Dynamic Predicate Systems

Fadi Barbàra

[1] University of Rome La Sapienza
[2] Fairgate Labs
fadi.barbara@gmail.com

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

1.1 Motivation

Identity represents us, for better or worse, in all our interactions. It is the cornerstone of modern digital and physical interactions, enabling trust, accountability, and personalized services.

There are fundamentally three identity models: centralized, where a single authority controls identity; federated, where multiple authorities coordinate through trust relationships; and decentralized, where individuals maintain control of their own identity. However, in practice, identities today are predominantly generated centrally—whether by government entities or private corporations—with a trusted issuer at the core.

What we seek to achieve is the ability to maintain privacy while interacting with non-trusted service providers. Privacy, in this context, is not merely a preference but a form of security: it protects individuals from unauthorized data aggregation, profiling, and potential misuse of personal information.

Verifiable Credentials (VCs) and Verifiable Presentations (VPs) represent a promising direction toward this goal. They enable selective disclosure, allowing users to prove specific attributes about themselves without revealing unnecessary information. The cryptographic foundations of VCs ensure integrity and authenticity while giving users control over what they share.

However, current VC/VP systems face significant limitations. Most critically, they lack support for dynamic identity attributes and cannot efficiently prove in zero-knowledge that a credential field satisfies range predicates. For instance, proving that one's credit score is above a certain threshold without revealing the exact score remains impractical without predefining categorical buckets or requiring credential reissuance.

1.2 Related Works

Anonymous and Verifiable Credentials (AVCs) leverage zero-knowledge proofs to demonstrate properties of a credential without exposing the underlying identity or specific values. This enables users to prove they satisfy certain conditions while preserving anonymity. A canonical example is credit score verification: rather than revealing the exact numerical score, an AVC could prove that the score exceeds a required threshold, enabling access to financial services without disclosing precise financial standing.

For AVCs to be deployable in real-world systems, they must satisfy critical properties, foremost among them **unlinkability**. A credential system achieves unlinkability when repeated verifications of the same attribute reveal nothing about the user's identity or the fact that multiple verifications have occurred. For instance, if a user repeatedly proves their credit score meets certain thresholds at different institutions, no verifier should learn their actual score, identity, or even recognize that the same individual has been verified before.

Unlinkability manifests in two forms: **Verifier–Verifier unlinkability**, which ensures that proofs remain unlinkable even when multiple verifiers (or the same verifier across sessions) attempt to correlate them; and **Issuer–Verifier unlinkability**, which maintains unlinkability even when the credential issuer colludes with verifiers.

The literature presents four principal families of Anonymous Credential Protocols (ACPs):

- Camenisch–Lysyanskaya (CL03) [**?**],
- Pointcheval–Sanders (PS16) [**?**],
- Selective Disclosure JWTs (SD-JWT) [**?**], and
- Boneh–Boyen–Shacham signatures (BBS, BBS+, BBS#) [**?**,**?**,**?**].

Among these, SD-JWTs and BBS signatures demonstrate the highest efficiency, achieving near-equivalent performance in both space and time complexity. However, SD-JWTs fail to provide Issuer–Verifier unlinkability, making them unsuitable for scenarios demanding strong anonymity guarantees. BBS signatures, by contrast, satisfy both forms of unlinkability while maintaining practical efficiency, positioning them as the current state-of-the-art for privacy-preserving credential systems.

1.3 Current Limitations

A critical gap remains in the current state-of-the-art: there are no range proof schemes that are efficient enough for practical deployment with the BBS family of signatures. This limitation reveals a deeper architectural problem in how credentials are designed and used.

To return to our credit score example: a credential holder wishing to prove their score exceeds 700 faces two equally unsatisfactory options. Either the issuer must embed categorical attributes (e.g., "poor," "medium," "high," "perfect")

alongside the numerical score—requiring reissuance whenever finer-grained categories are needed—or the system must rely on inefficient range proof constructions that remain impractical for real-world deployment.

This highlights a critical research need: **Dynamic Predicate Systems**, where credentials can:

- Reference external state provably
- Update specific attributes without full reissuance
- Compose multiple credentials for range proofs
- Handle time-based deprecation gracefully

Current systems force a trade-off between credential longevity and predicate precision, which is a fundamental limitation that needs addressing.

2 BBS Signatures

BBS signatures are short group signatures that enable efficient selective disclosure and unlinkability. Their security relies on the q-Strong Diffie-Hellman (q-SDH) assumption in pairing-friendly groups.

Definition 1 (q-SDH Assumption). *Let G_1, G_2, G_T be groups of prime order p with a bilinear pairing $e : G_1 \times G_2 \to G_T$. Let $g_1 \in G_1$ and $g_2 \in G_2$ be generators. Given the tuple $[g_1, g_1^x, g_1^{x^2}, \ldots, g_1^{x^q}, g_2, g_2^x]$ for some unknown $x \xleftarrow{\$} \mathbb{Z}_p^*$, it is computationally infeasible to compute $\frac{1}{g_1^{x+e}}$ for some $e \in \mathbb{Z}_p$.*

2.1 Signature Scheme

A BBS signature scheme consists of three algorithms:

Definition 2 (BBS Signature Scheme). *A BBS signature scheme is a tuple of polynomial-time algorithms* $(\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify})$ *where:*

- $\mathsf{Setup}(1^\lambda, L) \to (\mathsf{pp}, \mathsf{sk}, \mathsf{pk})$*: Given security parameter λ and message length L, outputs public parameters* pp*, secret key* sk*, and public key* pk*.*
- $\mathsf{Sign}(\mathsf{sk}, M) \to \sigma$*: Given secret key* sk *and message vector $M = (m_1, m_2, \ldots, m_L)$, outputs signature σ.*
- $\mathsf{Verify}(\mathsf{pk}, M, \sigma) \to \{0, 1\}$*: Given public key* pk*, message M, and signature σ, outputs* 1 *if valid,* 0 *otherwise.*

2.2 Issuance

The issuance protocol proceeds as follows:

Setup. Generate the cryptographic parameters:

- Prime p and groups $G_1 \neq G_2 \neq G_T$ of order p
- Pairing $e : G_1 \times G_2 \to G_T$
- Generators $g_1 \in G_1$ and $g_2 \in G_2$
- Sample secret key $x \xleftarrow{\$} \mathbb{Z}_p^*$
- Compute public key $\mathsf{pk} = g_2^x$

Sign. To sign a message vector $M = (m_1, m_2, \ldots, m_L)$:

1. Sample generators $h_1, h_2, \ldots, h_L \xleftarrow{\$} G_1$ (deterministically created from a constant seed and a PRF)
2. Sample $e \xleftarrow{\$} \mathbb{Z}_p$ (deterministically created from x and M)
3. Calculate A as:
$$A = (g_1 h_1^{m_1} h_2^{m_2} \cdots h_L^{m_L})^{\frac{1}{x+e}}$$
4. Output signature $\sigma = (A, e)$

The signature on M is the tuple (A, e).

2.3 Verification

To verify a signature $\sigma = (A, e)$ on message $M = (m_1, m_2, \ldots, m_L)$ with public key $\mathsf{pk} = g_2^x$:

1. Compute $B = g_1 h_1^{m_1} h_2^{m_2} \cdots h_L^{m_L}$
2. Check the pairing equation:
$$e(A, g_2^x \mathsf{pk}) = e(B, g_2)$$
3. Equivalently, verify:
$$e\left(B^{\frac{1}{x+e}}, g_2^{e+x}\right) = e(B, g_2)$$

The signature is valid if and only if the pairing equation holds. The security of this verification relies on the q-SDH assumption, which ensures that without knowledge of the secret key x, it is computationally infeasible to forge valid signatures.

2.4 PoK & Selective Disclosure

3 Dynamic Predicate Systems: Research Directions

4 Conclusion