Measuring the Performance of Candidate Verifiable Credential Schemes for the EU Digital Identity Wallet

Angelos Ioannis Lagos
t8210079@aueb.gr
AUEB
Athens, Greece

Diomidis Spinellis
dds@aueb.gr
AUEB & TU Delft
Greece & Netherlands

Nikolaos Alexopoulos
alexopoulos@aueb.gr
AUEB
Athens, Greece

Abstract

Digital identity systems face an inherent tension between verifiability, non-transferability, and privacy. While current European Digital Identity Wallet prototypes rely on Selective-Disclosure JWTs (SD-JWTs) to achieve minimal disclosure and backward compatibility, these designs fail to uphold the strict three-way unlinkability requirement mandated by EU regulations. Schemes based on BBS signatures have been proposed as a possible solution to this problem. While these schemes satisfy the stated requirements, their use of complex cryptographic operations incurs a non-negligible performance overhead, especially relevant for mobile devices. To empirically investigate this overhead, we construct an extendable benchmarking suite for end-to-end wallet operations. Using our suite, we perform experiments comparing the performance of several privacy-preserving verifiable credential implementations on a desktop computer, a smartphone, a smartwatch, and a low-cost single-board computer. We find that performance varies significantly (often by an order of magnitude) based on hardware, employed scheme, and programming language of the implementation. Performance differences between schemes and implementations are more pronounced on resource-constrained devices. Positively, with the best-performing unlinkable BBS variant, a smartwatch needs around 1.5 seconds for credential presentation, latency that can be considered practically acceptable. Our results inform discussions on the transition to privacy-preserving mobile identity wallets, while our open-source suite can be used to benchmark future implementations.

CCS Concepts

• **Software and its engineering** → *Software performance*; • **Human-centered computing** → *Empirical studies in ubiquitous and mobile computing*.

Keywords

benchmark study, privacy-preserving digital identity, mobile devices

ACM Reference Format:
Angelos Ioannis Lagos, Diomidis Spinellis, and Nikolaos Alexopoulos. 2026. Measuring the Performance of Candidate Verifiable Credential Schemes for the EU Digital Identity Wallet. In *The 41st ACM/SIGAPP Symposium on Applied Computing (SAC '26), March 23–27, 2026, Thessaloniki, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3748522.3779904

SAC '26, Thessaloniki, Greece

ACM ISBN 979-8-4007-2294-3/2026/03
https://doi.org/10.1145/3748522.3779904

1 Introduction

Digital interactions increasingly underpin everything from banking and health-care to public administration and social participation. At the heart of every such interaction lies an *identity assertion*: a set of statements that convince a verifier—be it a website, a ticketing gate, or a border checkpoint—that an unknown party (holder) on the other side of the transaction is entitled to act in a certain capacity. Dominant solutions for this task remain *centralized* and *aggregative*: large identity providers mint static identifiers, observe every authentication event, and monetize the resulting behavioral dossiers. A single database compromise or policy misstep can therefore jeopardize the privacy, security, and even the civic rights of millions of citizens at the same time [15].

Recent regulatory initiatives such as the European Digital Identity Framework (eIDAS 2.0) and the parallel technical work at the W3C and OpenID Foundation point to a radically different model. *Decentralized Identifiers (DIDs)* [25] give users cryptographically verifiable names that are not anchored in a central registry, while *Verifiable Credentials (VCs)* [24] let trusted issuers sign arbitrary attribute sets that holders can later present in *selective-disclosure* form. In principle, those two building blocks should allow a citizen to prove just the facts that matter for a transaction (e.g. age verification) without leaving a globally unique correlation handle.

Unfortunately, theory and practice have not yet converged. The reference design endorsed by the first Architecture and Reference Framework for the European Digital Identity Wallet (EUDIW) relies on *Selective-Disclosure JWTs (SD-JWTs)*[1] [11, 27] bound to long-lived subject keys. A series of security reviews by leading EU cryptographers [2] has shown that this design leaks a stable fingerprint across presentations and therefore violates the strict *unlinkability* mandate of EU regulations[2]. This has led to growing attention on alternatives based on zero-knowledge primitives, such as schemes using BBS signatures [4, 8] with adaptations known as BBS+ [1] and BBS2023 [28], which provide unlinkability by-design but introduce higher computational and bandwidth costs.

Understanding these costs in practice is important for implementers, especially in the context of mobile devices which are provisioned to act as digital identity wallets. First, organizations need to have a clear understanding of the cost of unlinkability and its feasibility: how much computational overhead, payload increase, and architectural complexity will result from adopting unlinkable

[1] JSON Web Tokens (JWTs) are compact, signed data structures for conveying claims in web protocols [16].
[2] Article 5a §16 of Regulation (EU) 2024/1183 [22]

schemes compared to linkable ones. Second, even if they recognize the privacy advantages, organizations lack clear guidance on which implementation path to pursue: whether to continue with legacy SD-JWTs or, if they decide to adopt fully unlinkable verifiable credentials, which unlinkable signature scheme to use. This uncertainty risks delaying adoption and may result in fragmented solutions that undermine interoperability across the European Digital Identity ecosystem.

Although there are several implementations and associated benchmarks for the cryptographic primitives themselves[3], few studies have compared their performance in realistic verifiable credential flows, the most notable ones being by Flamini et al. [12, 13]. However, to the best of our knowledge, no study has benchmarked performance on real mobile phones, smartwatches, or low-cost matchbox-sized devices—the exact types of devices provisioned to be used as digital identity wallets, considering the EU mandate[4] for member states to provide digital wallets to every resident, accessible at no cost for acquisition and usage. This gap, particularly important for deployment decisions, motivates our work.

Contributions. We design and release a benchmarking suite[5] that automates performance measurement of credential issuance and presentation flows. This tool allows practitioners and policymakers to reproduce our results, extend the analysis to new schemes, and make evidence-based decisions about deployment. We then use the tool to provide an empirical benchmarking study of three candidate credential schemes—legacy SD–JWT, BBS+, and BBS2023—measuring end-to-end issuance, presentation, and verification performance on four distinct hardware platforms, including for the first time mobile phones and smartwatches.

Main Findings. Our measurements show that unlinkable credential schemes, despite their additional cryptographic complexity, achieve practical performance across all evaluated devices. The BBS 2023 implementation in Rust consistently delivered sub-second presentation times on smartphones and under two seconds on smartwatches, confirming the feasibility of unlinkable flows for real-world deployments. SD-JWT remains faster but provides weaker privacy guarantees, highlighting a clear trade-off between speed and unlinkability. Across all tests, implementation efficiency, particularly the choice of language and library, had a significant impact on latency. Payload sizes for BBS-based presentations were predictable and modest, decreasing with higher disclosure ratios and growing approximately linearly with the number of attributes. These findings collectively indicate that unlinkable verifiable credentials can meet the responsiveness and bandwidth requirements of the European Digital Identity Wallet.

2 Background and Related Work

In this section, we provide some background information on verifiable credentials and selective disclosure approaches and cover related work on the benchmarking of implementations of such approaches.

2.1 Verifiable Credentials (VCs)

Verifiable Credentials (VCs) are a standardized method for expressing claims about a subject in a manner that is cryptographically secure, privacy-respecting, and machine-verifiable. They serve as digital counterparts to traditional credentials like driver's licenses or academic degrees, enabling individuals to present proofs of attributes or qualifications in digital interactions.

The concept of VCs emerged from the need to address the limitations of traditional, centralized identity systems. Prior to the development of VCs, identity verification relied heavily on centralized authorities and federated identity models, such as SAML, which placed the identity provider at the center of the authentication process. These models often required users to disclose more information than necessary and lacked mechanisms for selective disclosure and user consent. The World Wide Web Consortium (W3C) recognized these challenges and initiated the development of a standardized data model for VCs. The first version, Verifiable Credentials Data Model 1.0, was published as a W3C Recommendation in November 2019 [23]. This specification provided a framework for expressing credentials on the web, defined the roles of issuers, holders, and verifiers, and established a data model that supports extensibility and interoperability.

A Verifiable Credential typically consists of the following:

- **Context**: A JSON-LD context that defines the terms used within the credential.
- **Type**: Specifies the type of credential, such as "VerifiableCredential" and any additional types relevant to the credential's content.
- **Issuer**: The entity that issues the credential, identified by a Decentralized Identifier (DID).
- **Issuance Date**: The date and time when the credential was issued.
- **Credential Subject**: The entity about whom the claims are made, also identified by a DID.
- **Credential Status**: Information about the current status of the credential, such as whether it has been revoked.
- **Proof**: A cryptographic proof, such as a digital signature, that allows verifiers to ascertain the credential's authenticity and integrity.

The W3C's Verifiable Credentials Data Model 2.0, published in 2023, builds upon the original specification by introducing enhancements for better interoperability and support for additional use cases [24]. It considers feedback from implementers and incorporates lessons learned from real-world deployments.

VCs are designed to be used in conjunction with Decentralized Identifiers (DIDs), which provide a means for entities to have unique, persistent identifiers that do not require a centralized registration authority. The combination of VCs and DIDs enables a decentralized identity ecosystem where individuals have greater control over their personal data and can present verifiable claims without relying on centralized intermediaries.

2.1.1 Roles in the Verifiable Credential Ecosystem. The Verifiable Credential (VC) ecosystem operates on a trust triangle involving three primary roles: the *Issuer*, the *Holder*, and the *Verifier* [24].

[3] For example https://github.com/docknetwork/crypto
[4] https://eur-lex.europa.eu/eli/reg/2024/1183/oj/eng
[5] https://github.com/Aglag257/Identity-Wallet-Flow-Benchmarking-Tool

The **Issuer** is an entity that creates and issues VCs to Holders after verifying certain attributes or qualifications. For example, a university may issue a VC to a student confirming the completion of a degree program. The Issuer cryptographically signs the credential to ensure its authenticity and integrity. The **Holder** is the entity that receives and stores VCs, typically in a digital wallet. The Holder can present these credentials to Verifiers when needed. The Holder has control over which credentials to share and with whom, enabling selective disclosure of information. The **Verifier** is an entity that requests and validates the authenticity and integrity of presented VCs to make informed decisions, such as granting access or services. The Verifier checks the digital signature of the VC and ensures that it has not been tampered with or revoked.

2.1.2 Unlinkability Requirements. EU regulations [22, 2] mandate strong technical unlinkability for the EU Digital Identity Wallet. We refer to this as *three-way unlinkability*: presentations should be unlinkable in a threat model that considers (a) malicious relying parties (verifiers), (b) identity providers (issuers), and (c) collusion among the two groups.

2.2 Selective disclosure approaches

2.2.1 SD-JWT. Traditional JSON Web Tokens (JWTs), as defined in RFC 7519 [16], encapsulate a set of claims in a compact, URL-safe format, signed by the issuer to ensure integrity and authenticity. However, JWTs inherently disclose all included claims upon presentation, which does not align with the principle of data minimization.

SD-JWTs, as specified in the IETF draft [11], introduce a method to selectively disclose claims within a JWT. This approach allows holders to reveal only the necessary information to verifiers, enhancing privacy without compromising the verifiability of the claims. The core mechanism of SD-JWT involves replacing the actual claim values in the JWT payload with salted hashes. During issuance, the issuer computes a hash of each claim value concatenated with a unique salt and includes these hashes in the JWT payload. The original claim values and their corresponding salts, termed *Disclosures* are provided to the holder separately. This design ensures that the JWT remains compact and does not expose sensitive information by default. The verifier can then compute the hash of each disclosed claim value and salt, comparing it against the corresponding hash in the JWT payload to validate authenticity. This process ensures that only the disclosed claims are revealed, while the integrity of the information is maintained. To further enhance security, especially against replay attacks or unauthorized use, SD-JWT supports Holder Binding. This feature binds the SD-JWT to a cryptographic key controlled by the holder. During presentation, the holder proves possession of the private key associated with the public key embedded in the SD-JWT, typically by signing a challenge or nonce provided by the verifier. This mechanism ensures that the SD-JWT cannot be used by unauthorized parties.

2.2.2 EU Cryptographers' Critique. EU cryptographers argue [2] that any scheme in which the issuer signs a stable hash structure, such as the aforementioned SD-JWT, violates the regulation's privacy mandate. Suppose Alice obtains an SD-JWT degree on Monday. Because the issuer signs the vector of digests once and for all, every future verifiable presentation (VP) must embed the same vector to remain verifiable. If Alice shows the credential at University A, the signed digests appear in University A's logs; when she later shows it at Bank B, identical digests appear in Bank B's logs. Colluding verifiers trivially stitch the two events together. Worse, if the issuer is summoned—perhaps years later—it can map those digests back to Alice's civil identity, reconstructing a complete usage profile.

The report further notes that issuing *multiple* unlinkable credentials would sacrifice non-transferability, because each credential would need a different subject key. Either the wallet stores hundreds of keys (unwieldy on current secure elements) or wallets can copy credentials between devices, undermining accountability. Anonymous credential systems, such as the ones based on the BBS signature scheme, avoid the dilemma by replacing static digests with re-randomized proofs, so every presentation is fresh yet still bound to a single long-term key [19, 11, 2].

2.2.3 Zero-knowledge proofs and BBS-based anonymous credentials. Zero-knowledge proofs [14] are cryptographic protocols that allow one party (the prover) to prove to another party (the verifier) that a statement is true without disclosing any additional information beyond the statement's truth. A famous example showcasing the concept of zero-knowledge proofs is the Ali Baba cave [21]. Zero-knowledge proofs have found many applications recently, especially in authentication systems and blockchains [3, 6].

BBS signatures, as introduced by Boneh, Boyen and Shacham [4] are cryptographic schemes that support efficient zero-knowledge proofs of knowledge of a valid message-signature pair. Using such schemes, a prover can prove to a verifier the possession of a valid signature by an issuer on a given message without revealing the signature itself and allowing linking. For a long time standardization efforts focused on the provably-secure variant of the scheme known as BBS+ [7, 1], since the initial construction was not proven secure. However, recently Tessaro and Zhu [28] proved the security of the initial more efficient construction, resulting in a scheme referred to as BBS2023 in this paper.

2.3 Related work

In the work most closely related to ours, Flamini et al. [12, 13] compared commitment-based and signature-based schemes (cmtList, merTree, CL, BBS, BBS+, PS) on credential size, proof generation, and verification times on both desktop (AMD64) and ARM devices (Raspberry Pi 3B+/4B). Their results indicated an order of magnitude performance difference (~1/10/100x) between each of the three devices tested. Notably, they did not perform tests on actual mobile phones or other smart devices and we could not locate a publicly available copy of their benchmarking code. The large differences observed in their experiments motivates our inclusion of mobile phones and lower-end smartwatch and matchbox-sized devices (RPi Zero) in our experiments.

Leuba [17] focused specifically on implementing BBS in the Swiss E-ID context, showing single-digit millisecond runtimes for issuance, proof generation, and verification on a desktop and an overhead of around 4-5x on a Raspberry Pi 4. The goal of their study was not to compare the overhead of different schemes, and they did not perform experiments on mobile devices.

Overall, to the best of our knowledge, we offer the first publicly available benchmarking suite for the comparison of complete end-to-end verifiable credential operations, while we perform the first comparative experiments on mobile phones, a smartwatch, and a low-cost matchbox-sized device.

3 Implementation

We implemented our benchmarking testbed in `Node.js`/TypeScript following the *OpenID4VCI* [18] issuance and the *OpenID4VP* [26] presentation specifications. Each run spins up three lightweight services, an Issuer, a Wallet, and a Verifier. These services exchange tokens and proofs over HTTP. Benchmarks are logged in JSONL format with per-run and aggregated statistics.

For the *linkable* baseline we use SD-JWT with ES256, implemented using the `jose`[6] library. For *unlinkable* selective-disclosure schemes, we integrate implementations of the BBS+ and BBS2023 variants. Specifically, we use `@mattrglobal/bbs-signatures`[7] for BBS+ implemented in Rust, `@digitalbazaar/bbs-signatures`[8] for a JavaScript-based BBS2023 implementation, and `@mattrglobal/pairing-crypto`[9] for a BBS2023 implementation in Rust. We note that all BBS implementations use the same pairing-friendly curve (BLS12-381) and thus offer the same level of security (~128 bits). Furthermore, the implementations of the two BBS variants in Rust come from the same organization, making them suitable candidates for comparison.

We chose these implementations based on two factors. First, the extent which the associated schemes are already embedded in existing industry standard specification discussions, making them more promising and probable candidate to be adopted in the future. For example, BBS signatures are already suggested by the cryptographers who critiqued the current status of the ARF [2] and also already have specifications from W3C [29]. Second, we considered implementations of different schemes and written in different programming languages to extract relevant observations about their performance. While other anonymous signature families exist and have several promising properties (e.g., BBS# [10]), they were excluded due to lack of production-ready libraries, since implementing our own would introduce efficiency imbalances.

Overall, our benchmarking harness extends a baseline SD-JWT setup using small scheme-specific adapters for each VC-library variant. Each implementation adds only a few hundred lines of JavaScript/TypeScript and preserves comparability across flows. Adding a new scheme or library variant mainly requires plugging its signing, disclosure/proof, and verification routines into this existing harness. Our implementation is available on GitHub.[10]

3.1 Design choices

We explicitly scope our implementation to isolate the costs of credential issuance, proof generation, and verification, while omitting components that would bias measurements, such as the following.

Discovery & metadata: OpenID Federation and full metadata endpoints are omitted. Endpoints are hard-coded since metadata adds network round-trips and parsing but does not affect signing or proof costs.

Trust establishment: The Verifier directly accepts keys provided in responses (no PKI). Real deployments must validate Issuer keys, but skipping this avoids bias toward specific trust frameworks and keeps cryptographic costs comparable.

Holder Binding: None of our implemented flows support holder (or key) binding because there are several approaches to this in the literature, and our choice could bias results.

These simplifications focus the benchmarks on the essential cryptographic operations and message sizes of each scheme. Specifically, the comparison measures the token exchanges mandated by OID4VCI and the selective-disclosure mechanics: *disclosure lists* for SD-JWT versus *derived proofs* for BBS/BBS+.

4 Measurements

This section presents a quantitative performance comparison of the four implementations presented in Section 3.

4.1 Experimental setup

All solutions were benchmarked over 50 complete issuance and presentation runs. For performance, we report the mean time to complete an operation across the 50 runs. We conducted experiments on the following devices.

Raspberry Pi Zero 2W: a "tiny 15 USD computer" equipped with a quad-core 64-bit ARM Cortex-A53 and 512 MB of RAM, running Raspberry Pi OS (64-bit). Used to evaluate lightweight edge deployment scenarios, acting as an identity token for users or as a verifier, e.g. in a smart lock.

Mobile phone (Xiaomi Poco M6 Pro): equipped with an octa-core chipset (2xCortex-A76 & 6xCortex-A55) and 8 GB of RAM, running HyperOS 2 (Android 15). This setup represents a typical mid-range mobile device.[11]

Desktop/Laptop (ASUS ROG Strix G713PU): with an AMD Ryzen 9 7940HX processor (16 cores, 32 threads, 2.4 GHz base), 32 GB RAM, and an NVIDIA GeForce RTX 4050 GPU. Experiments were executed under Windows 11 using the Windows Subsystem for Linux (WSL2).

Smartwatch (Google Pixel Watch): equipped with a dual-core Cortex-A53 and 2 GB of RAM, representing a wearable-class device, a natural candidate for future mobile identity wallets.

4.2 Research Questions

Our experiments try to answer the following research questions:

1. **Implementation choices:** What is the effect of scheme selection on performance, credential and payload sizes?
2. **Scalability:** What is the effect of the number of attributes on performance?
3. **Hardware capabilities:** What is the effect of different hardware platforms on performance?
4. **Programming language:** What is the effect of the implementation language on performance?

[6] https://github.com/panva/jose
[7] https://github.com/mattrglobal/bbs-signatures
[8] https://github.com/digitalbazaar/bbs-signatures
[9] https://github.com/mattrglobal/pairing_crypto
[10] https://github.com/Aglag257/Identity-Wallet-Flow-Benchmarking-Tool

[11] We also performed experiments on a Redmi Note 11 but results were similar, so we avoid reporting these measurements separately.

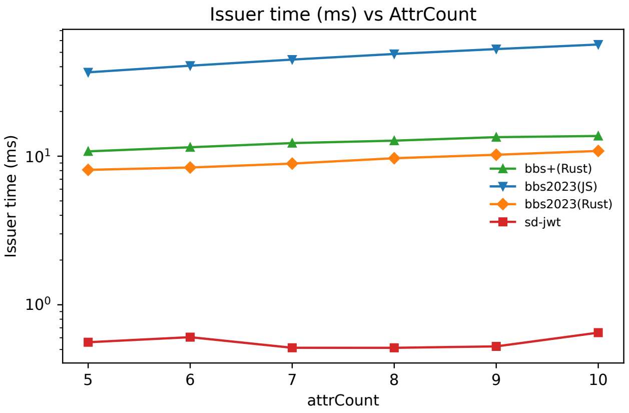

Figure 1: Issuer time vs. number of attributes.

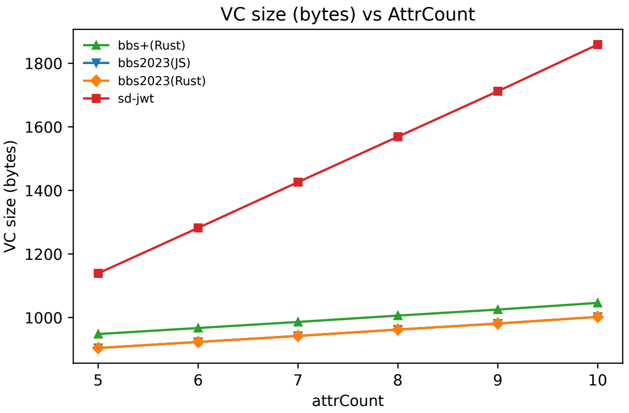

Figure 2: VC size vs. number of attributes.

We first study issuance performance, considering only the desktop implementation, as issuance is performed once for each credential by specialized entities. Then, we move to presentation where mobile devices are the most relevant. For presentation, we further differentiate between the prover (wallet) and the verifier. We focus specifically on prover performance, as this is the most relevant for mobile computing and is expected to be the most resource-intensive part of the process.

4.3 Credential issuance results

4.3.1 Issuer time. Figure 1 shows the time to issue a credential with variable attribute count. Overall, time is practically minimal for all implementations (<60 ms). The JavaScript implementation of BBS is the only one that requires more than 10 ms to issue a credential. Performance is sublinear to the number of attributes in the examined range.

4.3.2 VC size. Figure 2 shows the size of the verifiable credential stored by the device in bytes. BBS variants lead to considerably smaller credentials that scale better with the number of attributes, compared to the baseline. The two BBS2023 implementations produce VCs of the same size.

4.4 Credential presentation results

Here, we present our results for the benchmarks on the two parties involved in certificate presentation (prover-wallet and verifier). Figure 3 provides the legend for all multi-device plots that follow to aid in readability.

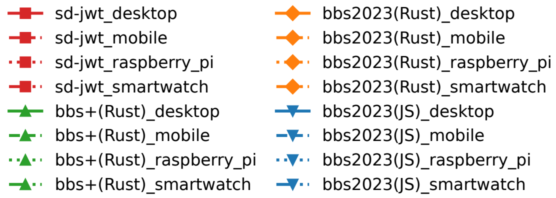

Figure 3: Annotation for multi-device plots.

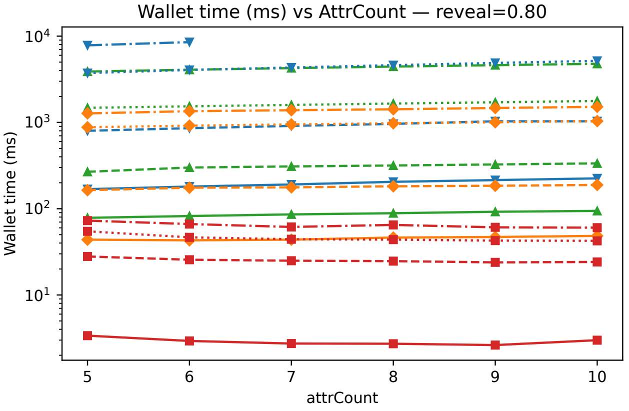

Figure 4: Wallet time vs. number of attributes at fixed reveal ratio. Curves show 4 implementations across 3 devices.

4.4.1 Wallet time. Figure 4 shows the time required for the prover (wallet) to generate the required attestations, requests and proof. We fix the reveal ratio to 80% and vary the attribute count from 5 to 10. Note that the y-axis is logarithmic. Detailed results for attribute counts of 5 and 10 are presented in Table 1. The BBS2023(JS) implementation occasionally crashed on the smartwatch for attribute counts greater than 6, especially when the runtime exceeded 10 seconds, and thus the prescribed 50 iterations could not be reliably performed for this experimental setting. Hence, results for these instances are not included in the plots and tables. Here, we can make the following observations regarding our stated research questions:

Implementation choices. The baseline (SD–JWT) incurs minimal overhead in all scenarios (< 75 ms). There is an order of magnitude difference (~10x) between the baseline and the most efficient BBS variant, BBS2023(Rust). The BBS+ variant is 2-3x slower than BBS2023. Differences are more pronounced on mobile devices.

Scalability. The effect of the number of attributes is generally small but not negligible, especially for resource-constrained devices. For example, the best-performing BBS variant required 1.3 seconds on

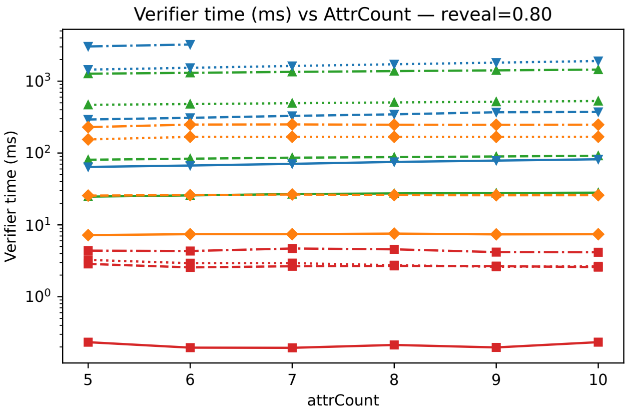

Figure 5: Verifier time vs. number of attributes at fixed reveal ratio. Curves show 4 implementations across 3 devices.

average to generate a proof for 5 attributes and 1.5 seconds for 10 attributes on the smartwatch.

Hardware capabilities. There are significant differences between the devices tested when considering unlinkable variants. For the best performing unlinkable scheme and 10 attributes, resource-constrained devices are an order of magnitude slower compared to the mobile phone and desktop, with the smartwatch requiring 1.5 seconds to generate the proof. Nevertheless, this figure can still be considered practical for everyday transactions on existing hardware. Differences on the baseline scheme are negligible.

Programming language. Comparing between the two implementations of BBS2023, the Javascipt one is around 5x slower than the Rust one. This agrees with observations by Pereira [20] who measured JavaScript to be around 6 times slower than Rust on a range of benchmarks. The JavaScript implementation incurred a runtime of more than 10 seconds on the smartwatch, suggesting that it is unsuitable for deployment on such platforms.

We note that, except for the presentation payload analysis in Section 4.4.3, results are reported for a fixed reveal ratio of 0.8. Experiments with varying reveal ratios showed minimal effect on execution time for both wallet and verifier operations. Consequently, we focus on a static reveal ratio across Tables 1 and 2 and the corresponding figures to better isolate the effect of attribute count and implementation differences. The full results and plots are available in our tool-repository.[12]

4.4.2 Verifier time. Figure 5 and Table 2 summarize the benchmark results for the verifier, similar to the previous section. Results generally align with the ones presented earlier for the prover, with the difference that verification is an order of magnitude lighter than the proof generation. Specifically, the best-performing unlinkable implementation on a smartwatch can verify claims of 10 attributes in a quarter of a second. Results suggest that verification of unlinkable credentials can be performed seamlessly on mobile hardware.

Table 1: Mean wallet (prover) times (ms) at reveal=0.80 and attributes 5,10. Rounded to nearest ms.

# Attr.	Impl.	Desktop	Mobile	RPiZ	Watch
5	BBS2023(JS)	169	800	3733	7830
	BBS2023(Rust)	44	164	879	1273
	BBS+(Rust)	78	268	1477	3890
	SD–JWT	3	28	55	73
10	BBS2023(JS)	225	1031	5173	–
	BBS2023(Rust)	48	189	1037	1516
	BBS+(Rust)	94	335	1773	4785
	SD–JWT	3	24	42	60

Table 2: Mean verifier times (ms) at reveal=0.80 and attributes 5,10. Rounded to nearest ms.

# Attr.	Impl.	Desktop	Mobile	RPiZ	Watch
5	BBS2023(JS)	64	293	1448	3043
	BBS2023(Rust)	7	26	155	229
	BBS+(Rust)	25	81	469	1270
	SD–JWT	< 1	3	3	4
10	BBS2023(JS)	82	373	1908	–
	BBS2023(Rust)	7	26	168	248
	BBS+(Rust)	28	92	529	1448
	SD–JWT	< 1	3	3	4

4.4.3 Presentation payload. Presentation payloads are deterministic and not affected by device type. Figure 6 shows the presentation payload for a fixed number of attributes while varying the reveal ratio. We observe that BBS-based presentations get smaller as the reveal ratio increases, while SD-JWT get larger. This follows directly from the mechanics of each scheme: for BBS/BBS+, the derived proof size is affected by the number of *unrevealed* messages, causing the overall payload to decrease even though more plaintext values are sent. SD-JWT, on the other hand, includes a separate disclosure object for each revealed claim, leading to a steady increase in payload as the reveal ratio grows.

Figure 7 then fixes the reveal ratio and varies the total number of attributes. In this configuration, the payload increases almost linearly for all schemes, since each additional attribute contributes new data and proof material. Among unlinkable variants, BBS2023(Rust) consistently produces the smallest payloads, followed by BBS2023(JS) and BBS+(Rust), while SD-JWT has the highest payload starting from ten attributes. We attribute the discrepancy in proof size between the two implementations of the BBS2023 scheme to implementation choices, such as the number of commitments and scalar responses included in the proof construction. A promising avenue for future work would be to experimentally test for such discrepancies between implementations of the same schemes to uncover bugs and standards deviations, as has been done, e.g. for TLS implementations [5, 9].

Overall, these results suggest that unlinkable schemes introduce a predictable and moderate communication overhead. In practice, developers can expect payload sizes to decrease with higher disclosure ratios for BBS-based schemes and to scale linearly with attribute count.

[12] https://github.com/Aglag257/Identity-Wallet-Flow-Benchmarking-Tool/blob/main/single_device_new/summary_aggregated.csv

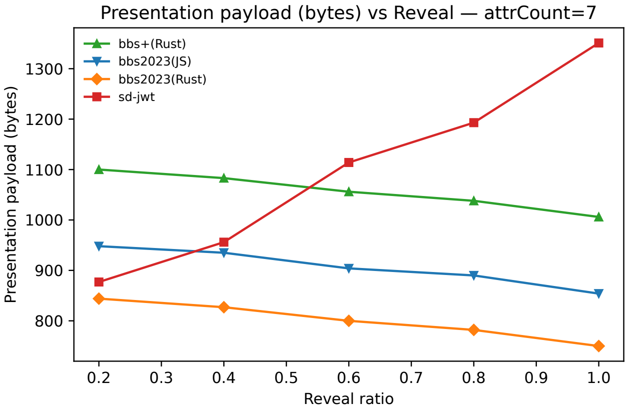

Figure 6: Presentation payload vs. reveal ratio (fixed attribute count).

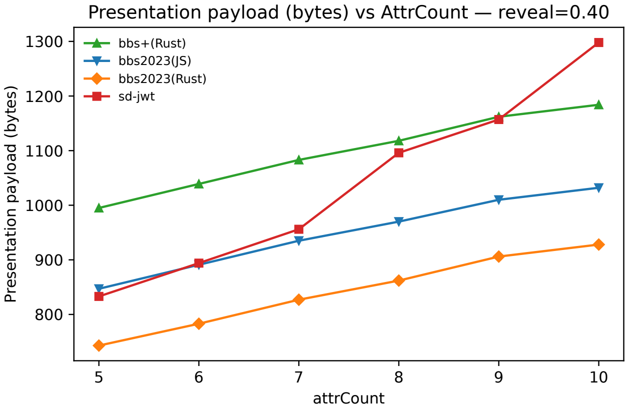

Figure 7: Presentation payload vs. attribute count (fixed reveal ratio).

5 Discussion

Our experimental findings confirm that unlinkable credential schemes can achieve practical performance on mobile hardware while maintaining strong privacy guarantees. The results demonstrate that the performance gap between linkable SD-JWTs and unlinkable BBS-based schemes, though still significant, is within tolerable limits for real-world deployment. The overhead observed on resource-constrained devices such as smartwatches and Raspberry Pi Zero 2W suggests that optimization of pairing-based cryptography remains a critical engineering task, but does not preclude usability for interactive identity proofs.

A key insight emerging from our measurements is that implementation language exerts an equal or even greater influence on latency than the underlying cryptographic primitive itself. For example, the Rust implementation of BBS2023 consistently outperformed its JavaScript counterpart by a factor of 4–5× across all devices. This highlights the need for native implementations when integrating unlinkable schemes into production wallets. From a systems perspective, our results support the feasibility of deploying unlinkable schemes within the European Digital Identity Wallet (EUDIW) ecosystem without compromising user experience with large latencies.

Finally, while the experiments were designed to reflect realistic end-to-end flows, various environmental factors can still influence performance in deployment. The following subsection discusses potential threats to validity and generalization, followed by conclusions on broader implications and future directions.

5.1 Threats to validity

Construct validity. Our benchmarks capture complete end-to-end credential flows, encompassing nonce generation, HTTP exchanges, token signing, and proof verification, rather than isolated cryptographic primitives. This design improves ecological validity for real developers by reflecting realistic wallet behavior. Although it inherently couples cryptographic and protocol overheads, attribution of latency to the cryptographic layer remains reliable because the compared implementations are nearly identical except for the signature schemes and their associated proof-generation logic.

Internal validity. All experiments were performed 50 times under controlled conditions, with background processes minimized to avoid external interference. Small variations may still arise from normal runtime behavior, but these do not materially affect the results. Each credential scheme was tested using a representative reference implementation. Although small optimizations or compiler differences could slightly change absolute values, we do not expect them to affect our conclusions.

Generalization. Our device selection covers representative hardware classes, desktop, smartphone, smartwatch, and low-cost single-board computer, but not all possible configurations. Future devices featuring hardware-backed secure elements, or alternative architectures may exhibit different scaling behavior. Similarly, our experiments focused on single-credential issuance and presentation; multi-credential or concurrent issuance/presentation could introduce additional overhead. Nevertheless, the relative performance trends observed across schemes and devices are expected to generalize to most modern wallet implementations. While we focus on latency and payload size, future work could measure additional dimensions such as energy consumption and memory overhead, especially on constrained devices."

Reproducibility. To mitigate these threats, all benchmarking code, and environment configurations have been released as open source,[13] enabling independent reproduction, comparison against new schemes, and validation across different runtime environments.

5.2 Conclusion

This paper presented a practical benchmarking study of complete verifiable credential flows executed on real hardware, representative of the devices envisioned for the European Digital Identity Wallet. By running experiments, on desktops, smartphones, smartwatches, and low-cost single-board computers, we quantified the real-world performance that implementers and policymakers can expect from candidate credential schemes.

Our results show that unlinkable credentials based on BBS signatures are feasible for everyday mobile use. The BBS 2023 Rust

[13] https://github.com/Aglag257/Identity-Wallet-Flow-Benchmarking-Tool/tree/main

implementation achieved presentation times under two seconds even on resource-constrained devices, demonstrating that privacy-preserving credential flows can remain responsive in real-world deployments. Differences across implementations and programming languages proved significant, underscoring the importance of native libraries for production-grade wallets. Beyond the reported measurements, our benchmarking suite can be used to benchmark other schemes and implementations, helping implementers quantify practical trade-offs and supporting evidence-based decisions within the European Digital Identity Wallet ecosystem.

Acknowledgments: This project received funding from the EU under the MSCA grant agreement No. 101108713.

References

[1] Man Ho Au, Willy Susilo, and Yi Mu. 2006. Constant-size dynamic k-taa. In *Security and Cryptography for Networks, 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006, Proceedings* (Lecture Notes in Computer Science). Roberto De Prisco and Moti Yung, (Eds.) Vol. 4116. Springer, 111–125. doi:10.1007/11832072_8.

[2] Carsten Baum et al. 2024. Cryptographers' Feedback on the EU Digital Identity's ARF. Tech. rep. Consensus feedback on the EU Digital Identity Wallet (EUDIW) ARF v1.4.0, June 2024. Independent group of cryptographers, (June 2024). Retrieved Sept. 10, 2025 from.

[3] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. 2014. Zerocash: decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014*. IEEE Computer Society, 459–474. doi:10.1109/SP.2014.36.

[4] Dan Boneh, Xavier Boyen, and Hovav Shacham. 2004. Short group signatures. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings* (Lecture Notes in Computer Science). Matthew K. Franklin, (Ed.) Vol. 3152. Springer, 41–55. doi:10.1007/978-3-540-28628-8_3.

[5] Chad Brubaker, Suman Jana, Baishakhi Ray, Sarfraz Khurshid, and Vitaly Shmatikov. 2014. Using frankencerts for automated adversarial testing of certificate validation in SSL/TLS implementations. In *2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014*. IEEE Computer Society, 114–129. doi:10.1109/SP.2014.15.

[6] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Gregory Maxwell. 2018. Bulletproofs: short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*. IEEE Computer Society, 315–334. doi:10.1109/SP.2018.00020.

[7] Jan Camenisch, Manu Drijvers, and Anja Lehmann. 2016. Anonymous attestation using the strong diffie hellman assumption revisited. In *Trust and Trustworthy Computing - 9th International Conference, TRUST 2016, Vienna, Austria, August 29-30, 2016, Proceedings* (Lecture Notes in Computer Science). Michael Franz and Panos Papadimitratos, (Eds.) Vol. 9824. Springer, 1–20. doi:10.1007/978-3-319-45572-3_1.

[8] Jan Camenisch and Anna Lysyanskaya. 2004. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings* (Lecture Notes in Computer Science). Matthew K. Franklin, (Ed.) Vol. 3152. Springer, 56–72. doi:10.1007/978-3-540-28628-8_4.

[9] Yuting Chen and Zhendong Su. 2015. Guided differential testing of certificate validation in SSL/TLS implementations. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2015, Bergamo, Italy, August 30 - September 4, 2015*. Elisabetta Di Nitto, Mark Harman, and Patrick Heymans, (Eds.) ACM, 793–804. doi:10.1145/2786805.2786835.

[10] Nicolas Desmoulins, Antoine Dumanois, Seyni Kane, and Jacques Traoré. 2025. Making bbs anonymous credentials eidas 2.0 compliant. *Cryptology ePrint Archive*.

[11] Daniel Fett, Kristina Yasuda, and Brian Campbell. 2025. Selective Disclosure for JWTs (SD-JWT). Internet-Draft draft-ietf-oauth-selective-disclosure-jwt-22. Work in Progress. Internet Engineering Task Force, (May 2025). 96 pp. https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt/22/.

[12] Andrea Flamini, Silvio Ranise, Giada Sciarretta, Mario Scuro, Amir Sharif, and Alessandro Tomasi. 2023. A first appraisal of cryptographic mechanisms for the selective disclosure of verifiable credentials. In *Proceedings of the 20th International Conference on Security and Cryptography, SECRYPT 2023, Rome, Italy, July 10-12, 2023*. Sabrina De Capitani di Vimercati and Pierangela Samarati, (Eds.) SCITEPRESS, 123–134. doi:10.5220/0012084000003555.

[13] Andrea Flamini, Giada Sciarretta, Mario Scuro, Amir Sharif, Alessandro Tomasi, and Silvio Ranise. 2024. On cryptographic mechanisms for the selective disclosure of verifiable credentials. *Journal of Information Security and Applications*, 83, 103789. doi:https://doi.org/10.1016/j.jisa.2024.103789.

[14] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. 1985. The knowledge complexity of interactive proof-systems (extended abstract). In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*. Robert Sedgewick, (Ed.) ACM, 291–304. doi:10.1145/22145.22178.

[15] 2018. Identity in a Digital World: A New Chapter in the Social Contract. Insight Report. Insight Report, World Economic Forum. World Economic Forum, Cologny/Geneva, Switzerland, (Sept. 2018). Retrieved Sept. 10, 2025 from https://www.weforum.org/reports/identity-in-a-digital-world-a-new-chapter-in-the-social-contract.

[16] Michael B. Jones, John Bradley, and Nat Sakimura. 2015. JSON Web Token (JWT). RFC 7519. (May 2015). doi:10.17487/RFC7519.

[17] Lukas Leuba. 2025. Privacy-preserving credentials with bbs: bbs signatures for enhanced privacy in the swiss e-id: a theoretical and practical analysis. Bachelor Thesis, Faculty of Science, Cryptology and Data Security Group, Institute of Computer Science. Supervisors: Prof. Christian Cachin and François-Xavier Wicht. Bern, Switzerland, (July 2025).

[18] T. Lodderstedt, K. Yasuda, T. Looker, and P. Bastian. 2025. OpenID for Verifiable Credential Issuance 1.0. Tech. rep. Final Specification. Authors affiliated with SPRIND, Mattr, and Bundesdruckerei. OpenID Foundation, OpenID Digital Credentials Protocols Workgroup, (Sept. 2025). https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html.

[19] T. Looker, V. Kalos, A. Whitehead, and M. Lodder. 2023. The BBS Signature Scheme. Tech. rep. Internet Research Task Force (IRTF). https://www.ietf.org/archive/id/draft-irtf-cfrg-bbs-signatures-03.html.

[20] Rui Pereira, Marco Couto, Francisco Ribeiro, Rui Rua, Jácome Cunha, João Paulo Fernandes, and João Saraiva. 2021. Ranking programming languages by energy efficiency. *Sci. Comput. Program.*, 205, 102609. doi:10.1016/J.SCICO.2021.102609.

[21] Jean-Jacques Quisquater et al. 1989. How to explain zero-knowledge protocols to your children. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings* (Lecture Notes in Computer Science). Gilles Brassard, (Ed.) Vol. 435. Springer, 628–631. doi:10.1007/0-387-34805-0_60.

[22] 2024. Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 April 2024 amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework. Official Journal of the European Union, L 2024/1183 (30 April 2024). Entry into force 20 days after publication; EU Regulation. (Apr. 2024). Retrieved Sept. 10, 2025 from https://eur-lex.europa.eu/eli/reg/2024/1183/oj.

[23] Manu Sporny, Dave Longley, and David Chadwick. 2022. Verifiable Credentials Data Model v1.1. W3C Recommendation. Ver. 1.1 – W3C Recommendation (3 March 2022). World Wide Web Consortium (W3C), (Mar. 2022). Retrieved Sept. 10, 2025 from https://www.w3.org/TR/vc-data-model/.

[24] Manu Sporny, Dave Longley, Markus Sabadello, Grant Davidson, Drummond Reed, and Daniel Lundkvist. 2025. Verifiable credentials data model v2.0. https://www.w3.org/TR/vc-data-model-2.0/. W3C Recommendation. (Feb. 2025). Retrieved Sept. 10, 2025 from.

[25] Manu Sporny, Dave Longley, Markus Sabadello, Drummond Reed, Orie Steele, and Christopher Allen. 2025. Decentralized Identifiers (DIDs) v1.1 Core architecture, data model, and representations. W3C Working Draft. Ver. 1.1. World Wide Web Consortium (W3C), (Sept. 2025). Retrieved Sept. 27, 2025 from https://www.w3.org/TR/did-1.1/.

[26] O. Terbu, T. Lodderstedt, K. Yasuda, D. Fett, and J. Heenan. 2025. OpenID for Verifiable Presentations 1.0. Tech. rep. Final Specification. Authors affiliated with MATTR, SPRIND, and Authlete. OpenID Foundation, OpenID Digital Credentials Protocols Workgroup, (July 2025). https://openid.net/specs/openid-4-verifiable-presentations-1_0.html.

[27] Oliver Terbu, Daniel Fett, and Brian Campbell. 2025. SD-JWT-based Verifiable Credentials (SD-JWT VC). Internet-Draft draft-ietf-oauth-sd-jwt-vc-11. Work in Progress. Internet Engineering Task Force, (Sept. 2025). 58 pp. https://datatracker.ietf.org/doc/draft-ietf-oauth-sd-jwt-vc/11/.

[28] Stefano Tessaro and Chenzhi Zhu. 2023. Revisiting BBS signatures. In *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V* (Lecture Notes in Computer Science). Carmit Hazay and Martijn Stam, (Eds.) Vol. 14008. Springer, 691–721. doi:10.1007/978-3-031-30589-4_24.

[29] World Wide Web Consortium (W3C). 2023. Data integrity bbs cryptosuites v1.0. https://w3c.github.io/vc-di-bbs/. W3C Working Draft / Candidate Recommendation. (2023).