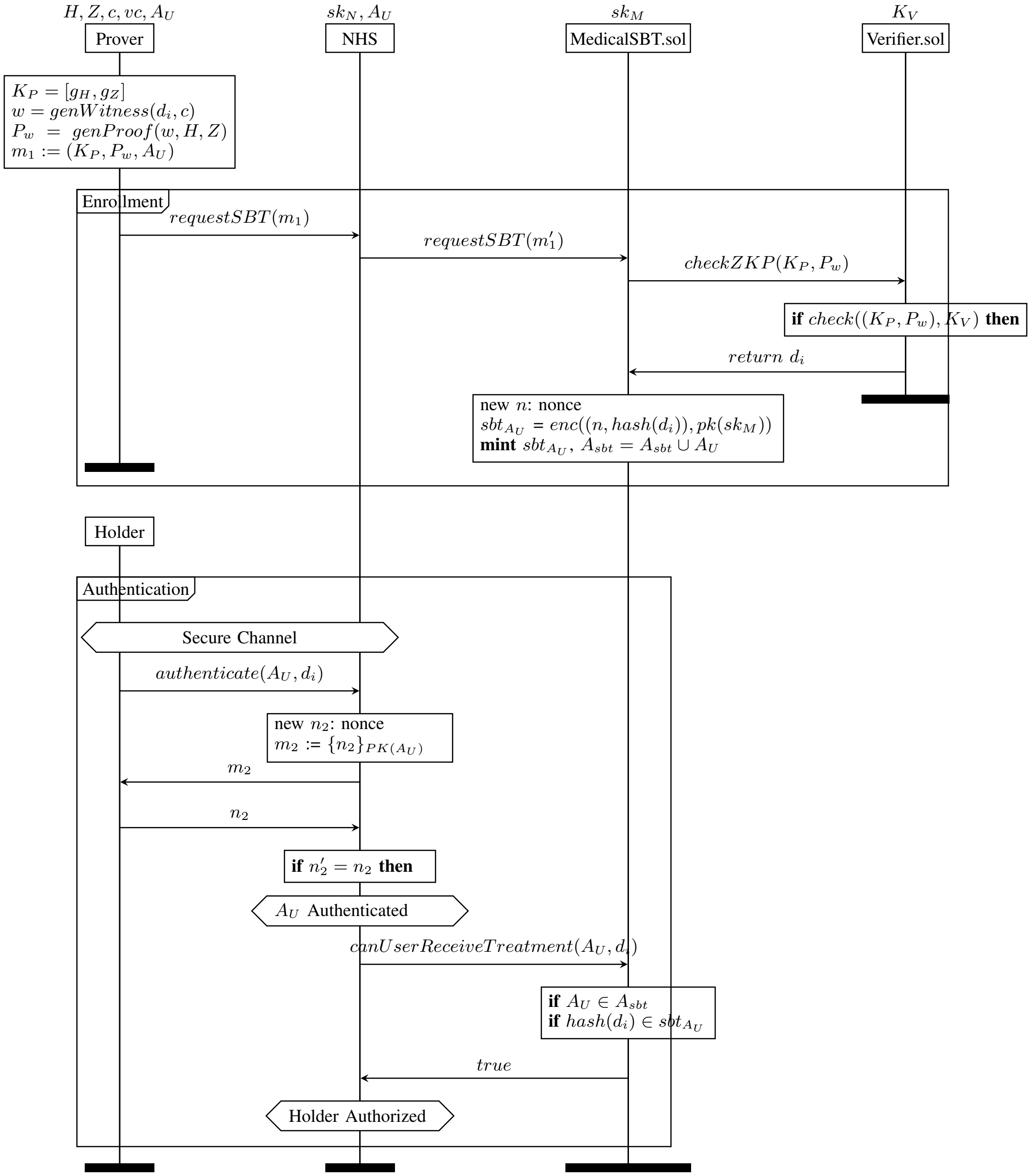

Fig. 3. Messaging protocol between Prover, NHS, MedicalSBT.sol, and Verifier.sol during enrollment and authentication phases.

was calculated based on the gas price in *Gwei*, assuming a fixed value of 13.98, and the Ethereum-to-EUR exchange rate of 1 ETH = 2,937.44 EUR, set according to prevailing market prices at the time of testing. To simulate a blockchain we run a Hyper-ledger Besu Docker image equipped with QBFT consensus over an iMac 3.3 GHz Intel Core i5 6 cores equipped with 16 GB 2667 MHz DDR4.

A. Cost Analysis

The overall cost of the authentication procedure is represented in Table I. For the initialization, the costs are relatively high since it requires the deployment of two smart contracts (*Verifier.sol* and *MedicalSBT.sol*), with 42.82 EUR spent by the user, with a notable reduction of cost with respect to previous implementation leveraging SBT and SSI [20]. Clearly, this cost can be hypothetically financed by the public health system. At this point, the system is ready to work, and the only cost that must be paid is the cost for the enrollment, namely the issuance of Single Treatment SBT and Category Treatment SBT. The price for both is more or less the same, with the Single Treatment SBT requiring less computation since it gives access to just one treatment, while the Category SBT provides access to a set of treatments, justifying its enhanced cost. The function *canUserReceiveTreatment*, required for the authentication, is of type *view*. It was not necessary to incur any gas costs during its execution. This is because the function does not modify the state of the blockchain but simply queries the data without performing any write operations. This is an important aspect of the system since this will be the operation executed most of the time. The revocation requires less gas compared to the issuance, and this is justified by the fact that it only involves the deletion of a record in the list of authorized addresses.

TABLE I
GROUPED OPERATIONS WITH EXECUTION TIME, GAS USED, AND COST.

Phase	Operation	Exec. Time (ms)	Gas Used	Cost in EUR
Initialization	Issuer Deployment	4203.92	604636	24.83
	User Deployment	8506.59	1042618	42.82
Credential Issuance	Single Cred. Iss.	102.50	0	0
	Category Cred. Iss.	91.88	0	0
Enrollment	Single BST	5368.14	851265	34.96
	Category BST	9602.44	920011	37.78
Authentication	Single BST	14.72	0	0
	Category BST	15.03	0	0
SBT Revocation	Single BST	8212.95	127103	5.22
	Category BST	4184.49	107938	4.43

While the gas cost calculations provided in this work offer a detailed assessment of the resource requirements for executing smart contract functions, it is worth exploring how Layer-2 (L2) solutions could further mitigate these costs. Anyway, it is interesting to notice that Layer 2 networks encounter is still subject to data availability problems: storing transactions completely offchain poses a risk of data loss [21]. For this aspect, we leave to future implementation a careful evaluation of the pros and cons of this potential implementation, which can surely pave the way for a cost reduction, and increased scalability.

B. Performance Analysis

Figure 4 illustrates the average time required for the enrollment procedure, which is divided into three key phases: ZKP creation, ZKP verification, and SBT minting. The total time for the execution of the enrollment is 5.36s for the single-category SBT, and 9.60s for the multi-category SBT, as shown in Table I. The time taken to generate the proof, P_w, is 975 ms for the single treatment and 1.20 s for the category treatment. This highlights the additional time needed when using a larger circuit for proof generation. For ZKP verification, the time required for P_w is approximately 149 ms for the single treatment and 114 ms for the category treatment. The most time-consuming operation is the SBT issuance, which takes 4.24s for the single treatment and twice as long for the category treatment. The time required for SBT issuance is significantly influenced by network occupancy and may vary accordingly. These times are strongly in line with typical approaches leveraging blockchain [22], [23].

Figures 5 and 6 present the system's performance in terms of authentication time. Both approaches show low authentication times, near 15ms although some fluctuations are observed. The single-category SBT stands out for its consistency, with minimal variation in execution times between iterations. In contrast, the multi-category approach, while demonstrating overall stability, exhibits slight fluctuations between the different subcategories. Overall, both approaches yield authentication times that align with typical expectations.

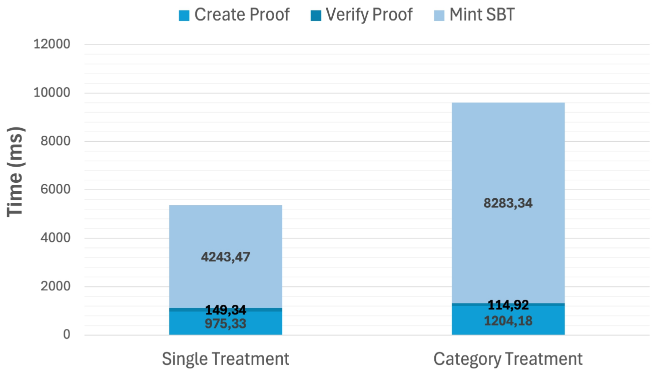

Fig. 4. Average Time for Enrollment divided into proof creation, proof verification, and SBT Minting

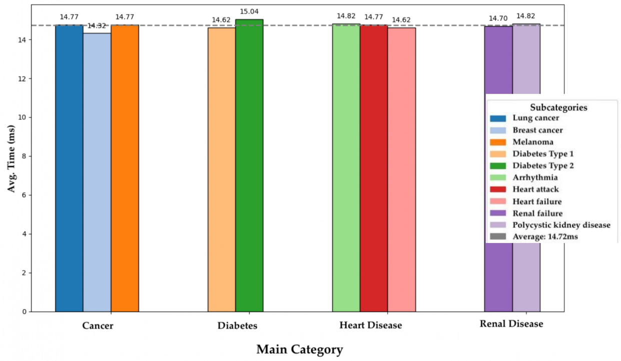

Fig. 5. Average Time for Authentication by subcategories - Arrhythmia

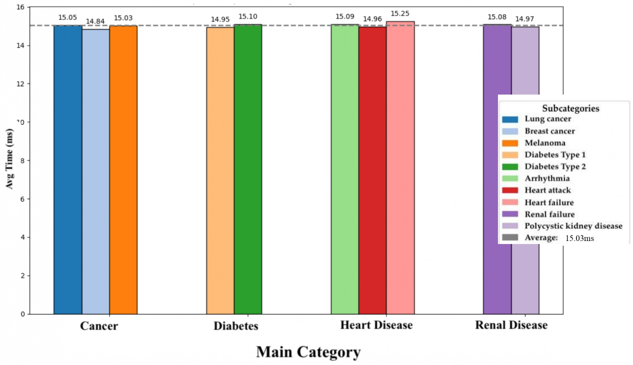

Fig. 6. Average Time for Authentication by subcategories - Renal Disease

C. Formal Verification

In this section, we evaluate the security of the protocol using formal threat modeling, employing the ProVerif tool [24] under the Dolev-Yao adversarial model. The Dolev-Yao model, widely recognized as a powerful abstraction [25] for analyzing cryptographic protocols, assumes that cryptographic primitives are perfect and that the attacker has complete control over the communication channel. The protocol is represented as a series of interactions involving alternating queries and responses. Our analysis is structured in two parts: (i) authentication of the holder and (ii) authorization of access.

1) Authentication: Authentication is achieved when the NHS successfully receives the holder's response to the nonce n_2, which was sent in the previous message. This nonce is encrypted using the address A_U, which is linked to the holder's public key. It is important to emphasize that this address A_U is only just shared prior to this step, and therefore no concrete guarantee regarding the identity of the user behind A_U is established at this stage. This guarantee is instead ensured during the subsequent authorization phase.

The following *correspondence query* captures the authentication logic:

$$HolderAuthenticatedByNHS(m_2, n_2) \Rightarrow \\ HolderSentLastMessageToNHS(m_2, n_2)$$

This property holds, indicating that if the NHS authenticates a holder (i.e., successfully receives the correct n_2), then the holder must have previously received m_2 and responded with the decrypted value. In other words, the NHS cannot complete the authentication event unless the holder has already participated in the protocol by decrypting and replying with n_2.

2) Authorization: The authorization phase ensures that any decision by the NHS to grant access is conditioned upon both a valid preceding interaction with the holder and a successful authorization verification by the smart contract `MedicalSBT.sol`. This is formalized through the following correspondence query:

$$\mathit{HolderAuthorizedByNHS}(m_2, n_2, m_3, m_4) \Rightarrow \\ \mathit{HolderSentLastMessageToNHS}(m_2, n_2) \wedge \\ \mathit{HolderAuthorizedByMRecord}(m_3, m_4)$$

This ensures that the authorization issued by the NHS is contingent upon two verifiable events: (1) A prior valid authentication interaction with the holder. (2) A matching authorization decision is recorded on-chain by the smart contract.

It is worth noting that this property critically depends on the smart contract's ability to verify that the tuple (A_U, d_i) included in m_3 exists in the array of released SBTs. Although these values could, in theory, be manipulated by the NHS, there is no incentive or security gain for the NHS to do so. The primary adversary considered in this model is the holder, who may act maliciously.

However, the holder cannot forge a valid m_3, as it must be linked to ownership of A_U, which was already validated during the authentication phase. Consequently, the authorization cannot be falsely triggered, because the nonce n_2, validated earlier, cannot be derived from the encrypted m_2 unless the holder has followed the legitimate protocol steps.

VI. Conclusion

The proposed architecture demonstrates that the integration of zk-SNARKs, which verify the validity of information contained in Verifiable Credentials (VCs), ensures that only the data strictly required for medical treatment is disclosed. Preliminary evaluations show that zk-SNARK proof generation and verification times average under 15.08 ms, while the token issuance time aligns with previous studies on the use of Soulbound Tokens (SBTs) in the medical domain [20]. These SBTs facilitate disease verification and enable authentication and access to medical services by employing Attribute-Based Access Control (ABAC), thereby keeping the user's specific disease confidential while still ensuring access to the appropriate reserved services. The effectiveness of SBTs was demonstrated through two distinct implementations: one targeting specific diseases, and another categorizing treatments to grant access to all services within a defined category. We also proposed a revocation mechanism for managing SBTs, enhancing their practical applicability in dynamic healthcare contexts. This solution addresses the privacy limitations of traditional authentication schemes, as its decentralized nature prevents user profiling and complies with GDPR legal obligations [26], [27].

In future work, we plan to perform an in-vitro experiment to gain insights into the scalability of the zk-SNARK circuits. This experiment will show our solution's efficiency for 10 diseases grouped into four categories. Lastly, we also planned to make a benchmarking campaign comparing zk-SNARKs with other privacy-preserving solutions (*e.g.*, ring signatures, homomorphic encryption) to strengthen the evaluation of our solution.

Acknowledgements

This work was partially supported by the project "DHEAL – COM- Digital Health Solutions in Community Medicine" under the Innovative Health Ecosystem (PNC) - National Recovery and Resilience Plan (NRRP) program funded by the Italian Ministry of Health.

References

[1] J. Lopez, R. Oppliger, and G. Pernul, "Authentication and authorization infrastructures (aais): a comparative survey," *Computers & Security*, vol. 23, no. 7, pp. 578–590, 2004.

[2] A. Kusyanti, H. P. A. Catherina, P. Effendrik, N. Santoso, and N. S. Ekowati, ""risky or trustworthy?": User behavior towards biometric authentication method," *Procedia Computer Science*, vol. 234, pp. 428–435, 2024.

[3] F. A. Yassine and G. Abdelkader, "Eeg-based biometric authentication using machine and deep learning approachs: A review," in *2024 8th International Conference on Image and Signal Processing and their Applications (ISPA)*. IEEE, 2024, pp. 1–8.

[4] R. Petrlic, "Ssi is here to support the rights of data subjects," in *2024 IEEE International Conference on e-Business Engineering (ICEBE)*, 2024, pp. 133–138.

[5] J. Kurmi and A. Sodhi, "A survey of zero-knowledge proof for authentication," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 1, 2015.

[6] M. Al Shabi and R. R. Marie, "Analyzing privacy implications and security vulnerabilities in single sign-on systems: A case study on openid connect." *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 4, 2024.

[7] S. Schwalm, D. Albrecht, and I. Alamillo, "eidas 2.0: Challenges, perspectives and proposals to avoid contradictions between eidas 2.0 and ssi," in *Open Identity Summit 2022*. Gesellschaft für Informatik eV, 2022, pp. 63–74.

[8] M. I. Lunesu, R. Tonelli, A. Pinna, and S. Sansoni, "Soulbound token for covid-19 vaccination certification," in *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 2023, pp. 243–248.

[9] S. Nikolić, S. Matić, D. Čapko, S. Vukmirović, and N. Nedić, "Development of a blockchain-based application for digital certificates in education," in *2022 30th Telecommunications Forum (TELFOR)*. IEEE, 2022, pp. 1–4.

[10] G. Kim and J. Ryou, "Digital authentication system in avatar using did and sbt," *Mathematics*, vol. 11, no. 20, p. 4387, 2023.

[11] J. Kalbantner, K. Markantonakis, D. Hurley-Smith, and C. Shepherd, "Zkp enabled identity and reputation verification in p2p marketplaces," in *2024 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2024, pp. 591–598.

[12] D. A. Luong and J. H. Park, "Privacy-preserving blockchain-based healthcare system for iot devices using zk-snark," *IEEE Access*, vol. 10, pp. 55 739–55 752, 2022.

[13] M. À. Cabot-Nadal, B. Playford, M. M. Payeras-Capellà, S. Gerske, M. Mut-Puigserver, and R. Pericàs-Gornals, "Private identity-related attribute verification protocol using soulbound tokens and zero-knowledge proofs," in *2023 7th Cyber Security in Networking Conference (CSNet)*. IEEE, 2023, pp. 153–156.

[14] H. Saidi, N. Labraoui, A. A. A. Ari, L. A. Maglaras, and J. H. M. Emati, "Dsmac: Privacy-aware decentralized self-management of data access control based on blockchain for health data," *IEEE Access*, vol. 10, pp. 101 011–101 028, 2022.

[15] D. Anshul and S. Roy, "A zkp-based identification scheme for base nodes in wireless sensor networks," in *Proceedings of the 2005 ACM symposium on Applied computing*, 2005, pp. 319–323.

[16] R. Shashidhara, R. Chirakarotu Nair, and P. Kumar Panakalapati, "Nextgen authentication: A secure blockchain-based protocol for content-centric networks with soulbound tokens," *IEEE Access*, vol. 12, pp. 111 293–111 310, 2024.

[17] Q. Stokkink, G. Ishmaev, D. Epema, and J. Pouwelse, "A truly self-sovereign identity system," in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*. IEEE, 2021, pp. 1–8.

[18] M. Barbareschi, B. Boi, F. Cirillo, M. De Santis, and C. Esposito, "Securing the internet of medical things using puf-based ssi authentication," in *CEUR Workshop Proceedings, 8th Italian Conference on Cybersecurity, ITASEC*, vol. 3731, 2024.

[19] S. Suratkar, M. Shirole, and S. Bhirud, "Cryptocurrency wallet: A review," in *2020 4th international conference on computer, communication and signal processing (ICCCSP)*. IEEE, 2020, pp. 1–7.

[20] B. Boi, F. Cirillo, M. De Santis, and C. Esposito, "Soulbound tokens: Enabler for privacy-aware and decentralized authentication mechanism in medical data storage," *Blockchain in Healthcare Today*, vol. 7, pp. 10–30 953, 2024.

[21] C. Huang, R. Song, S. Gao, Y. Guo, and B. Xiao, "Data availability and decentralization: New techniques for zk-rollups in layer 2 blockchain networks," *arXiv preprint arXiv:2403.10828*, 2024.

[22] A. Iftikhar, K. N. Qureshi, F. B. Hussain, M. Shiraz, and M. Sookhak, "A blockchain based secure authentication technique for ensuring user privacy in edge based smart city networks," *Journal of Network and Computer Applications*, vol. 233, p. 104052, 2025.

[23] I. T. Javed, F. Alharbi, B. Bellaj, T. Margaria, N. Crespi, and K. N. Qureshi, "Health-id: A blockchain-based decentralized identity management for remote healthcare," in *Healthcare*, vol. 9, no. 6. MDPI, 2021, p. 712.

[24] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, "Proverif 2.00: automatic cryptographic protocol verifier, user manual and tutorial," *Version from*, vol. 16, pp. 05–16, 2018.

[25] I. Cervesato, "The dolev-yao intruder is the most powerful attacker," in *16th Annual Symposium on Logic in Computer Science—LICS*, vol. 1. Citeseer, 2001, pp. 1–2.

[26] M. Dieye, P. Valiorgue, J.-P. Gelas, E.-H. Diallo, P. Ghodous, F. Biennier, and E. Peyrol, "A self-sovereign identity based on zero-knowledge proof and blockchain," *IEEE Access*, vol. 11, pp. 49 445–49 455, 2023.

[27] N. Naik and P. Jenkins, "Your identity is yours: Take back control of your identity using gdpr compatible self-sovereign identity," in *2020 7th International Conference on Behavioural and Social Computing (BESC)*, 2020, pp. 1–6.