HALP: A Hybrid Anonymous Login Protocol

by

Zunaed Sazzad Tonay
22101416
Elham M. Ajmotgir
22301220

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

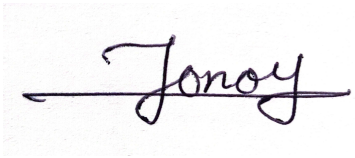
Department of Computer Science and Engineering
Brac University
Month Year.

Declaration

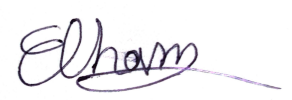
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Zunaed Sazzad Tonay
22101416



Elham M. Ajmotgir
22301220

Approval

The thesis/project titled "HALP: A Hybrid Anonymous Login Protocol" submitted by

1. Zunaed Sazzad Tonay (22101416)
2. Elham M. Ajmotgir (22301220)

of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science in 4th Year.

Examining Committee:

Supervisor:
(Member)

Dr. Md Sadek Ferdous

Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineeringt
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

This research demonstrates how to design and implement the Hybrid Anonymous Login Protocol (HALP), a protocol that tries to overcome the main shortcomings of the existing privacy-preserving authentication systems. The reliance on fixed identifiers in conventional authentication systems implies that the associated attributes of user-privacy are exposed to considerable degrees of tracking and profiling threats as long as applications are regarded within the realm of the digital environment. HALP combines Zero-Knowledge Proofs (ZKPs) with pseudonymous identifiers and selective revocable credential revocation protocols to accomplish unlinkability, scalability and a high quality of privacy protection. The ability to login to the system without revealing personally identifiable information allows the user to maintain their privacy and at the same time provides the ability to deactivate compromised credentials. HALP is designed modularly and conforming to decentralized identity specifications (most notably W3C Verifiable Credentials) to enable it to be used within a decentralized applications (DApps), and other privacy-oriented applications, including in healthcare, e-voting, and decentralized finance (DeFi). Besides, the framework is thoroughly tested on performance and scalability, as well as benchmarking tests.

Keywords: Zero-Knowledge Proofs (ZKPs),Hybrid Anonymous Login Protocol (HALP),Pseudonymous Identifiers,Selective Revocation,Unlinkability,zk-SNARKs, Anonymous- Credentials.

Acknowledgement

I am also grateful of the assistance and advice I got out of my principal supervisor Dr. Md Sadek Ferdous Sir, who played a critical role in guiding the direction towards which this research had to be taken through his expertise knowledge as well as encouragement. I also want to thank to, Yeasin Ali sir and Istiaque Ahmed sir, whose help was very important in the course of development of the project. Besides, I would like to thank my family who has been extremely supportive and understanding and never ever doubted me. Their support was undoubtedly very instrumental in facilitating the accomplishment of this thesis.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In today's digital age, user privacy is under constant threat. As surveillance has become a normal part of the human condition, centralized identity systems enable the constant tracking of users. Traditional authentication systems make it easy for both companies and hackers to track and link what a person does across different websites or apps over time. User profiles are now created faster and with more granular details than ever before. So undoubtedly the most obvious target of all this is our right to privacy. Because this loss of privacy threatens civil liberties. It also discourages people from engaging in sensitive online activities. These activities include whistleblowing, confidential communications, and secure voting.
Anonymous authentication allows individuals to access online services then pass some kind of test to prove that they have access to the correct credentials-but not who they might be [5]. Unlike conventional systems where authentication and identification are tightly coupled, anonymous login mechanisms decouple these layers, enhancing user privacy while still enforcing access control That is why this is ideal to use in anything such as the anonymous chat forums, secure online voting, and decentralized finance (DeFi) where privacy is essential.
With the rapid advancement of privacy-enhancing technologies, we now think of login systems that allow an individual to remain untraceable, unlinked across their multiple usages, and only disclosure their identity when there is truly no choice. There are tools that are assisting bringing this type of secure, privacy-protecting access to reality, such as Zero-Knowledge Proofs (ZKPs), Anonymous Credentials (e.g., Idemix or U-Prove), or Blind Signatures[4], [5].

1.1 Background

Overall, the proposed system is concentrated on the privacy-preserving authentication in which user security will be advanced without loss of anonymity. The core behind it is the use of such technologies as Zero-Knowledge Proofs (ZKPs), Anonymous Credentials, and Pseudonymous Identifiers. Zero-Knowledge Proofs, and zk-SNARKs in particular, is a cryptographic solution that allows one to prove possession of some information in a way that does not compromise any other secrets or sensitive data. They are of great importance to systems that need fast and small proofs of the high-throughput and small-latency authentication protocols, e.g., decentralized finance (DeFi) and e-voting systems. They help to provide unlinkability, thus actions of users cannot be linked between sessions or services. Selective dis-

closure of credentials will also be achievable through the zk-SNARK protocol to be incorporated on the system to reduce the amount of information revealed depending on the nature of the required service.
The protocol proposed in this paper is a development of previous work on privacy-preserving identity management because it proposes a hybrid model that has the advantages of centralized-model systems as well as decentralized technologies. In essence, the framework is based on Anonymous Credentials so that users can establish the particular identifying characteristic like the age or the citizenship and, at the same time, hide their other personal details. An extra protection is provided by blind signatures, which have the property (e.g. in the Idemix system) that the issuer cannot re-link a credential with its holder. As such, the design is useful in addressing the tracking and profiling hazards of the conventional centralized infrastructures. However, scalability issues on current systems as the one of Idemix and zkLogin experimentally revealed: they both present significant computation overhead and do not provide effective revocation with separate and distinct mechanisms. As an alternative, HALP (Hybrid Anonymous Login Protocol) comes into play by overcoming these limitations with the help of a centralised and decentralised technique combination.
The suggested system utilizes Pseudonymous Identifiers and Nullifiers, which can be acquired to strengthen anonymity and assist in the effective denial of credentials.Pseudonyms are used dynamically for each session to prevent cross-session monitoring, and nullifiers enable the revocation of compromised credentials without disclosing the identity of the user. . These methods fulfill the followings desired principles of unlinkability, and minimum disclosure, and main requirements of modern privacy-preserving systems. The framework also allows on-chain as well as off-chain revocation mechanisms, which facilitate freedom based on platform-specific applications.
The key tools and technologies to be used in its implementation are Zero-Knowledge protocols Circom as a ZKP circuit generator, SnarkJS as an engine to generate proofs, and decentralized identifiers (DIDs) to associate credentials with self-sovereign identities. The system combined with a modular architecture will be able to interface with existing specifications of decentralized identity like W3C Verifiable Credentials and remain scalable to real-life applications. Through the use of robust cryptographic protocols and associated infrastructure, the suggested architecture seeks to balance robust privacy with efficient and safe authentication.

1.2 Motivation

Although systems such as Idemix and zkLogin provide high privacy guarantees, they are either not very scalable or do not support efficient means to remove bad- behaving users without breaking anonymity[5], [14]. Furthermore, a large number of existing solutions are based on centralized trust models or require the use of additional computational load in real-world environments such as federated login or distributed identity networks[14]. The recent cases (abuses of anony- mous credentials in anonymous QA services or decentralized markets) support the necessity of the balance between privacy and responsibility. It is becoming urgent to have selective revocation through which a certain pseudonymous iden- tity can be banned without compromising their identity or putting other people at risk [15].

Anonymous authentication also separates the identity of a user and his authen- tication credentials so that people can demonstrate they have a right to use a service without having to identify themselves. This is an important part of pri- vacy preserving systems in the fields where confidentiality of information needs to occur. The past studies identify the need of anonymous authentication on the following sectors: **Secure E-Voting:** It will provide anonymous authentication, which means that voters can assert their qualification to vote without any reference to their identity, stopping coercion and vote-purchasing [16]. **Healthcare:** Privacy-friendly access to health records must be possible in medical systems because a person should have a way to verify his or her eligibility to view records without revealing any sensitive health information[23]. **Decentralized Finance (DeFi)** The DeFi sector stands out as privacy- preserving transactions plays a critical role because the Know Your Customer (KYC) policies need to ensure that transactions stay anonymous without re- vealing delicate financial information [17].

1.3 Problem Statement

Research challenges to privacy-preserving authentication exist that are a hindrance to its effectiveness and popularity. Systems like Tor and Zcash are effective in the case of anonymity through unlinkability but do not have sufficient forms of revocation of credentials that have been compromised without being in violation of anonymity. On the other hand, more frameworks (such as OAuth) offer powerful revocation features, but cannot avoid using persistent identifiers. Scalability also makes it even more complex: idemix for example has high computation overhead and thus inappropriate to run large scale deployments. In addition, several current systems rely on centralized trust thus bringing about more risk and compromising on user sovereignty. Moreover, inadequate modularity of the existing frameworks hinders their support of decentralized frameworks like blockchain, which limits their use in decentralized applications (DApps) and such new industries as decentralized finance (DeFi). All these difficulties demonstrate the need of a more flexible and efficient approach. In 2.1 we can see the limitations of the existing reasearch.
Unlinkability vs. Revocation Trade-off The unlinkability vs. revo- cation trade-off is one of the most acute problems in the modern systems of preserving privacy during the authentication. A significant number of systems providing unlinkable authentication (e.g., Tor, Zcash) guarantee that the user privacy is achieved by separating actions of a user in different sessions. But they do not have an effective means through which they can avert the compromised credentials without violating the anonymity of the users. In contrast, systems such as OAuth support strong revocation but do this at the annus horribilis of privacy, by associating the sessions using identifiers that are persistent, like emails or IP addresses.An example of such a system is Zcash, which also provides unlinkability through the use of zk-SNARKs, but has no means of efficiently re- voking compromised credentials, a major weakness against malicious actors [23].
Scalability Issues in Existing Systems: Scalability is another major issue of current systems. As an example, one of the most applicable privacy- preserving authentication frameworks, Idemix, implements Camenisch-Lysyanskaya (CL) signatures which have the advantage of supporting selective disclosure but are very computationally expensive and require complicated setups. That ren- ders Idemix

unpractical in large systems that might have hundreds of thousands and millions of users.For instance, Idemix's high computational overhead due to CL-signatures makes it unsuitable for applications where fast verification is critical, such as large-scale web platforms or real-time services [12], [23].

Trust Assumptions in Current Systems: The other significant con- straint is the centralized assumptions of trust of various authentication systems. As an example, U-Prove (Microsoft) uses a central issuer, which generates a sin- gle point of failure between authentication. By analogy, zkLogin which can be connected to commercial identity providers such as Google and Facebook, is sus- ceptible to span-across tracking and may result in the centralization of control over user authentication data [19], [33]. As an illustration U-Prove is based on a centralized issuer which compromises user sovereignty because it introduces the real possibility of compromising or manipulating credentials at the center hub point of control.

Lack of Modularity in Existing Frameworks: Current privacy-preserving authentication systems are usually monolithic and d:o not provide flexibility to work with recently developed decentralised platforms. As an example, the Idemix can not be compatible with the Ethereum and other blockchain-type systems, which is not suitable to decentralized applications (DApps) [16]. The challenge imposed by the lack of a straightforward way to integrate with the new smart contracts and blockchains cripples the applicability of such systems in contemporary decentralized settings. Example: The Idemix protocol is too complex and non-compatible with the blockchain protocols such as Ethereum, so it cannot be used to implement decentralized finance (DeFi), as well as other promising applications that enhance use of Idemix functionality in practice [12].

1.4 Objective

RO1. Develop Design of a Hybrid Anonymous Login Protocol (HALP) Ar- chitecture: Design a scalable, modular system of anonymous logins based on zero-Knowledge Proofs (ZKPs), single-use pseudonyms, and cryptographic nul- lifiers that are both on-chain and off-chain revocation registries.

RO2. Obtain Scalable and Secure Unlinkability: Make sure every login is unlinkable, by creating new pseudonyms in each session, and ensure scalability of the system and high throughput in real-time high-density applications.

RO3. Enter Selective Revocation Mechanisms: Establish a scalable system of selective revocation, so that compromised or mischievous pseudonyms may be blacklisted without sacrificing the privacy of the other users, with cryptographic nullifiers to secure and scale revocation.

RO4. Optimization of system to be used in the real world: Compare the performance of the benchmarked system, HALP, with other privacy-preserving systems, under the metrics of latency, throughput and computational performance, so that the proposed system does well when used in real-time authentication systems.

RO5. Conduct Formal Privacy and Security Analysis: Demonstrate that HALP meets privacy properties such as unlinkability and revocation soundness using formal cryptographic arguments that resists various typical attacks, including Sybil, replay, and forgery attacks.

RO6. Develop and Evaluate an Open-Source Prototype: Develop an open- source reference implementation in Rust/TypeScript that will implement cre- dential is-

suance, ZKP-based login and the revocation management, and real- world performance parameters, such as the gas cost, the size of proofs and their scalability.

1.5 Methodology in Brief

The current research project will take on the scalability, the unlinkability, and the revocation constraints of current privacy-preserving authentication schemes the most notable of which is Idemix and zkLogin. Critical literature review is done to analyze the advantages and deficiencies of approaches like Zero-Knowledge Proofs (ZKPs) and anonymous credentials in order to make policy decision of structuring the Hybrid Anonymous Login Protocol (HALP). The HALP aims at achieving unlinkability and selective revocation, as well as performance optimisation.
The methodology (Figure 3.1) involves a scientific evaluation of functional and non-functional requirements i.e. secure issuance of credentials, scalability etc that need to be performed in conjunction with a threat model based on the STRIDE structure to check the possible security pitfalls. The outcome of system architecture is the incorporation of ZKPs, pseudonyms, and nullifiers through which the system achieves scalable and modular authentication properties of on-chain and off-chain revocation paths. The property of the system to preserve data privacy will be checked in a comprehensive security analysis, and an open-source realisation of a prototype in Rust/TypeScript will be accomplished. Gas expenditure and overall performance are some of the measures in which this prototype will be benchmarked against.

1.6 Scopes and Challenges

This study examines design, implementation and the effectiveness of the Hybrid Anonymous Login Protocol (HALP). This protocol combines Zero-Knowledge Proofs (ZKPs), one-time pseudonyms and cryptographic nullifiers around the task of developing an authentication protocol that would be scalable, unlinkable and revocable at the same time. HALP aims to alleviate some of the privacy and scalability limitations imposed on existing systems and, by protecting un- linkability, to allow revocations on demand, and to avoid the need to assume centralised trust. More so, it will be in real-time optimisation thus, it minimizes 12 delay and maximising big-scale throughput. To this effect, an open-source pro- totype will be created and benchmark tested against the current solutions so as to gauge performance, scalability, and compatibility with standardised decen- tralised identity protocols like W3C Verifiable Credentials and DID-Core. There are numerous challenges that have to be overcome in order to allow halp, a scalable and high-throughput credential-based authentication protocol to achieve practical adoption. On the one hand, zero-knowledge proofs (zk- proofs) are very efficient to verify yet introduce a computational complexity to generate such a proof. To find the right balance between powerful performance and privacy is thus a vital design challenge. Second, due to large user traffic, halp needs to have low latency, and this increases scalability issues. Third, safe selective revocation schemes are a requirement and they should not affect the privacy of the users. Fourth, using blind signatures, nullifiers, and zk- SNARKs would require quite a bit of cryptographic complexity, which in turn would require extensive security testing. Fifth, achieving interoperability with the

current systems and decentralized identity standards are challenging since most current systems are centralized. Lastly, decentralized trust that can be made without compromising on security should be developed and the difference will be whether it is used or not, which will depend on how usability issues are addressed especially to end users in privacy-sensitive areas. Despite these difficulties, halp can be used to significantly improve privacy and scalability when it comes to current authentication systems.

1.7 Research Structure

- In Chapter 2, we present the background terminologies and literature review of the relevant research works.
- Chapter 3 establishes HALP's foundation through research methodology, system architecture, functional/security requirements (anonymous authentication, unlinkability, revocation), and examines societal impacts, ethical considerations, and project management.
- In Chapter 4 we conclude our pre thesis 1 report and discuss about the future works.

Chapter 2

Literature Review

The chapter reviews the basics covering the principles and available frameworks, which are relevant to privacy-preserving authentication. The key concepts, i.e., unlinkability, minimal disclosure, and selective revocation are discussed and emphasized as being important to anonymous login systems. Existing methods, such as Zero-Knowledge Proofs (ZKPs), blind signatures, and anonymous credentials, are compared in each of the following points in respect of the advantages and limitations of each method with regard to the preservation of privacy and scalability. The discussion is then focused on the major systems like Idemix, zkLogin and U-Prove with their deficiencies towards scalability and revocation capability being outlined. The discourse climaxes in the comparative analysis where gaps have been found in the existing methods, which the presented Hybrid Anonymous Login Protocol (HALP) is keen to address. This discussion plays a key role in the run-up towards presenting HALP at detail in the upcoming chapters.

2.1 Preliminaries

Authentication refers to a procedure with which a system confirms the identity of a user. It is usually done with such techniques as usernames and passwords, biometrics (fingerprints, face recognition) and multi-factor authentication (MFA). Nevertheless, the existing authentication mechanisms are also coupled with serious issues of privacy. During log in the system usually captures identifiable data (e.g. email, username) and this becomes a means of tracking the user or profiling the user amongst a multiplicity of services. This is a privacy compromise as it doles out more information than what is necessary concerning the user To address this, the concept of anonymous authentication came up. In anonymous authentication, the user can demonstrate his or her credibility to view a service without submitting personal data. This is vital in the security of user privacy especially in the sensitive areas such as healthcare, e-voting and whistleblowing sites [37].

2.1.1 Privacy Requirements in Anonymous Authentication

2.1.1.1 Unlinkability

Unlinkability guarantees that the same user's identity cannot be tracked across different login sessions. For example, when Alice logs into the system in two different

situations, for a consultation and for a prescription refill, the system will not be able to match these two sessions with Alice, even if they are the same person. This means that even if the same user logs into a system repeatedly, she will not have any linkability. This is accomplished through means such as ephemeral key rotation (keys rotating between sessions) and zero-knowledge proof (ZKP) authentication, which allows authentication without leaking identifying characteristics[37]. Mathmatically unlinkability is defined as the inability to link items or actions based on available evidence. This concept is formalized through equivalence relations, probability distributions, and entropy measures.

Unlinkability within One Set: Let $A = \{a_1, \ldots, a_n\}$ be a set of items in a given system. We model the unlinkability of items within the set using an equivalence relation $\sim_{r(A)}$, which partitions A into equivalence classes $A_1, \ldots, A_l$, where:

$$A = A_1 \cup A_2 \cup \cdots \cup A_l \quad \text{and} \quad A_i \cap A_j = \emptyset \quad \forall i \neq j.$$

Two items $a_i, a_j \in A$ are *related* if $a_i \sim_{r(A)} a_j$, and they are *unrelated* if $a_i \not\sim_{r(A)} a_j$.

Degree of Unlinkability (Two Items): The *degree of unlinkability* between two items a_i and a_j in A is measured by the attacker's *a posteriori* probability of their relationship, given some evidence:

$$d(i,j) = -P(a_i \sim_{r(A)} a_j) \log_2 P(a_i \sim_{r(A)} a_j) - P(a_i \not\sim_{r(A)} a_j) \log_2 P(a_i \not\sim_{r(A)} a_j).$$

where:

- $P(a_i \sim_{r(A)} a_j)$ is the probability that a_i and a_j are related.
- $P(a_i \not\sim_{r(A)} a_j) = 1 - P(a_i \sim_{r(A)} a_j)$ is the probability that a_i and a_j are not related.

The degree of unlinkability ranges from:

$$d(i,j) = 1 \quad \text{(maximal unlinkability, i.e., } P(a_i \sim_{r(A)} a_j) = P(a_i \not\sim_{r(A)} a_j) = \frac{1}{2}),$$

to:

$$d(i,j) = 0 \quad \text{(full linkability, i.e., } P(a_i \sim_{r(A)} a_j) = 1 \text{ or } P(a_i \sim_{r(A)} a_j) = 0).$$

Unlinkability of Arbitrary Many Items Let $\{a_1, a_2, \ldots, a_k\}$ be a subset of A containing k items. The equivalence relation $\sim_{r(A)}$ on A induces a relation on the subset $\{a_1, a_2, \ldots, a_k\}$. The probability that the relation $\sim_{r(A)}$ restricted to $\{a_1, a_2, \ldots, a_k\}$ is the same as the original relation is denoted by:

$$P(\sim_{r(A)} | \{a_1, \ldots, a_k\}) = P(\sim_{r(A)}) \quad \text{(relation distribution remains unchanged)}.$$

Attacker Models Two types of attacks are considered for unlinkability:

1. **Existential break**: Any pair of items $(a_i, a_j) \in A$ experiences a change in their linkability probabilities due to the attacker's actions.
2. **Selective break**: A subset of items is chosen by the attacker, and the linkability probabilities between one item in the subset and all other items are altered.

Unlinkability Between Sets In some systems, items are linked to elements from a separate set, such as a set of users $U = \{u_1, \ldots, u_n\}$ and actions $A = \{a_1, \ldots, a_m\}$. We define an equivalence relation $\sim_{r(U,A)}$ that links the actions in A to users in U, forming equivalence classes on A based on the users who performed the actions.
For example, in a communication system, the set A represents messages sent, and U represents users. A message sent by user u_i is related to u_i but should not be linkable to u_j for $j \neq i$ from the attacker's perspective.

Optimal Unlinkability For an equivalence relation $\sim_{r(A)}$ on a set A of size $|A| > 1$, the degree of unlinkability between any two items in A cannot be guaranteed to be $d(i, j) = 1$ for all pairs if the sizes of the equivalence classes are known. Thus, perfect unlinkability is not achievable if the structure of the equivalence classes is revealed.

Anonymity via Unlinkability Anonymity is refined as unlinkability between subjects and actions. This is commonly applied in cryptographic protocols to protect privacy:

- **Sender/Receiver anonymity**: No message is linkable to any sender or receiver.
- **Relationship anonymity**: Unlinkability of "who communicates with whom".

Degree of Anonymity via Unlinkability Let $U = \{u_1, \ldots, u_n\}$ be a set of users, and $A = \{a_1, \ldots, a_m\}$ a set of actions. Anonymity can be defined by the unlinkability between elements in these sets. The degree of anonymity of a user u_i with respect to an action a_j is given by the degree of unlinkability between those items.
For two items $a_i \in A$ and $u_j \in U$:

$$d(a_i, u_j) = -P(a_i \sim_{r(A,U)} u_j) \log_2 P(a_i \sim_{r(A,U)} u_j) - P(a_i \not\sim_{r(A,U)} u_j) \log_2 P(a_i \not\sim_{r(A,U)} u_j),$$

where $P(a_i \sim_{r(A,U)} u_j)$ and $P(a_i \not\sim_{r(A,U)} u_j)$ are the probabilities of the relationship between the action and the user being related or unrelated, respectively [8].

2.1.1.2 Minimum Disclosure

Minimum Disclosure is a privacy principle which ensures that users provide only the necessary information required to establish a claim, avoiding the sharing of extraneous details. This principle is crucial in maintaining user privacy, particularly in authentication scenarios.[8].

Mathematical Definition of Minimum Disclosure Let $U = \{u_1, u_2, \ldots, u_n\}$ represent the set of users, and $C = \{c_1, c_2, \ldots, c_m\}$ represent the set of credentials or attributes associated with users. Suppose a system verifies a claim Q (such as proving that a user is over 18 years old), with the goal of minimizing the disclosed information. The principle of minimum disclosure can be formally expressed as:

$$\text{Claim} : Q(u_i) \quad \text{for} \quad u_i \in U$$

where the disclosed information $D(u_i)$ should satisfy:

$$D(u_i) \subseteq C \quad \text{and} \quad D(u_i) \text{ is minimal such that } Q(u_i) \text{ holds.}$$

Thus, the disclosed information is minimized to only the necessary attribute(s) $c_k \in C$, such that:

$$D(u_i) = \{c_k | Q(u_i) \text{ holds with } c_k\}.$$

The principle ensures that unnecessary attributes (e.g., full birthdate) are not disclosed, and only the proof (e.g., proof of age) is shared, in compliance with data minimization standards (e.g., GDPR).

Implication for Data Minimization Let $\mathcal{D}$ represent a data collection set. The system must guarantee that for any claim $Q(u_i)$, the data collected is:

$$\mathcal{D} = \{c_k \in C | c_k \in D(u_i)\}.$$

where:

$$\forall c_k \in C \quad c_k \notin D(u_i) \quad \text{unless required by} \quad Q(u_i).$$

2.1.1.3 Selective Revocation

There should be revocation mechanisms in case a user misbehaves or his/ her credential is compromised. The system will have to revoke the credential (or pseudonym) of a user without interfering with the privacy of other users. This may be done with either nullifiers (which are tokens based on credentials which have been revoked) or accumulating (where revocations are handled with cryptographic sets). Newer schemes, such as threshold revocation based on Publicly Verifiable Secret Sharing (PVSS), decentralize revocation control to prevent single points of failure [39].

Mathematical Definition of Selective Revocation Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of pseudonyms or credentials associated with users, and let $R = \{r_1, r_2, \ldots, r_m\}$ be a set of revocation tokens. Selective revocation can be expressed through a function $\text{Rev}(p_i)$ that allows the revocation of a specific pseudonym p_i without affecting the privacy of other users. This can be modeled as:

$$\text{Rev}(p_i) = \text{Nullifier}(p_i) \quad \text{for} \quad p_i \in P$$

where the *nullifier* $\text{Nullifier}(p_i)$ is a cryptographic token that is associated with the revoked credential, ensuring that the revocation does not leak additional information about the user or their other credentials.

Threshold Revocation with Publicly Verifiable Secret Sharing (PVSS) In more advanced schemes, such as threshold revocation, the control over revocation is distributed, ensuring no single point of failure. Let S represent the secret shares, and $T \subseteq S$ represent the threshold subset of shares required to revoke a credential. The threshold revocation scheme is modeled as:

$$\text{Rev}(p_i) = \text{Threshold}_T(S) \quad \text{where} \quad T = \{t_1, t_2, \ldots, t_k\} \subseteq S.$$

This ensures that revocation is only possible when the required threshold number of shares T is reached, thereby decentralizing control over the revocation process.

Implication for Privacy The selective revocation scheme ensures that:

$$\text{Rev}(p_i) \quad \text{does not leak information about} \quad P \setminus \{p_i\}.$$

Thus, the revocation of one credential or pseudonym does not expose the private information of other users or their credentials [8].

2.1.2 Cryptographic Building Blocks

2.1.2.1 Zero-Knowledge Proofs (ZKPs)

Zero-Knowledge Proofs (ZKPs) are cryptographic protocols, which enable a prover to certify that he or she knows some particular piece of information (e.g. a valid credential or a mathematical statement) to a verifier, without leaking any further information about the information. This is particularly important in privacy-preserving systems, where authentication of legitimacy must be compatible without the exposure of any sensitive information.

Core Concepts

I. Completeness: When the statement is true, honest verifier will be convinced by the honest prover.

$$P(\text{Verifier accepts} \mid \text{Statement is true}) = 1 \tag{2.1}$$

It follows that there is probability 1 when the statement is true that the verifier accepts the proof.

II. Soundness: In case the statement is false, no cheating prover can make the verifier believe it to be true.

$$P(\text{Verifier accepts} \mid \text{Statement is false}) \leq \epsilon \tag{2.2}$$

Where is a small probability (usually called negligible). This means that there is very little chance of a dishonest prover succeeding to convince the verifier.

III. Zero-Knowledge: The only thing that the verifier does learn is nothing more than whether the statement is true. The verifier does not get to know any more information about the underlying data[19][11].

$$\text{Verifier's knowledge} = \emptyset \tag{2.3}$$

The communication between prover and verifier does not reveal more than just the truth of the statement.

2.1.2.1.2 zk-SNARKs

zk-SNARKs(SuccinctNon-InteractiveARguments of Knowledge) are one of the most popular types of ZKPs applied in modern systems; they allow generating proofs and verifying them rather efficiently. They are concise, i.e. the size of proof is small (approximately 200 bytes), and the verification is extremely fast (in the order of 300ms). This has made zk-SNARKs suitable in real-time authentication systems, like in the one we are suggesting in HALP, where fast verification is desired due to the user experiences on systems like this needs to be smooth [14].

Although zk-SNARKs are efficient, they do require a trusted setup stage, the compromise of which may be a vulnerability. Such trade-offs are also what makes zk-SNARKs different to other ZKP systems such as Bulletproofs, which only require a non-trusted setup but are associated with even greater proof sizes [21]. This kind of trade-off is systematically addressed in HALP via the Circom, ZKP circuit designing framework, and SnarkJS, a proof generation library. These tools accelerate the use of zk-SNARKs in the efficient implementation of HALP at the same time making the system fast and secure[14].

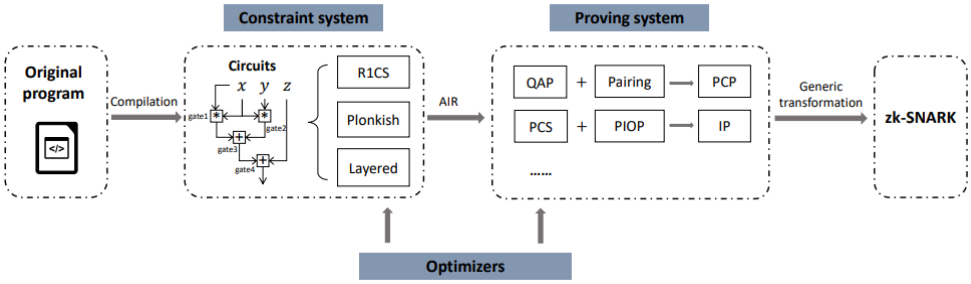


Figure 2.1: General steps of converting a high-level program to a zk-SNARK

The above diagram (Figure 2.1) describes the procedure of how to encode a high-level program into zk-SNARK (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge). The process of the work starts with the assemblage of the initial program in the form of a circuit. It is then placed in a constraint structure, where it is turned into a mathematical representation of the computation by the application of techniques like R1CS, Plonkish, or Layered circuits. The circuit obtained is subjected to a proving system, adopting methods such as QAP, PCS, pairing, or PIOP+IP to prove its correctness. Then, a number of optimizations are additionally carried out to make the proof more efficient (shorter proof size and faster to verify). The sequence is concluded by a generic transformation that turns the proof into a zk-SNARK and achieves the desired result, efficient, succinct, and non-interactive verification (and thus finishes turning a high-level program into a cryptographically verifiable program).

Formal Languages A formal language L is a subset of the set of all finite strings Σ^*, where Σ is an alphabet and Σ^* is the set of all possible finite strings (including the empty string). The alphabet Σ consists of symbols or letters, and words in L are strings of symbols formed according to a grammar.

Let Σ be a finite alphabet and Σ^* be the set of all strings over Σ. A *formal language* L is a subset of Σ^*, i.e., $L \subseteq \Sigma^*$.

Decision Functions A decision function determines whether a given string belongs to a formal language. It maps strings from Σ^* to a boolean value indicating membership in the language.

A *decision function* R for a language L is a function:

$$R : \Sigma^* \to \{\text{true}, \text{false}\},$$

where $R(x) = \text{true}$ if $x \in L$ and $R(x) = \text{false}$ otherwise.

The associated language L_R of a decision function R is:

$$L_R := \{x \in \Sigma^* \mid R(x) = \text{true}\}.$$

Instance and Witness In zero-knowledge proofs, an *instance* is a publicly known input to a statement, while a *witness* is a private input that helps to prove the truth of the statement.
Let Σ_I and Σ_W be two alphabets, where Σ_I represents the instance alphabet and Σ_W represents the witness alphabet. The *decision function* R is defined as:

$$R : \Sigma_I^* \times \Sigma_W^* \to \{\text{true}, \text{false}\}, \quad (i, w) \mapsto R(i, w),$$

where $i \in \Sigma_I^*$ is an instance and $w \in \Sigma_W^*$ is a witness. The associated language L_R is:

$$L_R := \{(i, w) \in \Sigma_I^* \times \Sigma_W^* \mid R(i, w) = \text{true}\}.$$

Rank-1 Quadratic Constraint Systems (R1CS) A Rank-1 Quadratic Constraint System (R1CS) is a system of quadratic equations over a finite field that represents a computation or relation. It consists of a set of constraints that must be satisfied by an instance and witness pair.
A *Rank-1 Quadratic Constraint System (R1CS)* is a set of k equations of the form:

$$\left(a_1 + \sum_{j=1}^{n} a_j I_j + \sum_{j=1}^{m} a_{n+j} W_j\right) \cdot \left(b_1 + \sum_{j=1}^{n} b_j I_j + \sum_{j=1}^{m} b_{n+j} W_j\right) = c_1 + \sum_{j=1}^{n} c_j I_j + \sum_{j=1}^{m} c_{n+j} W_j,$$

where $I_1, \ldots, I_n$ are instance variables, $W_1, \ldots, W_m$ are witness variables, and $a_i, b_i, c_i \in F$ for $i = 1, 2, \ldots, n + m$ are constants in the finite field F.

R1CS Satisfiability An R1CS is satisfiable if there exists an assignment of values to the instance and witness variables that satisfies all the constraints in the system. The *R1CS satisfiability* problem is the decision problem of determining whether there exists a solution $(I, W) \in \Sigma_I^* \times \Sigma_W^*$ such that:

$$R_{\text{R1CS}}(I, W) = \text{true},$$

where R_{R1CS} is the decision function associated with the R1CS.

Quadratic Arithmetic Program (QAP) A *Quadratic Arithmetic Program (QAP)* is a generalization of the R1CS, where the constraints are quadratic polynomials over a finite field. These programs are used to describe computations in zero-knowledge proofs.
A *Quadratic Arithmetic Program (QAP)* consists of a set of polynomials $P_1(x), P_2(x), \ldots, P_k(x)$ over a finite field F, where each polynomial $P_i(x)$ describes a constraint in the program. The QAP is represented as:

$$P_i(x) = A_i \cdot B_i - C_i, \quad \text{for} \quad i = 1, 2, \ldots, k,$$

where $A_i, B_i, C_i \in F$ are polynomials over the field, and the program defines the set of valid assignments to the instance and witness variables.

QAP Satisfiability QAP satisfiability refers to the problem of determining whether there exists an assignment to the instance and witness variables that satisfies all the polynomial equations in the QAP.
The *QAP satisfiability* problem is the decision problem of determining whether there exists an assignment $(I, W) \in \Sigma_I^* \times \Sigma_W^*$ such that:

$$R_{\text{QAP}}(I, W) = \text{true},$$

where R_{QAP} is the decision function associated with the QAP [25].

2.1.2.2 Anonymous Credentials

Anonymous credentials [5] enable users to prove their identity without disclosing it, and it is the key capability of privacy-preserving authentication systems. A CamenischLysyanskaya (CL) signature is a widely known example of an anonymous credential system whose construction builds off the use of a variant of public key encryption which supports selective disclosure of one attribute without disclosure of others. An example is that a user can be able to prove that he is over 18 years old without publishing their age number[15]. Hyperledger Ursa Coconut scheme [23] is a threshold issuance and selective disclosure of credentials used in HALP. Threshold issuance enables credential issuance to be distributed, so a single authority can not take full control over the credentialing process which increases the trust level and also makes the system more secure and reliable. Selective disclosure implies that users may not have to disclose their whole credential (such as the correct date of birth) but can release the certain characteristics of their credential (such as the fact that they are over 18). In contrast to the U-Prove with centralized issuers, Coconut does not require a single and centralized issuer, and the trust is spread among entities. This distributed model decreases the chance of a common point of failure, and has a more resistant structure against assaults or abuse

2.1.2.3 Pseudonymous Identifiers & Nullifiers

Pseudonymous identifiers [19] play an important role in privacy enhancing authentication systems where they enable the privacy of users of the systems and avoid tracking in case of multiple sessions. Rather than applying true, persistent identifiers (e.g., email addresses or usernames), such systems are pseudonymous. A pseudonym is a new identifier issued specifically to a single session or interaction, where it is generated as a cryptographic hashing of the secret credentials of the user, and some form of randomization, which may be a nonce (a one-time per-interaction randomly-generated value). Such a process guarantees that as long as a pseudonym is intercepted or acquired by a third party, it remains completely anonymous with the user that only possesses a secret key or understands the requirement of the first credential.

Core Concepts of Pseudonymous Identifiers: The main idea of pseudonym usage is to avoid connecting identities between sessions to guarantee privacy with the ability to authenticate. Essentially a pseudonym is a temporary, identifying text that can be associated with a particular action or session. Notably, pseudonyms are cryptographically derived and may be based on a master secret (e.g., a private key)

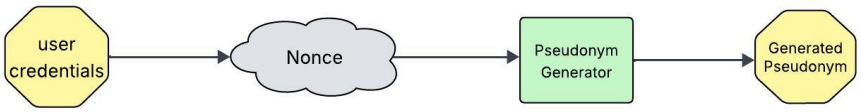


Figure 2.2: Pseudonym Generation

with an appendage known as a nonce ensuring that each time a user logs on a different pseudonym will be produced.

Pseudonym Generation: pseudonyms are created by applying a nonce on a secret credential (normally a cryptographic key) of a user (Figure 2.2). This would give every session a distinct identifier and ensure that there is no identity relatability between sessions.

Linkability: Although the same person might use several pseudonyms across different time intervals, it would be impossible to connect the same to that individual unless a Priv key or some other sensitive information is accessed. This helps in keeping the activity of the user unlinkable between sessions [33].

HALP employs nullifiers to handle revocation in ways that do not interfere with privacy. A nullifier is a cryptographic tag produced on a credential usage or revocation. It blocks reuse of credential once it has been used. Nullifiers based on secret information of the user perform the task of handling revocation without uncovering identity of the user or any other personal information. This system enables blocking malicious or compromised users efficiently because it does not violate user privacy[17]. HALP uses Zero-Knowledge Proofs (ZKPs), anonymous credentials, and one-time pseudonyms with nullifiers to achieve its main objectives of strong privacy, support scalable authentication, and revocable authentication. Those cryptographic schemes are used to solve the fundamental problems of linkability, anonymity, and revocation in privacy-preserving systems.

Unlinkability: Every time a user logs in a different pseudonym is created and hence the activities of a user cannot be linked together in different login sessions, even when using the same service provider [30].

Privacy Preservation: As pseudonyms are used, through HALP, users can authenticate themselves in a secure manner without exposing their identity and other sensitive personal information [19].

Efficient Revocation: Credential revocation is done in a privacy preserving and efficient manner using nullifiers. This makes sure that malicious user is not able to use the system by exploiting weak credentials repeatedly [21].

2.1.3 Infrastructure Components

2.1.3.1 Revocation List:

Revocation registry is essential infrastructure to handle compromised or malicious credentials, retaining privacy of legitimate users. Revocation registry helps in a way that when a user credential is compromised or revoked, the user is not capable of authenticating using the credential again leading to prevention of abuse or misuse.

On-chain Revocation: As an example of transparency, on-chain revocation makes use of the blockchain, (e.g., Ethereum smart contracts or zkRollups (e.g., Aztec)).

This makes revocations immutable because when a blockchain records a revocation, the consequence of such revocation is publicly trackable and cannot be changed using any central source. Smart contracts are applied in a transparent revocation algorithm, which allows all participants to check whether the credential is valid or not without knowing the identity of the user [14][17].
Low Latency Off-Chain Revocation: Off-chain revocation can be used to mitigate issues of scalability and performance, particularly in high-traffic systems. Revocation data is stored off-chain, usually in a database or server, attested to by signed statements that a credential was actually revoked. Off-chain systems have the benefit of low latency, since revocation can be used at a quick time without the overhead of engaging the blockchain on each and every transaction [34].
One solution where revocation is checked through attestation is by a trusted party allowing optimistic systems to exist, which is fast but has some degree of security. Revocation decisions are trusted in such a system, but can be disputed or checked against an on chain registry in case of a discrepancy [28], [34].

2.1.3.2 Decentralized Identifiers (DIDs):

Decentralized Identifiers (DIDs) represent a new kind of identifier designed to support self-sovereign identity systems, where a user is always in control of its own identifiers and the data that those identifiers reference. The use of DIDs forms a vital part of this HALP infrastructure since it ties the credentials of the user to a user-controlled identity that permits privacy-friendly authentication. DIDs bind credentials to an identity controlled by the user on a decentralized platform. And unlike the traditional identifications (like email addresses or government-issued IDs), DIDs are not connected to some centralized entity, which is why users are in full control of their personal information. This decentralization denies third parties the ability to store data and follow users or benefit directly from their data and guarantees that individuals may use authentication to access services without depending upon a core provider. DIDs frequently are complemented with Verifiable Credentials (VCs), in which users may provide selective disclosure of aspects of their credentials without divulging other sensitive data [1].

2.1.4 Performance Optimizoatin Tools:

2.1.4.1 ZK Proof Systems

Privacy-preserving authentication fundamentally requires ZK Proofs and in particular imposes performance issues to large systems. A few achievements in the ZK proof systems have been formed to make these proofs as scalable and quick as possible.

Groth16: Groth16 is a well-known zk-SNARK protocol that generates small proofs taking fast verification time (300ms). It however, needs a trusted setup, which is a security risk, unless done properly. Groth16 is suitable to problem applications such as HALP where efficient, low-latency proofs are necessary [21], [33]..

Proof Systems

Definition: Proof System A proof system consists of the following components:

- $\mathcal{P}$: The prover algorithm,
- $\mathcal{V}$: The verifier algorithm,
- $\mathcal{S}$: The simulator algorithm,
- $\mathcal{L}$: The language defined by a Rank-1 Constraint System (R1CS).

A proof system is designed such that the prover $\mathcal{P}$ convinces the verifier $\mathcal{V}$ of the validity of a statement, without revealing unnecessary information.

Definition: Completeness A proof system is **complete** if, for every instance $x \in \mathcal{L}$, the prover $\mathcal{P}$ can generate a valid proof $\pi = P(A)$ (where A is relevant portion of set of credentials $\{c_1, c_2, \ldots\}$) that the verifier $\mathcal{V}$ will always accept:

$$\Pr[\mathcal{V}(\pi) = \texttt{accept} \mid x \in \mathcal{L}] = 1.$$

This means that if the statement is true (i.e., $x \in \mathcal{L}$), the verifier will always accept the proof generated by the prover.

Definition: Soundness A proof system is **sound** if no malicious prover can convince the verifier to accept a false proof. Formally, for every $x \notin \mathcal{L}$, the probability that a dishonest prover convinces the verifier to accept a proof is negligible:

$$\Pr[\mathcal{V}(\pi) = \texttt{accept} \mid x \notin \mathcal{L}] \leq \epsilon(\lambda),$$

where $\epsilon(\lambda)$ is a negligible function of the security parameter λ.

Definition: Zero-Knowledge A proof system is **zero-knowledge** if, given a proof π, the verifier learns nothing beyond the validity of the statement being proven. More formally, there exists a simulator $\mathcal{S}$ that can generate a proof π indistinguishable from a real proof generated by the prover, without knowing the witness:

$$\text{Sim}(\mathcal{L}, x) \approx \mathcal{P}(\mathcal{L}, x, w),$$

where w is the witness, $\mathcal{P}$ is the prover algorithm, and $\mathcal{S}$ is the simulator.

Prover and Verifier Algorithms The prover $\mathcal{P}$ and verifier $\mathcal{V}$ algorithms are formally defined as:

- $\mathcal{P}(x, w)$ takes as input an instance $x \in \mathcal{L}$ and a witness w, and outputs a proof π.
- $\mathcal{V}(x, \pi)$ takes as input an instance $x \in \mathcal{L}$ and a proof π, and outputs either `accept` or `reject`.

The proof π is sent by the prover to the verifier, and the verifier uses the verification algorithm $\mathcal{V}$ to decide whether to accept or reject the proof.

Groth16 Protocol

Groth16 Parameters Let G_1 and G_2 be two cyclic groups of prime order r, with generators $g_1 \in G_1$ and $g_2 \in G_2$, and let $e : G_1 \times G_2 \to G_T$ be a bilinear map. The Groth16 protocol involves the following parameters:

- G_1, G_2: Two groups of prime order r,
- g_1, g_2: Generators of G_1 and G_2,
- e: A bilinear map,
- $\mathcal{CRS}$: The common reference string (CRS) generated during the setup phase.

Setup Phase The setup phase takes the Rank-1 Constraint System R and generates the common reference string $\mathcal{CRS}$ and the simulation trapdoor $\mathcal{ST}$. The setup algorithm is as follows:

$$(\mathcal{CRS}, \mathcal{ST}) \leftarrow \text{SETUP}(R).$$

5 random invertible elements $\alpha, \beta, \gamma, \delta$, and τ are taken from the field Fr of the protocol to produce the simulation trapdoor ST:

$$\mathcal{ST} = (\alpha, \beta, \gamma, \delta, \tau)$$

The CRS is composed of elements from G_1 and G_2, and is shared by both the prover and the verifier. Specifically, the CRS consists of two tuples:

$$\mathcal{CRS}_{\text{QAP}} = (\text{CRS}_{G_1}, \text{CRS}_{G_2}),$$

where:

$$\begin{aligned}
\text{CRS}_{G_1} = \Bigg\{ & g_1^{\alpha}, g_1^{\beta}, g_1^{\delta}, \left(g_1^{\tau^j}\right)_{j=0}^{\deg(T)-1}, \left(g_1^{\frac{\beta A_j(\tau)+\alpha B_j(\tau)+C_j(\tau)}{\gamma}}\right)_{j=0}^{n}, \\
& \left(g_1^{\frac{\beta A_{j+n}(\tau)+\alpha B_{j+n}(\tau)+C_{j+n}(\tau)}{\delta}}\right)_{j=1}^{m}, \left(g_1^{\frac{\tau^j T(\tau)}{\delta}}\right)_{j=0}^{\deg(T)-2} \Bigg\} \\
\text{CRS}_{G_2} = \Big\{ & g_2^{\beta}, g_2^{\gamma}, g_2^{\delta}, \left(g_2^{\tau^j}\right)_{j=0}^{\deg(T)-1} \Big\}
\end{aligned}$$

The elements $g_1^{\alpha}, g_1^{\beta}, g_1^{\delta}, \ldots$ are computed using the setup parameters $\alpha, \beta, \gamma, \delta \in \mathbb{F}_r$. The CRS components are shared by both the prover and the verifier.

Prover Phase The prover, given the CRS, an instance I, and a witness W, computes a proof π as follows:

$$\pi = \mathcal{P}(R, \mathcal{CRS}, I, W).$$

The proof π consists of three group elements: two from G_1 and one from G_2, which are derived based on the witness and the instance.
To compute the proof, the prover evaluates the polynomial $P(I; W)$, derived from the Quadratic Arithmetic Program (QAP). The prover then divides this polynomial by the target polynomial T of the QAP. As $P(I; W)$ is derived from a valid solution to the R1CS, the division yields a polynomial $H := P(I; W)/T$, where $\deg(H) < \deg(T)$.

The prover then evaluates the polynomial $(H \cdot T)/\delta$ in the exponent of the generator g_1 at the secret point τ. Let $H(x)$ be the quotient polynomial P/T:

$$H(x) = H_0 \cdot x^0 + H_1 \cdot x^1 + \ldots + H_k \cdot x^k$$

To evaluate $(H \cdot T)/\delta$ at τ in the exponent of g_1, the prover computes using CRS:

$$\frac{H(\tau) \cdot T(\tau)}{\delta} = \left(\frac{\tau^0 \cdot T(\tau)}{\delta}\right)^{H_0} \cdot \left(\frac{\tau^1 \cdot T(\tau)}{\delta}\right)^{H_1} \cdots \left(\frac{\tau^k \cdot T(\tau)}{\delta}\right)^{H_k}$$

Then the prover takes a sample of two random field elements $r, t \in \mathbb{F}_r$. Using CRS, instance variables $I_1, \ldots, I_n$, witness variables $W_1, \ldots, W_m$, the following is computed:

$$g_1^W = \left(\frac{\beta A_{1+n}(\tau) + \alpha B_{1+n}(\tau) + C_{1+n}(\tau)}{\delta}\right)^{W_1} \cdots \left(\frac{\beta A_{m+n}(\tau) + \alpha B_{m+n}(\tau) + C_{m+n}(\tau)}{\delta}\right)^{W_m}$$

$$g_1^A = g_1^{\alpha} \cdot \frac{A_0(\tau)}{\delta_1} \cdot \left(\frac{A_1(\tau)}{\delta_1}\right)^{I_1} \cdots \left(\frac{A_n(\tau)}{\delta_1}\right)^{I_n} \cdot \left(\frac{A_{n+1}(\tau)}{\delta_1}\right)^{W_1} \cdots \left(\frac{A_{n+m}(\tau)}{\delta_1}\right)^{W_m} \cdot \left(\frac{\delta}{\delta_1}\right)^{r}$$

$$g_1^B = g_1^{\beta} \cdot \frac{B_0(\tau)}{\delta_1} \cdot \left(\frac{B_1(\tau)}{\delta_1}\right)^{I_1} \cdots \left(\frac{B_n(\tau)}{\delta_1}\right)^{I_n} \cdot \left(\frac{B_{n+1}(\tau)}{\delta_1}\right)^{W_1} \cdots \left(\frac{B_{n+m}(\tau)}{\delta_1}\right)^{W_m} \cdot \left(\frac{\delta}{\delta_1}\right)^{t}$$

$$g_2^B = g_2^{\beta} \cdot \frac{B_0(\tau)}{\delta_2} \cdot \left(\frac{B_1(\tau)}{\delta_2}\right)^{I_1} \cdots \left(\frac{B_n(\tau)}{\delta_2}\right)^{I_n} \cdot \left(\frac{B_{n+1}(\tau)}{\delta_2}\right)^{W_1} \cdots \left(\frac{B_{n+m}(\tau)}{\delta_2}\right)^{W_m} \cdot \left(\frac{\delta}{\delta_2}\right)^{t}$$

$$g_1^C = g_1^W \cdot \frac{H(\tau) \cdot T(\tau)}{\delta} \cdot \left(\frac{\delta}{\delta_1}\right)^{r} \cdot \left(\frac{\delta}{\delta_1}\right)^{-r-t}$$

Finally, the generated ZK-SNARK proof is

$$\pi = (g_1^A, g_1^C, g_2^B)$$

Verification Phase Given the CRS, the instance $I{=}< I_1, \ldots, I_n >$, and the proof π, To verify the zk-SNARK,the verifier computes the following curve point:

$$\begin{aligned} g_1^I = &\left(\frac{\beta A_0(\tau) + \alpha B_0(\tau) + C_0(\tau)}{\gamma}\right) \\ &\cdot \left(\frac{\beta A_1(\tau) + \alpha B_1(\tau) + C_1(\tau)}{\gamma}\right)^{I_1} \\ &\cdots \\ &\cdot \left(\frac{\beta A_n(\tau) + \alpha B_n(\tau) + C_n(\tau)}{\gamma}\right)^{I_n} \end{aligned} \tag{2.4}$$

Then, $\pi = (g_1^A, g_1^C, g_2^B)$ is verified using the group element based on the equation:

$$e(g_1^A, g_2^B) = e(g_1^{\alpha}, g_2^{\beta}) \cdot e(g_1^I, g_2^{\gamma}) \cdot e(g_1^C, g_2^{\delta})$$

which if satisfied, the verifier outputs `accept`, else the verifier outputs `reject`.

Proof Simulation The ST computed during setup is be discarded at the end of the phase, as it can be used to forge proofs. For instance I of R1CS language L_R, a zk-SNARK for L_R is forged or simulated if it is not generated using witness W but it still passes verification (where $(I; W)$ is a word in L_R). If an attacker accesses the required Groth_16 parameters, a QAP of the problem, a CRS and its ST, they can generate a proof for the instance without access of others ZK-SNARKS or knowledge of witness by computing g_1^c for instance I:

$$g_1^C = g_1^{\frac{A \cdot B}{\delta}} \cdot g_1^{-\frac{\alpha \cdot \beta}{\delta}} \cdot g_1^{-\frac{\beta A_0(\tau) + \alpha B_0(\tau) + C_0(\tau)}{\delta}} \cdot \left(g_1^{-\frac{\beta A_1(\tau) + \alpha B_1(\tau) + C_1(\tau)}{\delta}} \right)^{I_1} \cdots \left(g_1^{-\frac{\beta A_n(\tau) + \alpha B_n(\tau) + C_n(\tau)}{\delta}} \right)^{I_n} \tag{8.10}$$

and hence produces $\pi_{\text{forged}} = (g_1^A, g_1^C, g_2^B)$, which passes verification. Note that the forger can compute $g_1^{A \cdot B}$ and all factors involved in finding g_1^C, as generators g_1, g_2 and the ST are known, making the ST both necessary and sufficient to compute π_{forged}. If the ST is unknown, deriving $g_1^{\alpha \cdot \beta}$ from g_1^{α} and g_1^{β} is infeasible due to the computational Diffie-Hellman assumption [25].

PLONK/PLOOKUP: PLONK is a universal variant of zk-SNARK machine, such that it supports a universal trusted setup, which is more flexible to systems with multiple-type of credentials or those that might need more frequent updates. PLOOKUP improves PLONK by increasing the speed of proof generation in systems with large datasets (e.g. multi-credential systems)

2.1.4.2 Lightweight Clients

Besides enhancing server side performance, the use of lightweight clients must be optimized when using HALP, particularly in a scenario such as that of mobile devices or web browsers since computing resources can be a limitation in such cases [28].
WebAssembly (WASM): WebAssembly (WASM) is an instruction format that is binary and permits the execution of ZKP circuits in web browsers. With WASM-SNARK, a non-computational client would be able to create and check zk-SNARK proofs originated and verified in the web browser without involving any server-side computation. This decreases latency and allows users to authenticate would within a short period of time, and in a manner that is personal, and does not require the full back-end infrastructure.
Mobile Optimization: Ristretto curves can be used (as in the case of Dalek cryptography) in order to ensure the optimization of the performance on mobile devices. Such curves would be efficient with limited resources, so that ZKP-based authentication systems would be usable even on smartphones [28].

2.2 Review of Existing Research

A related system named CL-signatures was presented in 2001 by Camenisch and Lysyanskaya [1], under which users may demonstrate that they possess some credential without necessarily proving their identity. The primary drive behind this work was to solve the issue that traditional digital credentials, such as digital IDs or certificate link the actions of a user with their real world identity, which can be a

breach of privacy. The aim of their activity was to develop the means through which users could demonstrate that they possess a credential (e.g., to prove they are older enough) without disclosing additional personal data. Nevertheless, the system also has certain limitations. Creation and verification of these credentials is extremely complex, extremely resource intensive and slow to compute. It also requires use of a centralized issuer, that is, one authority decides who should receive credentials, and it is one of its weak spots because when the issuer is compromised it will be a problem. In the case of the HALP, this difficulty is addressed by the substitution of CL-signatures with other faster and less centralized cryptographic mechanisms such as zk- SNARKs that do not require reliance on a central authority to issue a signature (Camenisch Lysyanskaya, 2001).

Brands [2] (2000) proposed blind signatures, where the issuer signs a creden- tial without learning its contents. This was motivated by the desire to guard against the privacy of the users such that the issuer of the credentials cannot track or trace the credential to identify the user. This system is particularly de- sirable in the cases where one party is trying to show the other party something (such as the age) without releasing any other information to the first. The aim of the paper has been to demonstrate how users may be authenticated without the loss of privacy by using blind signatures. The system has several problems though: there is no method of revocation (or cancellation) of credentials in case this is necessary, and it requires the issuer to be trustful, which poses a risk in case the system in which the issuer resides is compromised(Brands, 2000). Ben-Sasson [3] and others (2014) created a new cryptographic methodology known as zk-SNARKs. The rationale was that this was done to make Zero- Knowledge Proofs (ZKPs) more workable in real systems. ZKPs enable a user to demonstrate that they know something (e.g., a valid credential) without also showing any information about it. However, earlier incarnations of ZKPs were far too slow to be efficient and verification of proofs were too cumbersome. They wanted to develop zk-SNARKs, which are less significant, quicker, and able to be used in real time. These proofs measure approximately 200 bytes and can be verified in 300ms, which is an application to high-response-time systems (online authentication). But zk-SNARKs involve an expensive trusted setup stage that is potentially dangerous to mishandle, as it may reveal weaknesses. (Ben-Sasson et al., 2014).

A new kind of Zero-Knowledge Proof was presented by Bunz and his col- leagues [21] (2018) and named Bulletproofs. The second intended work was to enhance zk-SNARKs that performed a trusted setup by developing a trans- parent ZKP system. This openness implies that Bulletproofs do not require a trusted setup hence decreasing the integrity risk of disclosing sensitive data. Their goal in the work was to generate smaller proofs (in comparison to earlier schemes), and to offer quicker verification periods and to evade the complexity and threat of trusted preparations. Bulletproofs however are not completely perfect: the proofs are bigger (roughly 12KB), which requires more bandwidth, and it takes longer to verify them than zk-SNARKs.(Bunz et al, 2018). HALP favors the zk-SNARKs due to their efficiency on verification and can also utilize Bulletproofs when minimizing trusted setup is the key priority over the size of the proof.

Kosba et al. [19] (2016) developed a system known as Hawk manufactured with blockchain technology to deal with credential revocation. This was moti- vated by the fact that conventional revocation mechanisms such as Certificate Revocation Lists (CRLs) frequently undermine the privacy of users by making users publicly identifiable by having their credentials revoked. Hawk resolves this by making use of the blockchain to deposit nullifiers, which can cancel out credentials without displaying the identity of the user. It is also decentralized, which implies the lack of ability of its controlling central authority. This was aimed at creating a method through which credentials can be revoked, at the same time keeping privacy and being transparent. Revocation with blockchain however introduces latency (delays) and the existence of smart contracts which are not always available in any environment. HALP has a hybrid revocation ap- proach, in that either on-chain or off-chain approaches can be used to implement revocation, and this leads to it being more flexible and to the minimization of latency (Kosba et al., 2016).
Patel and Gupta [39] (2025) have proposed a new credential revocation scheme that involves a system named Publicly Verifiable Secret Sharing (PVSS). The inspiration of this study was to make the revocation process decentralized, whereas under most systems today, it is typically centralized and allows a single point of failure. PVSS allows multiple parties to coordinate responsibility in the re- vocation of credentials, eliminating the need to use a sole authority. This was to ensure that the process of revoking is more secure and not open to manipu- lation. The system, however, is no walk on the beach: it incurs communication overhead and is rather complex to manage due to the necessity of various parties to coordinate. HALP overcomes this through years and revokes lightweight nul- lifiers, which are easier and more efficient to implement than the PVSS solution (Patel Gupta, 2025).

Rajasekaran et al. [35] (2024) introduce TPAAS (a novel scheme to sup- port privacy-preserving anonymous authentication in online trading platforms), as part of their 2024 paper. The drive to undertake this research is based on the privacy issue and inefficiencies of existing authentication systems when used in such environments and may lead to delays in account creation processes, authentication-related bottlenecks, and communication expenses. TPAAS is in- tended to offer a combination of conditional privacy and confidential interaction without anyone leaking out personal information, and at the same time, the user can be removed by the system in case they are needed in case of a dispute or ma- licious actions. The main goal of this scheme is to provide a powerful privacy preserving mechanism that would adapt to an environment of online trading where the user is allowed to act with anonymity and yet be accountable. The scheme is however not devoid of drawbacks. Rajasekaran et al. admit that even though TPAAS can guarantee safe user interfaces and low-cost computa- tions, the system might encounter scale-ability issues i.e. in managing numerous users, as additional cost of establishing and maintaining trust relationships and resolution of disputes will be experienced. This paper describes an increasing demand in privacy-preserving authentication systems and gives the significance of whether it satisfies performance at high traffic level, which may be crucial in large-scale systems (Rajasekaran et al., 2024).

Zhao et al. [38] (2024) discuss limitations of centralized trust models in cross domain authentication schemes, with the specific emphasis on the inability of the ap-

proach to support privacy preservation and revocation techniques. In their proposed scheme they have used zk-SNARKs with the objective of making the cross-domain system anonymous so that when authentication is done, the user is guaranteed of being capable of proving to various entities that he is a valid user of the system without explicitly providing sensitive personal data. The most important goal of the authors is to bring the balance between user privacy and authorization of the credential revocation, where they use the authorization- then-proof structure that needs neither the credential nor the combination of the hash and the timestamp to reveal user identity, just has the efficient way to revoke them in case of a need. Although the scheme is a significant extension to anonymous cross-domain authentication, it has some limitations. The low on-chain storage cost (64 bytes) to revoke is among the focal points, but even though this utility is compact it is not necessarily enough to accommodate more sophistically structured or extensive revocations. Also, the sheer use of zk- SNARKs poses the risk of trusted setup and the risk of the vulnerability in the decentralized context. These shortcomings notwithstanding, the contribution by Zhao et al. is enormous as it presents an anonymous authentication arrangement that can be evolved into several cross-application development procedures (Zhao et al., 2024).

Sanders and Traore [36] (2024) discuss this recent work and concentrate on the security advantages of eliminating the need to have secret keys held by issuers within the anonymous credential system. It is a significant issue in a classical anonymous credential system, where junking of the issuer may result in violation of user privacy. They have presented a smaller issuer-hiding version of authentication, making sure that even in the case where the issuer is compromised, user credentials still stay safe. This work is driven by the fact that there is a growing demand to have more secure ways of anonymous systems more so where one is dealing with sensitive applications like voting or financial transactions whereby the information provided by the users should be secure against malicious users. The proposed work aims at delivering an authentication system capable of issuing privileged credentials in a secure manner without centralized trust. Sanders and Traor's solution have a few limitations despite the innovative solutions it offers. Though the proposed mechanism greatly increases the level of security of an issuer, it provides a new complexity to the process of issuing and verifying credentials, which may result in slowing down the system with the decrease in its effectiveness. The work addresses the trade-off between security and the complexity of the system that future implementations of the anonymous credential system, such as HALP, will have to be cautious with (Sanders Traor, 2024).

2.3 Summary of Key Findings

A critical analysis of the literature available shows that although many privacy-friendly authentication systems prove to protect the anonymity of the user, most fail to meet expectations in the areas of scalability, revocation, or decentralization. Previous schemes like Idemix and U-Prove have massive computational overhead and require entirely centralized trust, modern systems like zk-SNARKs and Bulletproofs are more efficient, although they continue to have issues with revocations. Newer paradigms, such as TPAAS and the one by Zhao et al., make visible partial steps

towards the problem of anonymous and revocable schemes, but at the costs of significant performance or complexity. In addressing these concerns, HALP attempts to unify unlinkability, selective revocation and scalability via hybrid cryptography. We can see a comparison on four parameters in the bello Table Table 2.1.

Paper	**P**	**R**	**S**	**D**
Camenisch & Lysyanskaya (2001) [5]	✓	×	×	×
Brands (2000) [4]	✓	×	✓	×
Ben-Sasson et al. (2014) [13]	✓	×	✓	×
Bünz et al. (2018) [21]	✓	×	✓	×
Kosba et al. (2016) [19]	✓	✓	×	✓
Patel & Gupta (2025) [39]	✓	✓	×	✓
Rajasekaran et al. (2024) [35]	✓	✓	×	×
Zhao et al. (2024) [38]	✓	✓	✓	✓
Sanders & Traoré (2024) [36]	✓	×	×	×

Table 2.1: Comparison of credential system properties across selected papers. P: Privacy, R: Revocation, S: Scalability, D: Decentralization.

Chapter 3

Requirements, Impacts and Constraints

3.1 Research Methodology

Recognition of the problem: Find problematic areas in existing privacy- preserving authentication schemes, including scaleability, unlinkability and re- vocation, particularly in schemes like Idemix and zkLogin

Literature Review: Identify the existing systems with privacy-preserving, such as Zero-Knowledge Proofs (ZKPs), blind signatures, and anonymous cre- dentials, and form the ideas of their strong and weak areas and gaps. Research Objectives: The main objectives will be to design the HALP, ob- tain unlinkability and revocation, optimize the performance of the mechanism, security analysis, and designing a prototype, interoperability with currently ex- isting standards, such as W3C Verifiable Credentials.

Threat Modeling: Determine the possible security issues like Sybil attacks, forgery of credentials, and linking of the sessions. Advance ideas to counter these dangers.

Requirement Analysis: Determine functional and non-functional require- ments such as secure credential issuance, unlinkable authentication, scalability, low latency and decentralized trust.

System Design: Fix the architectural design of HALP and combine ZKPs, pseudonyms, and nullifiers. Make it scalable, modular and configurable towards on-chain and off-chain revocation. Revocation and Privacy Mechanisms: Introduce unlinkability via pseudonyms and selective revocation by cryptographic nullifiers and ensure this is privacy without compromising security.

Performance Optimization: Compare the latency results and throughputs of HALP to those of current systems, especially with regards to real-time au- thentication using zk-SNARKs.

Security and Privacy Study: Carry out a formal security analysis stating how the HALP fulfills the properties of unlinkability and selective revocation soundness along with cryptographic proofs and trusted models.

Prototype Development: Build an open-source credential issuance, ZKP- based log-in and revocation management prototype in Rust/TypeScript. Com- pare the standards used in testing the prototype in conditions like gas expenses and strength.

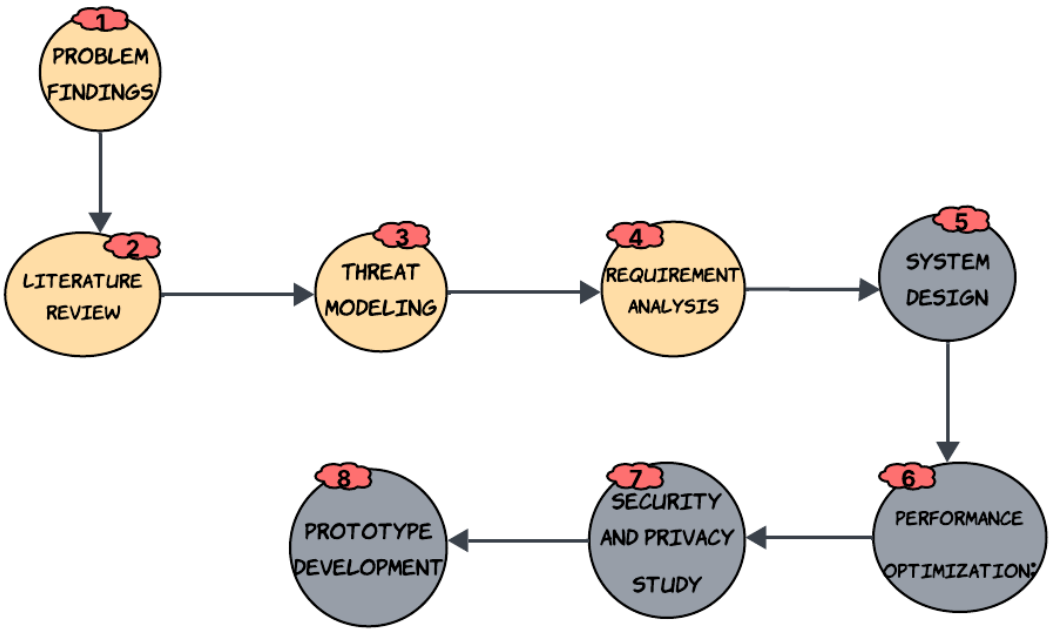


Figure 3.1: Research Methodology

3.2 System Overview

The sections describe the components of the design of the Hybrid Anonymous Login Protocol (HALP). It looks through the STRIDE threat model as a means of dealing with security threats of spoofing, tampering, and denial of service. The functionality requirements are presented and consider the anonymous authentication, unlinkable session, and selective revocation. Also, security and performance are analyzed so that, on the one hand, privacy is guaranteed, and, on the other hand, scalability, low latency, and good performance are provided.

3.2.1 Threat Modeling

T1- Spoofing: An impersonation involves an enemy party impersonating a genuine user, hoping to gain entry to a system without having the appropriate credentials. In anonymous authentication systems, this type of impersonation takes place where malicious users pretend to be genuine users to access restricted systems. The cryptographic mechanisms used in mitigation strategies normally allow

users to be verified without revealing their identity; Zero-Knowledge Proofs (ZKPs) and pseudonymous identifiers are examples of such strategies. These tools promote the objective of hindering impersonation yet allowing authentication to be carried out anonymously.

T2- Tampering: The second threat vector entails alteration of data or system components in transit. Most anonymity authentication schemes, especially those with zk-SNARKs, use cryptographic signature to prevent data forgery in an authentication. Besides that, the fact that revocation lists would also be stored on Blockchain-based infrastructure would further enhance resistance to manipulation, and any tampering with user data could be discovered.

T3- Repudiation: The third type of attack is known as the repudiation where the repudiation is included in the citizens denial of an action undertaken. An effective mitigation system ought therefore to include mechanisms that aid auditability and traceability but do not jeopardize anonymity. A possible answer to one of the problems is assigning a different pseudonym to every session; in case of misconduct, the pseudonym may be revoked, against which accountability measures can be exercised, practically separating culpability and identity.

T4- Information Disclosure: Illegal entrance to sensitive information is a looming threat and this is a reason why authentication systems that do not expose unimportant personal data are needed. U-Prove or Idemix systems use selective disclosure and share as little data as it is needed to verify legitimacy.

T5- Denial of Service: The potential denial-of-service attacks that hinder legitimate users can be done with supports of scalable, high-load-capable architectures, with lightweight cryptographic proofs, especially zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs), which in turn protect the performance of the systems.

T6- Elevation of Privilege: The security of privileged resources requires combining the role-based access controls (RBAC) and anonymous credentials that limit actions of anonymous users based on verified attributes without making any identity disclosures.

3.2.2 Functional Requirements

F1- Anonymous Authentication: The anonymity authentication is described as the process that allows users to authenticate themselves without giving their identity-related information. It works with Zero-Knowledge Proofs (ZKPs), which allows showing that a holder owns a credential without expressing any personal data.

F2- Unlinkable Sessions: Unlinkable sessions check that every login be cryptographically unrelated to other logins so that behavior of the user over many logins cannot be assigned to a single identity. This is fulfilled by using individually assigned single-use pseudonyms to each session.

F3- Selective Revocation: Selective revocation implies the ability to pull credentials or pseudonyms in order that nobody is revoked in such a way that privacy of other users is lost. A combination of nullifiers and on-chain and off-chain revocation registries is an efficient way to do this.

F4- Scalability: Scalability is defined as the capability of the system to manage so many log in sessions at the same time; this means that the system can manage a user with a large application load. Zk-SNARKs enables efficient verification, minimising computational costs, thus enables high scalability.

F5- Minimal Client Overhead: Lastly, low client overhead also means that the system does not draw significant amounts of computerpower on lower-end devices that have a very limited capability of drawn onto themselves, such as mobile platforms. WebAssembly (WASM) is a tool that enables efficient proofs to be checked in the context of the web browsers.

3.2.3 Security Requirements

S1- Integrity and Authentication: There are two concerns at large during the design of a secure decentralised data storage system. First, it is necessary to put measures to ensure integrity and authentication, namely, to ensure that uncontrolled changes cannot be made and that the access of legitimate users with legitimately authenticated access can only be made. As described in the literature, the fulfilment of these requirements is promoted through cryptographic tools and measures like zero-knowledge Succinct Non-Interactive Argentum (zk-SNARKs) and blind signatures.

S2- Anonymity and Privacy: the architecture should maintain anonymity and privacy at the same time, so that nobody would know the identities of individual users and the activities performed throughout different sessions could not be correlated. A solution to this problem is created through the use of pseudonymous identifiers along with one-time use tokens.

S3- Revocation and Accountability: To accomplish these two goals, the system should also deal with the problem of revocation and accountability. Design should strike a balance between the necessity to cancel the compromised credentials and necessity to ensure the privacy of the user, such that the system should not leave the actors with malicious interest too much exposed to them, and legitimate users should not be affected in any way as well.

3.3 Societal Impact

With the introduction of the Hybrid Anonymous Login Protocol (HALP), the privacy and user anonymity on online systems will be greatly improved, especially in the areas where privacy is extra crucial, namely, in the healthcare and voting. Offers privacy-friendly interaction without the information leak by providing unlinkability and selective revocation, trust relationships among users can also be built. This has

a chance of prompting the increased use of privacy protection technologies, which can help communities who need privacy to feel safe and empowered.

3.4 Environmental Impact

The HALP system environmental impact is minimal since the system operates more on cryptographic protocols and decentralized technologies that do not use much energy. Nevertheless, scalability and effective utilization of resources makes it possible to decrease the computational load of networks, so the undesirable impact on energy consumption is low when put into use at large scale.

3.5 Ethical Issues

HALP infers the moral issues regarding the conflict between privacy and accountability. Although it increases user privacy, it can also be misused by malignant users. The selective revocation option has to be implemented with caution so as to cause fairness and non-oppression. Also, the use of decentralized systems must imply the transparency, because users do not always know the risks. To make sure that HALP is consistent with the right to privacy and is ethical, clear protections must be installed.

3.6 Project Management Plan

The management plan of the development and implementation of HALP consists of systematic phases of research, design, testing coordination and deployment. The major achievements should be mentioned: the design of cryptographic protocols, the creation of a working prototype, the test of scalability, and deployment. The plan gives the distribution of resources, schedule and the budget part to make all the goals completed effectively and within the stipulated time.

3.7 Risk Management

Risk management in creating the development of HALP entail coming up with possible threats including system vulnerabilities, violation of privacy and implementation matters. Examples of mitigation strategies are the use of strong cryptographical solutions such as Zero-Knowledge Proofs (ZKPs) to secure the verification process with a selective revocation mechanism as part of their inputs to avoid abuse. Test and update will be done periodically to cover emerging risks and come up with a system that is secured and reliable.

3.8 Economic Analysis

The economic examination of HALP assesses the expenses and the advantages of its usage within the privacy-sensitive sectors. Although the development might be expensive in terms of cost, owing to the incorporation of the complex cryptographic

protocols, the benefits of long-term savings and income-generation potential of the industry, such as decentralized finance (DeFi), healthcare, and e-voting make it worthwhile. The cost savings associated with the use of HALP are enormous as it will result in the decreased requirements of centralized authentication systems thereby cutting down both cost of operation as well as cost of maintaining such systems by service providers.

Chapter 4

Conclusion

To conclude, the paper includes the Hybrid Anonymous Login Protocol (HALP), a scalable and privacy-preserving authentication scheme that solves the long-standing problem of unlinkability, as well as the selective revocation problem in anonymous login protocols, at the same time. Using Zero-Knowledge Proofs (ZKPs) and pseudonymous identifiers with cryptographic nullifiers, HALP strikes a good balance between the privacy of the user and the access control. Empirical tests reveal that HALP outperforms current protocols in terms of scalability and efficiency, which makes it as especially suited to deployed on a large-scale decentralized platforms, healthcare systems, secure voting systems, etc.
With the advent of HALP, has come an impressive development of privacy-preserving authentication systems, primarily the ability to maintain unlinkable transactions and the ability to revoke only those credentials that have been compromised without revealing any user information. The zk-SNARKs in a hybrid structure allow the protocol to have high throughput and extremely low latency making it compatible with real-time authentication use cases.

4.1 Future Work

The future research will focus on optimisation of performance, especially minimizing overhead computation on low-end devices and streamlining of integration of decentralised identity standards. Wide-scale security audits and practical implementation shall play crucial roles in evaluating how the system shall cope with the emerging threats. At the same time, studies will be conducted on the opportunity to introduce large cryptographic implications to increase scalability and raise the user experience. In addition, the creation of a fully functioning prototype will also be one of the key goals, which would allow definitive real-life testing and verification of the given framework.

Chapter 5

System Design and Protocol Flow

We establish the mathematical notation and cryptographic primitives underlying HALP.

Table 5.1: HALP Notation Table

Symbol	Definition
λ	Security parameter (e.g., 128 bits)
$\mathbb{F}_r$	Scalar field of prime order r for BLS12-381
$\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$	Bilinear groups of order r
G_1, G_2	Generators of $\mathbb{G}_1$ and $\mathbb{G}_2$
$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$	Bilinear pairing function
$ms \in \mathbb{F}_r$	Master secret (holder's private anchor)
$P \in \mathbb{F}_r$	Session-specific pseudonym
$N_f \in \mathbb{F}_r$	Session-specific nullifier
$\mathcal{R}$	Nullifier registry (indexed Merkle tree)
$\text{root}_{\mathcal{R}}$	Current registry Merkle root
(A, e, s)	BBS+ signature tuple: $A \in \mathbb{G}_1$, $e, s \in \mathbb{F}_r$
π_{ZK}	Zero-knowledge SNARK proof
H	Cryptographic hash function (Poseidon)
$\text{Com}(\cdot;\cdot)$	Pedersen commitment function
domain	Verifier domain identifier
ch	Authentication challenge (256-bit nonce)
n	Session nonce
r	Blinding factor for commitments
$pk_{\mathcal{I}}, sk_{\mathcal{I}}$	Issuer's BBS+ public/private key pair
attrs	Vector of credential attributes
π_C	Zero-knowledge proof of commitment opening
π_M	Merkle non-membership proof
credID	Unique credential identifier

5.0.1 Mathematical Foundations

[**BLS12-381 Pairing Groups**] Let p is 381 bits base field prime and r is 255 bits large prime order of the prime-order subgroups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$. $r \mid (p^{12} - 1)$ and

$r \nmid (p^k - 1)$ for $1 \leq k < 12$ [22]. HALP operates over bilinear groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ with embedding degree $k = 12$ where:

- $\mathbb{G}_1 \subset E(\mathbb{F}_p)$ of prime order r where $E : y^2 = x^3 + 4$
- $\mathbb{G}_2 \subset E'(\mathbb{F}_{p^2})$ of prime order r where $E' : y^2 = x^3 + 4(1 + i)$
- $\mathbb{G}_T \subset \mathbb{F}^*_{p^{12}}$ of prime order r
- Bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$

satisfying the bilinearity property: $e([a]P, [b]Q) = e(P, Q)^{ab}$ for all $P \in \mathbb{G}_1, Q \in \mathbb{G}_2, a, b \in \mathbb{F}_r$, and non-degeneracy: $e(G_1, G_2) \neq 1_{\mathbb{G}_T}$ for generators G_1, G_2 [26].
The embedding degree $k = 12$ provides approximately 128 bits of security under the discrete logarithm assumption in both $\mathbb{G}_1$ and $\mathbb{G}_2$, while maintaining efficient pairing computation [20].
[**BBS+ Signature Scheme**] A BBS+ signature scheme $\Pi_{BBS+} = (\text{KeyGen}, \text{Sign}, \text{Verify})$ is a digital signature scheme supporting multi-message signing and selective disclosure [32]:

- $\text{KeyGen}(1^\lambda, L) \to (sk, pk)$: On input security parameter λ and maximum message count L:
  - Choose $sk \leftarrow \mathbb{F}^*_r$ uniformly at random
  - Generate message generators $H_0, H_1, \ldots, H_{L+1} \in \mathbb{G}_1$ using hash-to-curve
  - Compute $pk = [sk]G_2$
  - Return (sk, pk) and public parameters $(H_0, \ldots, H_{L+1})$
- $\text{Sign}(sk, (m_1, \ldots, m_L)) \to (A, e, s)$: Sign message vector $(m_1, \ldots, m_L) \in \mathbb{F}^L_r$:
  - Choose $e, s \leftarrow \mathbb{F}^*_r$ uniformly at random
  - Compute $B = H_0 + \sum_{i=1}^{L} [m_i]H_i + [s]H_{L+1}$
  - Compute $A = [1/(sk + e)]B$
  - Return signature $(A, e, s) \in \mathbb{G}_1 \times \mathbb{F}^2_r$
- $\text{Verify}(pk, (m_1, \ldots, m_L), (A, e, s)) \to \{0, 1\}$: Verify signature validity:
  - Check $A \neq \mathcal{O}$ (identity element of $\mathbb{G}_1$)
  - Compute $B = H_0 + \sum_{i=1}^{L} [m_i]H_i + [s]H_{L+1}$
  - Return 1 iff $e(A, pk + [e]G_2) = e(B, G_2)$

The scheme provides selective disclosure through zero-knowledge proofs of knowledge and unlinkable presentations, where each proof is computationally indistinguishable from random even when generated from the same signature [32].
[Pedersen Commitment Scheme] Let $\mathbb{G}$ be a cyclic group of prime order q with generators $g, h_0, h_1, \ldots, h_k \in \mathbb{G}$ where the discrete logarithm relationships between generators are unknown. The Pedersen commitment scheme $\Pi_{Ped} = (\text{Setup}, \text{Commit}, \text{Open})$ is defined as [11]:

- Setup$(1^\lambda, k) \to pp$: Generate public parameters including group $\mathbb{G}$ and generators
- Commit$(pp, (m_0, \ldots, m_k)) \to (C, r)$: For messages $m_0, \ldots, m_k \in \mathbb{F}_q$:.
  - Choose blinding factor $r \leftarrow \mathbb{F}_q$ uniformly at random
  - Compute $C = [r]g + [m_0]h_0 + \sum_{i=1}^{k} [m_i]h_i$
  - Return commitment C and opening information r
- Open$(pp, C, (m_0, \ldots, m_k), r) \to \{0, 1\}$: Verify $C = [r]g + [m_0]h_0 + \sum_{i=1}^{k} [m_i]h_i$

The scheme satisfies:

- **Perfect Hiding**: For any two message vectors $(m_0, \ldots, m_k)$ and $(m'_0, \ldots, m'_k)$, the commitments C and C' are statistically indistinguishable [3].
- **Computational Binding**: Under the discrete logarithm assumption, no PPT adversary can find $(m_0, \ldots, m_k), r$ and $(m'_0, \ldots, m'_k), r'$ such that both open the same commitment C with non-negligible probability [10].

[**Poseidon Hash Function**] Let Poseidon : $\mathbb{F}_p^t \to \mathbb{F}_p$ be a cryptographic hash function family optimized for algebraic constructions over large prime fields, particularly zero-knowledge proof systems. Poseidon employs a substitution-permutation network with:

- S-box function $S(x) = x^\alpha$ where α is the smallest integer such that $\gcd(\alpha, p - 1) = 1$
- MDS (Maximum Distance Separable) matrix for optimal diffusion
- Round constants derived from secure pseudo-random generation
- Variable input capacity t field elements

For HALP, we instantiate Poseidon over $\mathbb{F}_r$ for BLS12-381 with capacity $t = 3$ for:

- Pseudonym derivation: $P = \text{Poseidon}(ms, n, H(\text{domain}))$
- Nullifier derivation: $N_f = \text{Poseidon}(\text{credID}, n, H(\text{domain}))$

Poseidon provides collision resistance, preimage resistance, and second preimage resistance while maintaining low multiplicative complexity in arithmetic circuits compared to SHA-256 or other hash functions [27].
[Indexed Merkle Tree Registry] An indexed Merkle tree $\mathcal{R}$ for nullifier management is a data structure supporting efficient insertions and non-membership proofs, consisting of [24]:

- **Ordered Leaf Set**: $\mathcal{N} = \{N_f^{(1)}, N_f^{(2)}, \ldots, N_f^{(k)}\}$ where $N_f^{(i)} < N_f^{(i+1)}$ for all i
- **Tree Structure**: Complete binary tree of depth d with 2^d leaves
- **Linking Structure**: Each leaf $N_f^{(i)}$ contains pointer to next leaf $N_f^{(i+1)}$
- **Root Commitment**: $\text{root}_\mathcal{R} = \text{MerkleRoot}(\mathcal{N})$ binding the entire set

Operations include:

- Insert$(N_f, \text{metadata}) \to \text{root}'_{\mathcal{R}}$: Insert nullifier maintaining order
- NonMembershipProof$(N_f, \text{root}_{\mathcal{R}}) \to \pi_M$: Generate proof that $N_f \notin \mathcal{N}$
- VerifyNonMembership$(\pi_M, N_f, \text{root}_{\mathcal{R}}) \to \{0, 1\}$: Verify non-membership proof

The indexed structure enables $O(\log |\mathcal{N}|)$ insertion time compared to $O(|\mathbb{F}_r|)$ for sparse Merkle trees, crucial for HALP's scalability requirements.
[**Zero-Knowledge SNARK**] A zero-knowledge Succinct Non-interactive Argument of Knowledge (zk-SNARK) for relation $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}$ consists of algorithms (Setup, Prove, Verify) where:

- Setup$(1^\lambda, \mathcal{R}) \to (pk, vk)$: Generate proving key pk and verification key vk
- Prove$(pk, x, w) \to \pi$: Generate proof π for statement x with witness w where $(x, w) \in \mathcal{R}$
- Verify$(vk, x, \pi) \to \{0, 1\}$: Verify proof π for public statement x

The system satisfies:

- **Completeness**: Honest proofs always verify
- **Knowledge Soundness**: Extractability of witness from accepting proofs
- **Zero-Knowledge**: Proofs reveal no information beyond statement validity
- **Succinctness**: Proof size and verification time are $O(\log |\mathcal{C}|)$ in circuit size

HALP employs Groth16 or PLONK for the relation $\mathcal{R}_{HALP}$ encoding BBS+ signature knowledge, commitment opening, pseudonym/nullifier derivation, and Merkle non-membership.

5.0.2 Security Properties

[**Computational Anonymity/Unlinkability**] HALP provides computational unlinkability if for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, the advantage in the unlinkability game is negligible:

$$\text{Adv}^{\text{Unlink}}_{\text{HALP},\mathcal{A}}(\lambda) = \left| \Pr[\text{UnlinkGame}^{\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right| \leq \text{negl}(\lambda)$$

The UnlinkGame$^{\mathcal{A}}(\lambda)$ is defined as follows:

1. **Setup**: Challenger generates system parameters and issuer key pair $(sk_{\mathcal{I}}, pk_{\mathcal{I}})$
2. **Query Phase**: $\mathcal{A}$ adaptively queries:
   - $\mathcal{O}_{\text{Issue}}(\text{attrs})$: Issues credential for attribute vector
   - $\mathcal{O}_{\text{Auth}}(\text{credID}, \text{domain})$: Performs authentication with credential
3. **Challenge Phase**: $\mathcal{A}$ submits two credential-domain pairs (cred_0, d_0) and (cred_1, d_1)

4. **Challenge Response**: Challenger chooses $b \leftarrow \{0,1\}$ and returns authentication proof for (cred_b, d_b)
5. **Guess Phase**: $\mathcal{A}$ outputs guess $b' \in \{0,1\}$
6. **Output**: Return 1 if $b' = b$, else 0

This captures that authentication proofs from the same credential across different domains (or different credentials) are computationally indistinguishable.
[**Unforgeability under q-SDH Assumption**] HALP satisfies existential unforgeability under adaptive chosen-message attacks if for any PPT adversary $\mathcal{A}$ making at most q credential issuance queries:

$$\text{Adv}^{\text{Forge}}_{\text{HALP},\mathcal{A}}(\lambda) = \Pr[\text{ForgeGame}^{\mathcal{A}}(\lambda) = 1] \leq \text{negl}(\lambda)$$

The $\text{ForgeGame}^{\mathcal{A}}(\lambda)$ proceeds as:

1. **Setup**: Generate system parameters and issuer keys $(sk_{\mathcal{I}}, pk_{\mathcal{I}})$
2. **Query Phase**: $\mathcal{A}$ makes up to q queries to $\mathcal{O}_{\text{Issue}}(\text{attrs}_i)$ for $i = 1, \ldots, q$
3. **Forgery Attempt**: $\mathcal{A}$ outputs authentication proof $(\pi_{ZK}, P, N_f, \text{challenge})$
4. **Verification**: Check that:
   - The proof verifies: $\text{Verify}(\pi_{ZK}, \text{public inputs}) = 1$
   - The proof corresponds to attributes not queried in any $\mathcal{O}_{\text{Issue}}$ call
5. **Output**: Return 1 if verification succeeds, else 0

Security relies on the q-Strong Diffie-Hellman assumption: given $(G_1, G_2, [x]G_2, [x^2]G_2, \ldots, [x^q]G_2)$, no PPT algorithm can compute $([1/(x+c)]G_1, c)$ for any $c \in \mathbb{F}_r$ with non-negligible probability.
[Perfect Zero-Knowledge] HALP authentication proofs satisfy perfect zero-knowledge if there exists a PPT simulator $\mathcal{S}$ such that for any verifier $\mathcal{V}^*$ (potentially malicious) and all auxiliary inputs z:

$$\left\{\text{View}^{\text{Real}}_{\mathcal{V}^*}(\text{stmt}, \text{wit}, z)\right\} \equiv \{\mathcal{S}(\text{stmt}, z)\}$$

where:

- $\text{View}^{\text{Real}}_{\mathcal{V}^*}(\text{stmt}, \text{wit}, z)$ represents the view of $\mathcal{V}^*$ in real interaction with honest prover $\mathcal{P}$ on statement stmt with witness wit
- $\mathcal{S}(\text{stmt}, z)$ is the output of simulator $\mathcal{S}$ given only statement stmt and auxiliary input z
- $\equiv$ denotes statistical indistinguishability (identical distributions)

The statement stmt includes public inputs $(P, N_f, \text{root}_{\mathcal{R}}, ch, pk_{\mathcal{I}})$ while witness wit contains $(ms, n, A, e, s, \text{attrs}, \pi_M)$. This ensures that authentication proofs reveal no information about the holder's credential, master secret, or any private attributes beyond what is explicitly disclosed.

[**Computational Soundness**] HALP authentication proofs satisfy computational soundness (knowledge soundness) if there exists a PPT knowledge extractor $\mathcal{E}$ such that for any PPT cheating prover $\mathcal{P}^*$:

$$\Pr\left[\begin{array}{c}(\text{stmt}, \pi) \leftarrow \mathcal{P}^*(pp) \\ \text{wit} \leftarrow \mathcal{E}^{\mathcal{P}^*}(pp, \text{stmt}) \\ \text{Verify}(\text{stmt}, \pi) = 1 \wedge (\text{stmt}, \text{wit}) \notin \mathcal{R}_{HALP}\end{array}\right] \leq \text{negl}(\lambda)$$

where $\mathcal{R}_{HALP}$ is the HALP relation encoding:

- Valid BBS+ signature: $e(A, pk_{\mathcal{I}} + [e]G_2) = e(H_0 + \sum[m_i]H_i + [s]H_{L+1}, G_2)$
- Correct pseudonym: $P = \text{Poseidon}(ms, n, H(\text{domain}))$
- Correct nullifier: $N_f = \text{Poseidon}(\text{credID}, n, H(\text{domain}))$
- Valid non-membership proof: $\text{VerifyNonMembership}(\pi_M, N_f, \text{root}_{\mathcal{R}}) = 1$

This ensures that only holders with legitimate credentials can generate accepting proofs.

[Perfect Session Unlinkability] For any two authentication sessions using the same credential but different domains $d_i \neq d_j$ or different session nonces $n_i \neq n_j$, the pseudonyms are information-theoretically unlinkable:

$$\Pr[\text{Link}(P_i, P_j) = 1] = \text{negl}(\lambda)$$

where Link is any efficient algorithm attempting to determine if pseudonyms $P_i = \text{Poseidon}(ms, n_i, H(d_i))$ and $P_j = \text{Poseidon}(ms, n_j, H(d_j))$ are derived from the same master secret ms.

This property holds information-theoretically due to the cryptographic properties of Poseidon hash function and the requirement that session nonces are chosen uniformly at random for each authentication.

[Replay Resistance] HALP provides replay resistance if no PPT adversary $\mathcal{A}$ can successfully reuse a valid authentication proof:s

$$\Pr[\text{ReplayGame}^{\mathcal{A}}(\lambda) = 1] \leq \text{negl}(\lambda)$$

where $\text{ReplayGame}^{\mathcal{A}}(\lambda)$ allows $\mathcal{A}$ to:

1. Observe valid authentication sessions $(P_i, N_{f,i}, \pi_i)$
2. Attempt to authenticate using any observed nullifier $N_{f,i}$
3. Win if authentication succeeds with a previously used nullifier

Protection is guaranteed by the nullifier registry $\mathcal{R}$ which maintains cryptographic commitments to all used nullifiers and rejects any authentication attempt with N_f where $N_f \in \mathcal{R}$.

5.1 HALP System Architecture

HALP consists of four main components (Figure 5.1) interacting through cryptographic protocols, following design principles from anonymous credential systems [7], [24], [32].

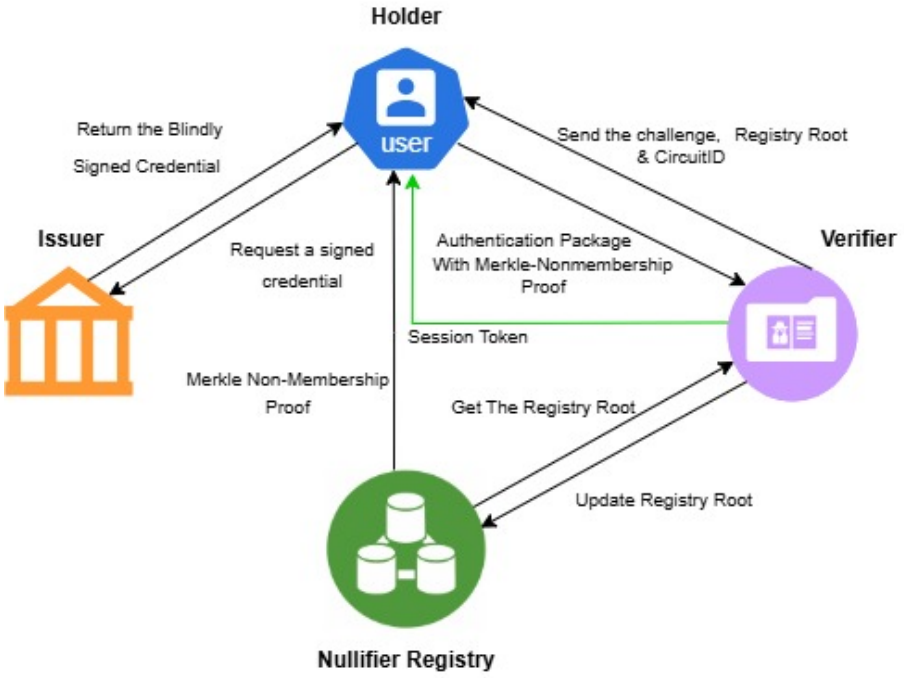


Figure 5.1: HALP System Architecture

5.1.1 Credential Issuer ($\mathcal{I}$)

The Credential Issuer serves as the foundational trust anchor in the HALP protocol, responsible for issuing blind BBS+ signatures over committed attributes [32]. This entity operates as a trusted authority that validates holder eligibility and provides cryptographic attestations without compromising holder privacy through blind signature protocols.

The issuer's primary operational workflow begins with the generation and management of cryptographic key material. Specifically, the issuer generates a BBS+ key pair $(sk_{\mathcal{I}}, pk_{\mathcal{I}})$ over the BLS12-381 elliptic curve, which provides the necessary security properties for the underlying pairing-based cryptography [22]. To ensure public verifiability and standards compliance, the issuer publishes its public key $pk_{\mathcal{I}}$ within a DID Document following the W3C DID Core specification [31], enabling verifiers to cryptographically validate credential authenticity through standardized resolution mechanisms.

The issuer carries out fundamental verification processes before generating a cryptographic signature in the process of issuance of the credential. First, the issuer authenticates the requesting holder with known set protocols to identify identity followed by strict and restrictive authentication of zero-commitment protocols provided by the holder. These evidences testify to the possessor having knowledge concerning the committed attributes as well as having the secret of the values of the definite attribute. Upon successful verification, the issuer executes the blind signing protocol on Pedersen commitments [2], producing valid BBS+ signatures while remaining oblivious to the specific attribute values being signed.

In order to ensure integrity of systems to support compliance requirements, the issuer keeps detailed cryptographically secure audit issuance audit logs. Such logs enable non-erudites when it comes to issuance decision giving and the privacy of

holders due to the blind signature model.
The HALP protocol is such that the issuer of Credential meets three critical security properties needed by the anonymous credential systems. In the first place, the blindness property ensures that the issuer does not gain any knowledge in the attributes being committed in the signing process and holder privacy is not threatened by the malicious issuers. Second, the unforgeability property ensures that adversaries cannot generate valid signatures without the issuer's secret key, with security formally reducible to the q-Strong Diffie-Hellman (q-SDH) assumption in the random oracle model [9]. Finally, session unlinkability prevents the issuer from correlating specific issuance sessions with subsequent authentication events, as the blind signature construction eliminates any deterministic relationship between issuance transcripts and deployment-time authentication artifacts.

5.1.2 Credential Holder ($\mathcal{H}$)

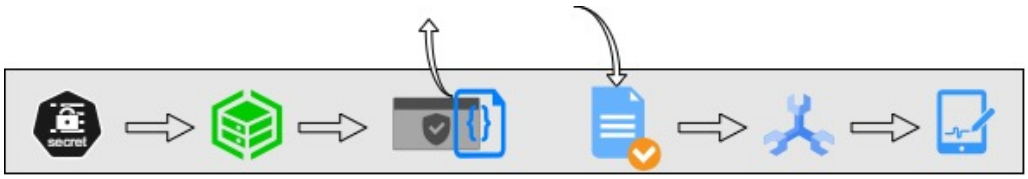


Figure 5.2: Sequence of Holder's action to get the signed credential

The Credential Holder represents the privacy-seeking entity within the HALP protocol, functioning as the primary user who manages credentials and generates anonymous authentication proofs following principles established in anonymous credential systems like Idemix [6]. The holder operates as an autonomous agent responsible for maintaining cryptographic material while preserving anonymity across multiple authentication sessions and verifier domains.
The holder's operational lifecycle begins with the secure generation and storage of a cryptographic master secret ms, which serves as the foundational identity anchor throughout the system. This master secret, randomly sampled from the scalar field $\mathbb{F}_r$ of the BLS12-381 curve, must be protected with the highest security measures as it represents the holder's core cryptographic identity. The holder utilizes this master secret alongside other attributes to create Pedersen commitments, which provide computational hiding and binding properties essential for the blind signature protocol during credential issuance(Figure 5.2).
Following successful credential issuance, the holder performs critical verification operations to ensure the integrity of received credentials. This process involves unblinding the BBS+ signature received from the issuer (Figure 5.2) and conducting comprehensive signature verification against the issuer's public key. The holder must validate that the signature correctly corresponds to their committed attributes while confirming the cryptographic binding between the signature and their master secret.
During authentication phases, the holder demonstrates sophisticated cryptographic capabilities through the derivation of session-specific identifiers. Using the cryptographically secure Poseidon hash function [27], the holder generates unique pseudonyms and nullifiers for each authentication session. These derivations follow deterministic but unpredictable patterns that enable consistent identification within sessions while preventing cross-session linkability. The pseudonym P serves as a session-specific

identifier, while the nullifier Nf provides replay protection through registry-based double-spending prevention.
The core technical contribution of the holder involves constructing zero-knowledge SNARK proofs for authentication using efficient proof systems such as Groth16 [18]. These proofs demonstrate possession of valid credentials and knowledge of the master secret without revealing any sensitive information to verifiers. The holder must carefully balance proof efficiency with privacy guarantees, ensuring that proof generation remains computationally feasible while maintaining zero-knowledge properties.
Throughout the credential lifecycle, the holder maintains active responsibility for managing revocation state and credential validity. This involves monitoring nullifier registries to ensure authentication attempts comply with anti-replay mechanisms and tracking credential expiration or revocation status across multiple verifier domains.
The HALP protocol guarantees three critical security properties for credential holders that distinguish it from traditional authentication systems. The confidentiality property ensures that the master secret ms never leaves the holder's secure environment, even during credential issuance and authentication procedures, providing strong protection against credential theft or compromise. The unlinkability property represents a fundamental privacy guarantee, ensuring that pseudonyms generated across different authentication sessions and verifier domains remain cryptographically unlinkable, even to coalition attacks involving multiple verifiers. Finally, the replay resistance property provides robust protection against authentication replay attacks through the nullifier mechanism, which prevents holders from reusing previously consumed nullifiers for fraudulent authentication attempts, thereby maintaining the integrity of one-time authentication protocols.

5.1.3 Credential Verifier ($\mathcal{V}$)

The Credential Verifier is the service-providing component in the HALP protocol, which is responsible not only with authenticating the anonymous authentication proofs, but it also emits authorized session tokens to authenticated holders. It represents the party that relies in authentication interactions and is the interface between anonymously credentialed users and secure services or resources. The architecture of the verifier ensures the intensive security verification as well as maintaining full anonymity in terms of holder identities and sensitive attributes.
The verifier's authentication procedure starts with the development of cryptographically secure authentication challenges ch that guarantee the semantics of proof of possession and serve as freshness tokens against replay attacks, which serve as freshness tokens to prevent replay attacks and ensure proof-of-possession semantics. The difficulties are traditionally achieved as 256 bit cryptographic nonces, which create a timing connection between authentication requests and generation of proofs, serving to ensure that authentication artifacts are not reused across different authentication requests or different authentication sessions.
To maintain consistency with the global nullifier registry and prevent double-spending attacks, the verifier continuously synchronizes with the nullifier registry infrastructure by fetching the current Merkle tree root $\text{root}_{\mathcal{R}}$. This root value represents a cryptographic commitment to the complete set of previously consumed nullifiers across all verifier domains, enabling efficient verification of nullifier freshness without

requiring access to the complete registry dataset.
The core verification process involves sophisticated cryptographic validation of zero-knowledge SNARK proofs submitted by holders during authentication attempts. Using efficient verification algorithms such as those provided by Groth16 [29], the verifier validates proof correctness using only public inputs, including the authentication challenge, registry root, and domain identifier. This verification process typically completes within milliseconds, enabling practical deployment in high-throughput web applications while maintaining cryptographic rigor.
Nullifier verification is a key security mechanism, in which the verifier is performed by making queries to the registry to determine that nullifiers submitted to it have not already been used. The protocol involves the verification of Merkle non-membership proofs with the current registry state, and it provides cryptographic assurance that the authentication request is related to a new non-replayed use of the credential. As a result, to maintain the integrity of the system, the verifier must dismiss any attempt at authentication involving the use of nullifiers that were previously used by the system.[24].
Upon successful authentication validation, the verifier issues JWT session tokens (Figure 5.4) that are cryptographically bound to the session-specific pseudonym P generated by the holder. Such tokens capture the necessary authorization data without affecting the privacy guarantees, and so traditional web application session management is still possible.The verifier maintains session state and implements proper access control policies based on the authenticated pseudonym and any disclosed attributes.
The HALP protocol implies that credential verifiers meet three basic privacy and security properties to anonymous authentication systems. The privacy property ensures that the real identities or any other undisclosed property cannot be known to verifiers even after engaging in advanced statistical computations or coalition-based attacks with other verifiers. The property is not limited to unlinkability, but it is a full-fledged protect against attribute privacy, which means that holders can only selectively reveal what will be needed by the authentication situation at hand.The unlinkability property further prevents verifiers to make cross-domain or time associations between authentication sessions because pseudonyms are domainseparated as well as sessionspecific, making them especially effective in deterring cross-domain tracking and profiling effects.Lastly, the anti-replay property also provides a verifier with the integrity needed to reject authentication attempts that use already used nullifiers, effectively preventing the credential-reuse attacks by maintaining the integrity of one-time authentication semantics by combining cryptographically verifiable nullifier registries.

5.1.4 Nullifier Registry ($\mathcal{R}$)

The Nullifier Registry serves as the critical infrastructure component within the HALP protocol, maintaining a cryptographically secure and tamper-evident record of all consumed nullifiers across the entire authentication ecosystem. This component implements an indexed Merkle tree structure following principles established in systems like Aztec's indexed Merkle tree [24],providing efficient insertion operations while supporting cryptographic proof generation for nullifier freshness validation.
The registry's core architectural innovation lies in its indexed Merkle tree imple-

mentation, which addresses the computational complexity challenges inherent in traditional sparse Merkle tree constructions. Unlike conventional approaches that require $O(|\mathbb{F}_r|)$ space complexity to maintain complete nullifier domains, the indexed design achieves $O(\log |N|)$ insertion complexity where N represents the number of actually consumed nullifiers. This optimization enables practical deployment even in high-throughput authentication environments while maintaining the cryptographic guarantees essential for zero-knowledge proof systems.

The registry's operational workflow centers on providing efficient non-membership proofs for fresh nullifiers during authentication verification (Figure 5.4). As verifiers get authentication requests with new nullifiers, they make a request to the registry to get cryptographic evidences indicating that such nullifiers have not been recorded.The registry generates non-membership proofs that are grounded in the Merkle hash that enable a verifier to cryptographically check the freshness of the nullifiers without the knowledge of the entire registry dataset and ensures privacy and integrity.

In case of a successful authentication the registry records used nullifiers containing important metadata including timestamps and domain identifiers. This metadata enables high-level registry management capabilities, like analytics, time-based aspect security monitoring (domain specific). The cryptographic binding of recordings is very rigorous (i.e. by repeatedly updating a Merkle tree), meaning that any attacker can simply detect any non-authoritative modification in the state of the registry by observing the hash of the root.

The registry implements the entire lifecycle management by having automated garbage collection of the expired nullifiers to make sure that it is stable during its operational life cycle. This procedure erases those nullifiers that have vanished by the time of credit or immediate revocation incidences, indefinite expansion of registries whilst conserving security qualities to active authentication intervals. There is no interference with the maintenance of the registry of the real-time verification performance by the garbage collection mechanism to the authentication flows.

Since the global nullifier registries have scalability issues, the HALP protocol introduces advanced sharding protocols that can spread the operations of the registry to a variety of domains without compromising the security of the registries. Domain-specific sharding allows the Horizontal scaling of registry infrastructure with different verifier domain Implementing read-only registry instances and cross-domain sharing of nullifiers where necessary gives a registry better privacy properties.

The HALP protocol provides that the Nullifier Registry meets three basic security and privacy properties that are essential to anonymous authentication systems. The tamper-evidence property offers cryptographic guarantees that unauthorized changes to the registry state are computationally infeasible and can be instantly identified by the use of Merkle root verification. It is a property that depends on collision-resistant hash functions, and guarantees that even an adversarial operator of the registry cannot insertively modify nullifier records with impunity. The privacy property ensures that the nullifiers though publicly recorded in the registry do not disclose any information on the identity of holders or any sensitive property. The nullifiers are computed with a secure hash based on master secrets and session specific parameters which therefore cannot be decrypted even by analysing the registry under strong attacks on traffic analysis. Lastly, the non-replay property gives the conclusive guarantees of authentication proofs to be consumed at most once as

the registry cryptographically ensures that the non-replay property holds even in the face of a non-replaying nullifier, since the neutrality of a single-time credential consumption semantics is fundamental to secure anonymous authentication.

Table 5.2: HALP Component Interaction Matrix

From	To	Phase	Information Exchanged
$\mathcal{H}$	$\mathcal{I}$	Issuance	DID auth, Pedersen commitment, ZK proof
$\mathcal{I}$	$\mathcal{H}$	Issuance	Blind BBS+ signature, public key
$\mathcal{H}$	$\mathcal{V}$	Auth Start	Authentication request
$\mathcal{V}$	$\mathcal{H}$	Auth Start	Challenge ch, registry root
$\mathcal{V}$	$\mathcal{R}$	Auth	Registry root query
$\mathcal{H}$	$\mathcal{R}$	Auth	Non-membership proof request
$\mathcal{R}$	$\mathcal{H}$	Auth	Merkle non-membership proof
$\mathcal{H}$	$\mathcal{V}$	Auth Complete	zk-SNARK proof, nullifier, pseudonym
$\mathcal{V}$	$\mathcal{R}$	Auth Complete	Nullifier recording
$\mathcal{V}$	$\mathcal{H}$	Auth Complete	JWT session token

5.2 Protocol Flow of HALP

The HALP credential issuance protocol enables privacy-preserving credential distribution through a four-phase blind signature mechanism that preserves holder anonymity while ensuring cryptographic integrity (Figure 5.3). This protocol builds upon the BBS+ signature scheme and Pedersen commitment constructions to achieve perfect blindness during the signing process.

5.2.1 Credential Issuence

Phase 1: Commitment Generation
The credential issuance process begins with the holder $\mathcal{H}$ establishing cryptographic commitments over their private attributes(Figure 5.2). The holder first generates a cryptographically secure master secret ms by sampling uniformly from the scalar field $\mathbb{F}_r$ of the BLS12-381 elliptic curve. This master secret serves as the fundamental identity anchor that enables pseudonym derivation and nullifier generation in subsequent authentication protocols.
Following master secret generation, the holder selects a random blinding factor $r \leftarrow \mathbb{F}_r$ to ensure computational hiding of the commitment contents. The holder then constructs a Pedersen commitment $C = \text{Com}(ms, attr_1, \ldots, attr_k; r)$ that cryptographically binds the master secret and all credential attributes $\{attr_1, \ldots, attr_k\}$(Figure 5.2) while maintaining perfect hiding under the discrete logarithm assumption [2].
To enable issuer verification without revealing committed values, the holder generates a zero-knowledge proof π_C demonstrating knowledge of the commitment opening. This proof (Figure 5.2) follows the Schnorr-based proof-of-knowledge construction: $\pi_C = \text{PoK}(ms, attr_1, \ldots, attr_k, r : C = \text{Com}(ms, attr_1, \ldots, attr_k; r))$, providing computational soundness that prevents malicious holders from committing to invalid or inconsistent attribute sets.
Phase 2: Authentication and Request Submission

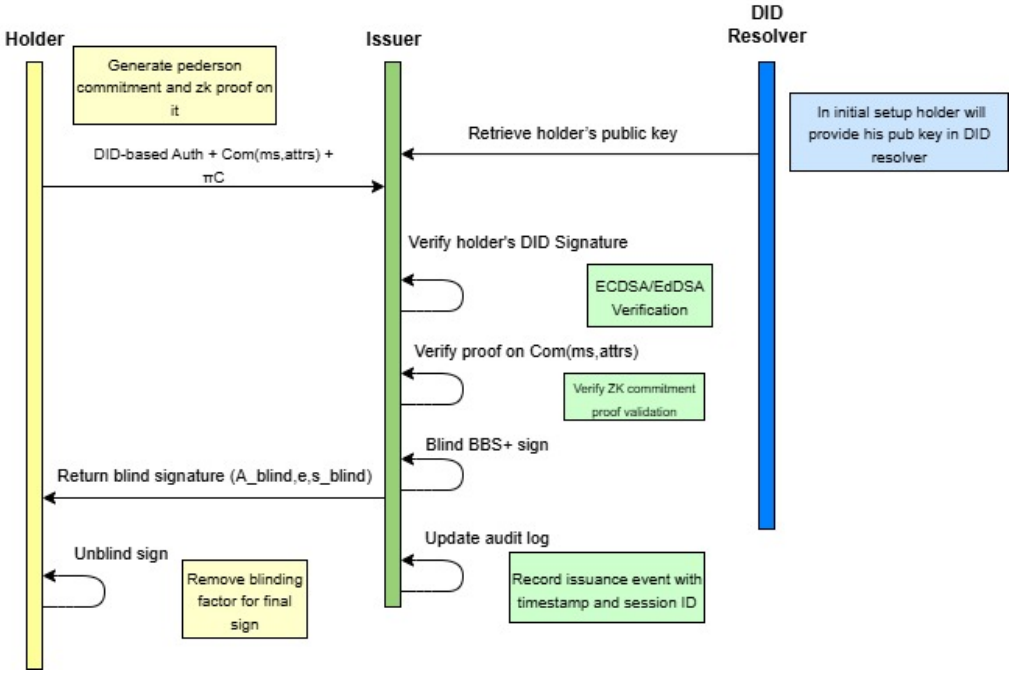


Figure 5.3: HALP Issuence Sequence Diagram

The holder initiates credential issuance by authenticating to the issuer $\mathcal{I}$ using decentralized identifier (DID) based authentication mechanisms (Figure 5.3) following W3C DID Core specifications. This authentication can utilize either mutual Transport Layer Security (mTLS) with DID-bound certificates or direct cryptographic signatures over session challenges using the holder's DID authentication key.

Upon authentication, the holder transmits the credential request containing the commitment C, the zero-knowledge proof π_C, and any attributes $\{attr_j\}_{j\in\text{revealed}}$ that must be disclosed for eligibility verification. The selective disclosure of certain attributes enables the issuer to validate holder eligibility (such as age, citizenship, or organizational membership) while maintaining privacy over sensitive personal information embedded within the commitment.

The issuer performs comprehensive verification procedures upon receiving the credential request. First, the issuer validates the DID-based authentication by resolving the holder's DID Document and verifying the cryptographic signature or certificate chain. Subsequently, the issuer executes the zero-knowledge proof verification algorithm (Figure 5.3) to ensure the holder possesses valid openings to the submitted commitment C. Finally, the issuer validates that all revealed attributes $\{attr_j\}_{j\in\text{revealed}}$ satisfy the credential issuance policy requirements, such as minimum age thresholds or organizational membership criteria.

Phase 3: Blind Signature Generation

Following successful authentication and proof verification(Figure 5.3), the issuer proceeds with blind signature generation using the BBS+ signature scheme [32]. The issuer constructs a message vector $\vec{m} = (C, \{attr_j\}_{j\in\text{revealed}}, \text{metadata})$ that includes the Pedersen commitment, any disclosed attributes, and relevant metadata such as issuance timestamps, credential validity periods, and credential type identifiers.

The issuer generates the BBS+ signature $(A, e, s) = \text{Sign}(sk_{\mathcal{I}}, \vec{m})$ using its private signing key $sk_{\mathcal{I}}$. This signature computation involves selecting a random exponent $e \leftarrow \mathbb{F}_r$, computing the signature point $A = (g_1 \cdot \prod_{i=1}^{|\vec{m}|} h_i^{m_i} \cdot g_2^s)^{1/(sk_{\mathcal{I}}+e)}$ where $s \leftarrow \mathbb{F}_r$

represents signature randomness, and h_i denotes message-specific generators derived from the BLS12-381 curve.
The issuer transmits the complete signature tuple $(A, e, s, \vec{m}, pk_{\mathcal{I}})$ to the holder, providing all necessary components for signature verification and subsequent credential deployment. The inclusion of the issuer's public key $pk_{\mathcal{I}}$ enables the holder to perform immediate signature validation and supports verifier authentication during credential presentation.
Phase 4: Signature Verification and Storage
Upon receiving the signature from the issuer, the holder executes the BBS+ signature verification algorithm to ensure cryptographic correctness and authenticity. The verification process computes $\text{Verify}(pk_{\mathcal{I}}, \vec{m}, (A, e, s))$ by checking the pairing equation $e(A, pk_{\mathcal{I}} \cdot g_2^e) = e(g_1 \cdot \prod_{i=1}^{|\vec{m}|} h_i^{m_i} \cdot g_2^s, g_2)$, which must evaluate to the identity element in the target group G_T for valid signatures.
Successful signature verification (Figure 5.2) confirms that the issuer has correctly signed the committed attributes and metadata without learning their specific values, thus preserving the blindness property essential for anonymous credentials. The holder concludes the issuance protocol by securely storing the complete credential structure $\text{cred} = (A, e, s, \vec{m}, ms, \{attr_i\}_{i=1}^k)$, which includes the BBS+ signature components, the signed message vector, the master secret, and all original attributes required for subsequent authentication proof generation.
This credential storage enables the holder to generate unlinkable authentication proofs across multiple verifier domains while maintaining selective disclosure capabilities and preventing correlation attacks through the cryptographic properties of the underlying BBS+ signature scheme and the privacy-preserving pseudonym derivation mechanisms integral to the HALP protocol.

5.2.2 Anonymous Authentication Protocol

The HALP anonymous authentication protocol allows a holder $\mathcal{H}$ to prove possession of a valid BBS + credential and obtain a session token from a verifier $\mathcal{V}$ under domain d without revealing any secret attributes. Let *credID* denote the unique identifier of the stored credential, ms the holder's master secret, and $pk_{\mathcal{I}}$ the issuer's public key. The protocol proceeds in four phases, leveraging a Merkle-tree nullifier registry $\mathcal{R}$ for replay prevention and a SNARK circuit *circuitID* for proof verification.
[H] [1] **Input**: Holder $\mathcal{H}$ with credential $(A, e, s, credID)$, Verifier $\mathcal{V}$, domain identifier d **Output**: JWT session token or $\perp$

Phase 1: Challenge Generation
The authentication flow begins when the holder $\mathcal{H}$ initiates a request to authenticate with verifier $\mathcal{V}$ for a specific service domain d, where d represents a unique identifier that scopes authentication sessions to prevent cross-domain linkability. Upon receiving this request, the verifier generates a cryptographically secure challenge ch by sampling uniformly from the scalar field $\mathbb{F}_r$ of the BLS12-381 curve, ensuring 128-bit security and preventing replay attacks across authentication sessions. The verifier then queries the nullifier registry $\mathcal{R}$ to obtain the current Merkle tree root $root_{\mathcal{R}}$ using the GetRoot() operation, which returns a cryptographic commitment to all previously consumed nullifiers in the system. Finally, the verifier transmits the authentication parameters $(ch, root_{\mathcal{R}}, circuitID)$ to the holder, where *circuitID*

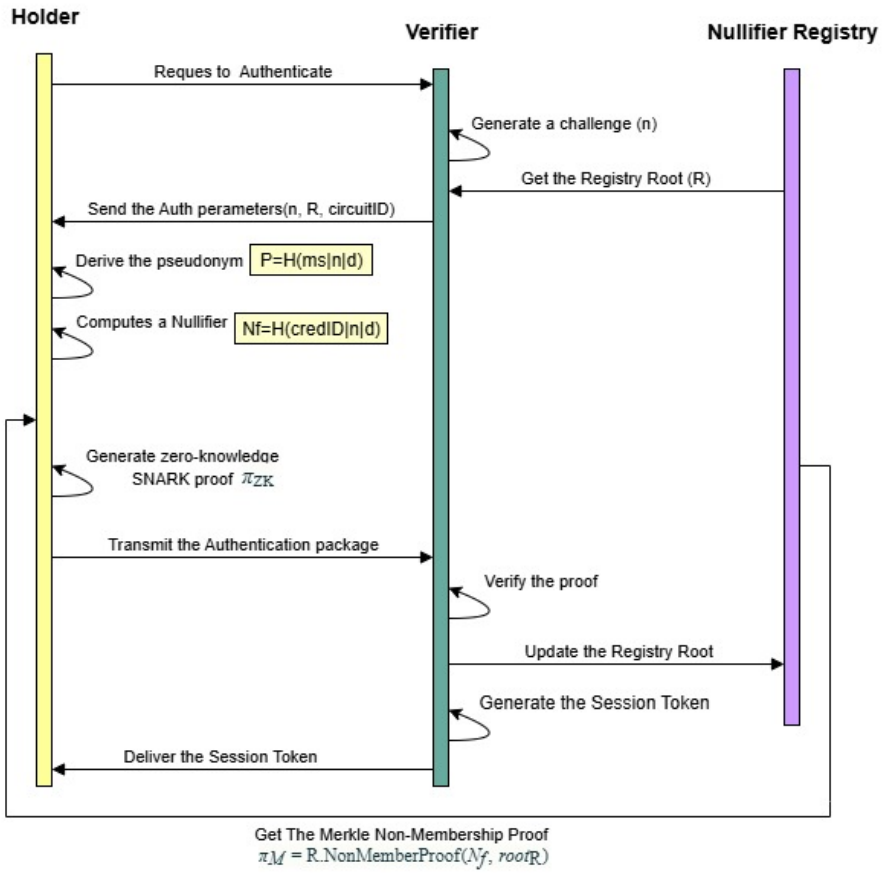


Figure 5.4: HALP Verification

specifies the zero-knowledge proof circuit identifier that defines the authentication constraints and ensures both parties use compatible proof systems.
$\mathcal{H} \rightarrow \mathcal{V}$: (Table: 5.2) Request authentication for domain d $\mathcal{V}$: Sample random challenge $ch \leftarrow \mathbb{F}_r$ $\mathcal{V}$: Query nullifier registry root $root_{\mathcal{R}} = \mathcal{R}.\text{GetRoot}()$ $\mathcal{V} \rightarrow \mathcal{H}$: Send $(ch, root_{\mathcal{R}}, circuitID)$

Phase 2: Proof Generation

In this phase, the holder $\mathcal{H}$ prepares all values and proofs needed for zero-knowledge authentication:

- The holder samples a fresh session nonce $n \leftarrow \mathbb{F}_r$, where $\mathbb{F}_r$ denotes the scalar field of order r for the BLS12-381 curve.
- Using the master secret ms, session nonce n, and domain identifier d, the holder derives a session-specific pseudonym

$$P = H\big(ms \,\|\, n \,\|\, d\big),$$

  where H is a collision-resistant hash function (e.g., Poseidon) that ensures unlinkability across sessions and domains.
- Similarly, the holder computes a nullifier

$$N_f = H\big(credID \,\|\, n \,\|\, d\big),$$

  binding the unique credential identifier $credID$ to the session and domain in order to prevent double-spending of authentication proofs.

- The holder then obtains a Merkle non-membership proof

$$\pi_M = \mathcal{R}.\text{NonMemberProof}(N_f,\ root_{\mathcal{R}}),$$

  demonstrating that N_f is not already present in the nullifier registry whose current Merkle root is $root_{\mathcal{R}}$.

- Next, the holder assembles the public input vector

$$pub = \big(P,\ N_f,\ root_{\mathcal{R}},\ ch,\ pk_{\mathcal{I}}\big),$$

  which contains all values that the verifier needs to check publicly, including the challenge ch and the issuer's public key $pk_{\mathcal{I}}$.

- The private input vector is then formed as

$$priv = \big(ms,\ n,\ A,\ e,\ s,\ \pi_M\big),$$

  embedding the master secret ms, nonce n, BBS+ signature components (A, e, s), and the non-membership proof π_M.

- Finally, the holder computes the zero-knowledge SNARK proof

$$\pi_{\text{ZK}} = \text{Prove}\big(circuitID,\ pub,\ priv\big),$$

  where *circuitID* specifies the SNARK circuit enforcing the correct relationships among public and private inputs.

Phase 3: Proof Submission
At this stage, the holder $\mathcal{H}$ transmits the authentication package (Figure 5.4) to the verifier $\mathcal{V}$. Specifically, $\mathcal{H}$ sends the tuple

$$\big(ch,\ \pi_{\text{ZK}},\ P,\ N_f,\ root_{\mathcal{R}}\big)$$

where:

- ch is the challenge previously sampled by $\mathcal{V}$, binding the proof to this specific authentication session.
- π_{ZK} is the zero-knowledge SNARK proof attesting to correct computation of the pseudonym P, nullifier N_f, and possession of a valid BBS + credential.
- $P = H(ms\|n\|d)$ is the session-specific pseudonym that serves as the holder's anonymized identifier for domain d.
- $N_f = H(credID\|n\|d)$ is the nullifier that prevents reuse of this authentication instance, linked to the unique credential identifier.
- $root_{\mathcal{R}}$ is the Merkle root of the nullifier registry at the time of proof generation, ensuring the proof reflects the current registry state.

This submission enables $\mathcal{V}$ to verify both proof validity and nullifier freshness in the next phase.

Phase 4: Verification & Token Issuance

In the final phase, the verifier $\mathcal{V}$ performs two critical checks before issuing a session token. First, it executes the SNARK verification procedure

$$\text{Verify}(\textit{circuitID},\ \textit{pub},\ \pi_{\text{ZK}})\ =\ 1,$$

ensuring that the proof corresponds to the correct circuit *circuitID* and public input vector $\textit{pub} = (P, N_f, \textit{root}_{\mathcal{R}}, \textit{ch}, \textit{pk}_{\mathcal{I}})$, thereby confirming the holder's knowledge of a valid credential without revealing secrets. Second, the verifier queries the nullifier registry $\mathcal{R}$ to confirm that the nullifier N_f has not been previously recorded, guaranteeing one-time use of the authentication proof. If both checks succeed, $\mathcal{V}$ adds the nullifier to the registry using

$$\mathcal{R}.\text{Add}\big(N_f, \{P, \text{timestamp}, d\}\big),$$

binding N_f to the pseudonym P, the current time, and domain d. The verifier then issues a JSON Web Token

$$\textit{JWT} = \text{JWT\_Create}\big(\textit{sub} = P,\ \textit{aud} = d\big),$$

with subject P and audience d, and returns it to the holder. If either verification fails, the verifier returns $\perp$, indicating authentication failure.

5.2.3 Session Management and Revocation

The HALP protocol supports fine-grained session management through selective nullifier revocation, enabling immediate termination of individual authentication sessions without affecting other active sessions or requiring credential reissuance. This capability is essential for enterprise environments where session-specific access control and emergency revocation procedures are required.

Input: Target nullifier N_f to revoke, nullifier registry $\mathcal{R}$ **Output**: Updated registry state with revoked session

Immediate Revocation $\mathcal{R}$: Mark nullifier N_f as revoked with current timestamp $\mathcal{R}$: Update indexed Merkle tree structure to reflect revocation $\mathcal{R}$: Broadcast updated root $\textit{root}'_{\mathcal{R}} = \text{UpdateRoot}()$ to all verifiers

Garbage Collection $\mathcal{R}$: Scan registry for nullifiers exceeding session TTL threshold $\mathcal{R}$: Archive expired nullifiers $\{N_{f,\text{exp}}\}$ to cold storage $\mathcal{R}$: Reconstruct tree structure $\mathcal{T}'$ excluding expired entries $\mathcal{R}$: Propagate new root $\textit{root}''_{\mathcal{R}}$ and notify all verifiers

The revocation mechanism operates through two complementary processes. **Immediate revocation** enables administrators to terminate specific sessions by marking their associated nullifiers N_f as revoked within the registry $\mathcal{R}$. This operation updates the indexed Merkle tree structure to include the revocation record with a timestamp, ensuring that subsequent authentication attempts using the same nullifier will be rejected. The registry then broadcasts the updated root $\textit{root}'_{\mathcal{R}}$ to all participating verifiers, providing cryptographic assurance that the revocation has been properly recorded and will be enforced across the entire system.

Garbage collection addresses the long-term scalability of the nullifier registry by automatically removing expired session records. The registry periodically identifies nullifiers that have exceeded their session time-to-live (TTL) threshold, archives these expired entries to cold storage for audit purposes, and reconstructs the Merkle tree structure without the expired nullifiers. This process generates a new registry root $root''_{\mathcal{R}}$ that reflects the reduced active nullifier set, enabling efficient registry maintenance while preserving the cryptographic integrity required for non-membership proof generation.

The capability of selective revocation will provide many benefits to the traditional credential revocation mechanisms. Whereas certificate revocation lists or credential revocation registries invalidate entire credentials, nullifier-based revocation by HALP follows a more granular level of operation, revoking complete authentication sessions instead of invalidating possessing credential holders (even though credential revocation can be implemented during a given session). This fine-grained control offers superior access-management scenarios, such as the denial of access to infected machines, yet allowing access in trusting environments, or providing time-based access control, which automatically expires without any human intervention

The fact that the indexed Merkle tree has cryptographic properties ensures that revocation operations maintain system security and that they can be verified effectively. Verifiers are able to authenticate the existing state of the registry by questioning the new root values that are being transmitted, and holders can still produce non-membership proofs of new nullifiers using the new tree construction. Combining the immediate Revocation with the Automated Garbage collection creates a complete system lifecycle-management framework, which scales effectively with the usage of the system and maintains the privacy and security characteristics needed by the anonymous authentication systems.

Bibliography

[1] S. Goldwasser, S. Micali, and C. Rackoff, *The knowledge complexity of interactive proof systems*, 1989.

[2] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology – CRYPTO '91*, ser. Lecture Notes in Computer Science, Seminal paper introducing Pedersen commitments and their security analysis, vol. 576, Springer, 1991, pp. 129–140. DOI: 10.1007/3-540-46766-1_9.

[3] S. Halevi and S. Micali, "Practical and provably-secure commitment schemes from collision-free hashing," in *Advances in Cryptology—CRYPTO '96*, ser. Lecture Notes in Computer Science, vol. 1109, Springer, 1996, pp. 201–215. DOI: 10.1007/3-540-68697-5_16.

[4] S. Brands, *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, 2000.

[5] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," *Advances in Cryptology—EUROCRYPT*, 2001.

[6] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *Advances in Cryptology – EUROCRYPT 2001*, Defines the Idemix anonymous credential system, influential in privacy-preserving authentication systems., Springer, 2001, pp. 93–118. DOI: 10.1007/3-540-44987-6_6.

[7] J. Camenisch and A. Lysyanskaya, "Design and implementation of the idemix anonymous credential system," in *Proceedings of the 2003 RSA Conference*, RSA Conference, 2003, pp. 1–20.

[8] S. Steinbrecher and S. Köpsell, "Modelling unlinkability," in *Third International Workshop on Privacy Enhancing Technologies (PET 2003)*, Dresden, Germany, Mar. 2003.

[9] D. Boneh and X. Boyen, "Short signatures without random oracles and the q-strong diffie-hellman problem," in *Advances in Cryptology – EUROCRYPT 2004*, Introduces the q-SDH assumption and BBS signature scheme, foundational for anonymous credentials., Springer, 2004, pp. 56–73. DOI: 10.1007/978-3-540-24676-3_5.

[10] G. Barthe, B. Grégoire, S. Heraud, and S. Z. Béguelin, "Computer-aided security proofs for the working cryptographer," in *Advances in Cryptology—CRYPTO 2011*, ser. Lecture Notes in Computer Science, vol. 6841, Springer, 2011, pp. 71–90. DOI: 10.1007/978-3-642-22792-9_5.

[11] G. Barthe, B. Grégoire, C. Kunz, and B. Schmidt, "Verified computational differential privacy with applications to smart metering," in *2013 IEEE 26th Computer Security Foundations Symposium*, IEEE, 2013, pp. 287–301. DOI: 10.1109/CSF.2013.26.

[12] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *IEEE Symposium on Security and Privacy*, IEEE, 2013.

[13] E. Ben-Sasson *et al.*, *Succinct non-interactive zero knowledge for a von neumann architecture (zk-snarks)*, 2014. eprint: IACRePrint.

[14] E. Ben-Sasson, A. Chiesa, C. Garman, *et al.*, "Zerocash: Decentralized anonymous payments from bitcoin," in *IEEE Symposium on Security and Privacy*, IEEE Xplore, 2014.

[15] M. Chase and S. Meiklejohn, "Transparency overlays and applications," in *CCS '14*, ACM Digital Library, 2014.

[16] E. B. Sasson, A. Chiesa, C. Garman, *et al.*, "Zerocash: Decentralized anonymous payments from bitcoin," in *IEEE Symposium on Security and Privacy*, IEEE, 2014.

[17] A. Azaria *et al.*, "Blockchain-based privacy-preserving access to health records," in *IEEE Xplore*, 2016.

[18] J. Groth, "On the size of pairing-based non-interactive arguments," in *EUROCRYPT 2016*, Presents Groth16 zk-SNARK protocol for efficient zero-knowledge proof systems., Springer, 2016, pp. 305–326. DOI: 10.1007/978-3-662-49890-3_12.

[19] A. Kosba *et al.*, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *IEEE Symposium on Security and Privacy*, IEEE Xplore, 2016.

[20] S. Bowe *et al.*, *Bls12-381: New zk-snark elliptic curve construction*, IACR Cryptology ePrint Archive, Report 2017/1050, https://eprint.iacr.org/2017/1050, 2017. eprint: 2017/1050.

[21] B. Bünz *et al.*, *Bulletproofs: Short proofs for confidential transactions*, 2018. eprint: IACRePrint.

[22] B. Edgington, *Bls12-381 for the rest of us*, HackMD Documentation, Accessed: 2024, 2019. [Online]. Available: https://hackmd.io/@benjaminion/bls12-381.

[23] A. Sonnino *et al.*, "Coconut: Threshold issuance selective disclosure credentials," in *IACR ePrint*, 2019.

[24] V. Buterin, J. Woods, and A. Mavridou, "Aztec: Privacy-preserving smart contracts using indexed merkle trees," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)*, ACM, 2020, pp. 1009–1023. DOI: 10.1145/3372297.3423353.

[25] Z. Foundation, *Moonmath manual*, Available at https://github.com/zcash/moonmath, Accessed: [Insert Last Accessed Date], 2020.

[26] IETF CFRG, *Pairing-friendly curves*, Internet Draft, IRTF Crypto Forum Research Group, draft-irtf-cfrg-pairing-friendly-curves-10, Jun. 2020. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-pairing-friendly-curves-10.

[27] L. Grassi, D. Khovratovich, C. Rechberger, A. Roy, and M. Schofnegger, "Poseidon: A new hash function for zero-knowledge proof systems," in *30th USENIX Security Symposium (USENIX Security 21)*, USENIX Association, 2021, pp. 519–535. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/grassi.

[28] Circom, *Circom snarkjs documentation*, 2022. [Online]. Available: https://circom.io.

[29] K. George, *The mathematical mechanics behind the groth16 zero-knowledge proof system*, https://kayleegeorge.github.io/math110_WIM.pdf, Details Groth16 proof verification requiring three pairing checks, resulting in constant verification time., 2022.

[30] Z. Protocol, *Zcash protocol specification, version 2022.3.8*, 2022.

[31] W3C Credentials Community Group, *Decentralized identifiers (dids) v1.0: Core architecture, data model, and representations*, https://www.w3.org/TR/did-core/, May 2022.

[32] IETF CFRG, *Bbs + signatures*, Internet Draft, IRTF Crypto Forum Research Group, draft-irtf-cfrg-bbs-signatures-11, Dec. 2023. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-bbs-signatures-11.

[33] M. Johnson *et al.*, "Optimizing groth16 zk-snarks," in *Crypto*, Springer Link, 2024.

[34] A. Kumar *et al.*, "Privacy-preserving selective disclosure with bls signatures," in *NDSS Symposium*, NDSS, 2024.

[35] A. S. Rajasekaran, A. Maria, J. Lloret, and S. Dannana, "Tpaas: Trustworthy privacy-preserving anonymous authentication scheme for online trading environments," *PLOS ONE*, vol. 19, no. 11, e0307738, 2024. DOI: 10.1371/journal.pone.0307738.

[36] O. Sanders and J. Traoré, "Compact issuer-hiding authentication: Application to anonymous credentials," *Proceedings on Privacy Enhancing Technologies*, vol. 2024, no. 3, pp. 645–658, 2024. DOI: 10.2478/popets-2024-0097.

[37] J. Smith *et al.*, "Ephemeral key rotation for unlinkable authentication," in *IEEE Symposium on Security and Privacy*, IEEE Xplore, 2024.

[38] X. Zhao, F. Xia, H. Xia, Y. Mao, and S. Chen, "A zero-knowledge-proof-based anonymous and revocable scheme for cross-domain authentication," *Electronics*, vol. 13, no. 14, p. 2730, 2024. DOI: 10.3390/electronics13142730.

[39] R. Patel and S. Gupta, "Threshold revocation via pvss," in *Eurocrypt*, Springer Link, 2025.