

Device Binding for Anonymous Credentials on Legacy Phones

Sofia Celi*, Anja Lehmann†, Shai Levin‡, Alexandros Zacharakis†
*Brave / University of Bristol, UK
Email: cherenkov@riseup.net
†Hasso Plattner Institute, University of Potsdam, Germany
Email: {anja.lehmann, alexandros.zacharakis}@hpi.de
‡Chalmers University of Technology, University of Gothenburg, Sweden
Email: shai.levin@chalmers.se

***Abstract*—Digital identity systems are being deployed worldwide to enable secure, privacy-preserving authentication. In this regards, Europe's EUDI Wallet mandates selective attribute disclosure and unlinkable authentication, effectively requiring anonymous credentials. Yet current proposed solutions omit them because they lack *device binding*: tying a credential to a secure hardware element on the user's device to prevent cloning or sharing. Device binding encodes a device public key in the credential and, at presentation time, produces a fresh signature under the hardware-protected secret key—a proof-of-possession (PoP). While academic solutions for device binding exist, they require pairing-friendly curves, whereas consumer phones are restricted to ECDSA over P-256.**

In this work, we bridge this gap by enabling device binding of pairing-based credentials using the device's standard ECDSA signature as the PoP. To explore the efficiency–complexity trade-off of such solution, we present three constructions. Our most efficient solution generates unlinkable binding proofs of size $\approx 1.5\,\mathrm{kB}$ in under $\approx 400\,\mathrm{ms}$ using a simple arithmetic circuit, whereas the most conservative circuit-free approach matches this proving time at $\approx 122\,\mathrm{kB}$. All solutions follow a common blueprint expressed in the *reductions of knowledge* framework (Crypto 2023), which enables modular composition of zero-knowledge proofs and eases both security analysis and implementation. This framework has previously been mainly used in a theoretical context, and our work demonstrates that it is a powerful tool to design, analyze and implement advanced real-world systems.

1. Introduction

Anonymous credentials enable privacy-preserving user authentication: users hold digital credentials, which are attested attributes from a trusted issuer, and can present them to Relying Parties (RP) in a decentralized manner. The presentation only reveals the information minimally necessary, e.g., proving that one is over 16 based on a credential containing full personal data. Anonymous credentials are well-established in the academic community, and are seeing increasing real-world attention. BBS-based credentials are currently considered the most mature candidate and are undergoing standardisation in IETF [1], ISO [2] and ETSI [3]. They are also being considered for the EUDI, Europe's upcoming digital identity system [4].

Non-transferability via device-binding. A key challenge for user-centric credentials is non-transferability, which is amplified when users can authenticate anonymously: because credentials are digital data, they can be copied or shared without restriction. A common mitigation is to bind credentials to a cryptographic key stored in a secure hardware element on the user's device. In traditional credential systems, this is achieved by embedding the device's public key among the credential's attested attributes. At presentation time, the user must produce a fresh signature – a Proof of Possession (PoP) – under the corresponding private key, which never leaves the secure hardware.

The upcoming European Digital Identity (EUDI) framework requires such device binding to Secure Elements (SEs) to achieve the desired security guarantees [4]. At the same time, the eIDAS regulation mandates that the EUDI Wallet be privacy-preserving, requiring selective attribute disclosure and unlinkable authentication [5], capabilities that essentially call for anonymous credentials.

While device-binding mechanisms for general anonymous credentials exist [6], [7], [8] and have been integrated into secure hardware as part of the TPM 2.0 standard [9], the SE deployed in consumer smartphones do *not* expose the necessary cryptographic APIs to support them. In practice, these SEs can only produce standard ECDSA signatures and lack support for the pairing-friendly curves required by all native anonymous credential schemes. Crucially, this constraint is not confined to older devices: extending the cryptographic capabilities of secure hardware is a multi-year endeavour spanning standardisation, certification, and hardware deployment cycles, so even new and high-end phones today – and in the foreseeable future – remain limited to ECDSA.[1] This incompatibility between anonymous credentials and the non-transferability requirement was a key

1. We use the term *legacy* throughout for brevity, but it should be understood broadly: it refers to the ECDSA-only constraint imposed by current and near-future secure elements alike, not merely to old hardware.

reason why anonymous credentials are not used in the initial EUDI Wallet deployment, which instead relies on batch-issuance of single-use credentials as a workaround [10].

Anonymous credentials for legacy systems. To address legacy hardware compatibility, circuit-based solutions such as zkCred [11], Longfellow [12], Vega [13], and OpenAC [14] build anonymous credentials entirely on top of legacy ECDSA signatures. This provides full compliance with existing digital identity systems, but at a cost in complexity and efficiency: even after heavy optimisation, Longfellow takes $500\,\mathrm{ms}$ and produces $300\,\mathrm{kB}$ per presentation. While this may suffice for occasional online presentations, it falls short for high-frequency or proximity-based use cases – e.g. over BLE or NFC – where bandwidth is limited and interaction times must remain below perceptible latency thresholds.

Limits of monolithic circuits. Beyond performance, these solutions pose major standardisation and certification challenges. The EUDI framework requires interoperability across member states and wallet providers, so every component of the cryptographic stack – signature scheme, proof system, and proven circuit – must be specified in full detail. Yet standardisation of circuit-based proof systems has stalled: efforts within the zkProofs [15] community have run for over eight years without convergence, leaving this deployment path distant and uncertain.

Monolithic circuits are also rigid: extending a large, tightly integrated circuit with features such as pseudonyms, cross-credential proofs, or deniability requires redesigning, re-optimising, and re-certifying the entire construction. Their complexity is also a security liability – subtle circuit bugs have repeatedly led to critical vulnerabilities in deployed systems [16].

BBS with ECDSA device binding. A different approach was taken by Ubique [17] in a EUDI innovation competition: rather than forcing the entire credential scheme through an ECDSA circuit, they use a natively ZK-friendly signature – BBS [18] – for the credential and confine ECDSA to device binding. This separation reflects where the legacy constraints actually lie: they are rooted in user devices, whose SEs are already deployed and cannot be updated, whereas issuers can more readily adopt modern signature schemes. Confining ECDSA to device binding retains the full expressiveness and efficiency of BBS for the credential layer, yielding a simpler and naturally extensible design.

Since BBS and ECDSA operate over incompatible curves (BLS12-381 and P-256) and the PoP proves statements about P-256 elements, two challenges arise: bridging the two curves efficiently, and proving P-256 operations while keeping the device public key hidden. Ubique addresses the former via [19] and the latter via the Schnorr-based zkAttest/CDLS scheme [20], [21], making their construction considerably simpler and more conservative than full SNARKs over ECDSA. Their prototype presents a BBS credential with ECDSA device binding in $700\,\mathrm{ms}$ with proofs of around $200\,\mathrm{kB}$, where the BBS component contributes only a few milliseconds and negligible size – the cost is dominated by the ECDSA PoP.

Towards more efficient proofs of possession. While this hybrid BBS-ECDSA approach is a significant architectural improvement over fully monolithic constructions, its bridging layer remains a bottleneck – still impractical for proximity scenarios such as NFC tap-and-go authentication, where message sizes should ideally remain within a few kilobytes. Its Schnorr-based proofs are inherently linear in the size of the proven computation, leaving little room for further compression. This motivates the following question:

How can we realize ECDSA device-binding with a more efficient proof system?

To address this question, we build on the *modular credential framework* of [22]. There, a user holding a BBS-signed credential over several attributes can perform a *committed disclosure*: rather than revealing attributes in the clear, the user provides a Pedersen commitment on BLS12-381 to the relevant values and proves in zero knowledge that these values carry a valid issuer signature. Any further functionality – device binding – is then constructed generically by consuming this commitment and proving an additional statement over the committed values. Since the BBS layer contributes only negligible cost in both time and proof size, this clean separation allows us to focus entirely on optimising the device-binding module.

1.1. Our Contributions

In this work, we present three constructions that significantly improve the state of the art for pairing-based anonymous credentials on legacy, ECDSA-only devices. The first two rely on *lightweight* arithmetic circuits and differ in the trade-off they strike between proof size and protocol complexity. The third formalises and analyses the *purely Schnorr-type proof* approach of Ubique–which was proposed without a formal treatment–within our framework. We further optimise this underlying *Schnorr-type proof* of [21] at both protocol level and in its implementation. Together, this yields a unified account of all three instantiations

Modular design via reductions of knowledge. All constructions follow a common blueprint, that we express in the *reductions of knowledge* framework of Kothapalli and Parno [23]. This framework enables modular composition of zero-knowledge proofs, greatly facilitating both security analysis and implementation. Previously employed mainly in theoretical contexts, our work demonstrates its utility for designing and implementing real-world protocols.

The core technical challenge. For ECDSA device binding, we need to prove P-256 operations given commitments living in BLS12-381 – a setting with inherent efficiency limitations. We address this in several ways. First, for our

circuit-based approach, we either apply efficient *foreign-field arithmetic* using the techniques of [24] to emulate P-256 operations directly within a circuit over BLS12-381, or rely on a curve that forms a chain with P-256 [20], nearly eliminating the overhead of non-native computation in the circuit. Second, for our purely-Schnorr construction, we forgo circuits entirely: we express the ECDSA verification equation as a composition of dedicated protocols for scalar multiplication and point addition, run natively over the same chain-compatible curve. This avoids both foreign-field overhead and the cost of a general-purpose proof system, at the price of larger proofs that grow with the Fiat-Shamir soundness parameter. Both circuit-based constructions further reduce circuit size through a novel *Committed Schnorr* protocol that (1) halves the number of required curve operations (for knowledge soundness error $2^{-\lambda}$) and (2) reduces the problem from scalar multiplication with a *hidden* scalar to one with a *public* scalar, which is simpler to express as constraints and, in some cases, more efficient to prove.[2]

A family of constructions. Our techniques, combined with the Schnorr-type credential of [17], yield a family of constructions with trade-offs not only in efficiency but also in implementation complexity, standardisation effort, and acceptable cryptographic assumptions (for instance, whether to rely on the non-standard curve forming a chain with P-256, or on proving systems with more advanced features). For the purely Schnorr-type proof approach, we optimise the protocol of [21] (based on the insecure zkAttest scheme [20]) by removing unnecessary components of the proof without compromising security. For the circuit-based constructions, we chose proving systems that minimise proof size, as this is the main barrier for many use cases, notably proximity-based authentication. We believe that adapting our blueprint to also optimise prover time – for instance using a lightweight variant of [12] for the device-binding circuit – is an interesting direction for future work.

Implementation and results. We implement and benchmark all three constructions; the code is available in [25]. Our most efficient variant produces unlinkable device-binding proofs of approximately $1.5\,\mathrm{kB}$ in under $400\,\mathrm{ms}$, while the most conservative –avoiding circuits entirely– achieves faster proving time with a proof size of $122\,\mathrm{kB}$. Importantly, our circuits are far simpler than those of fully monolithic approaches: they encode *only the device-binding relation* and benefit from a much simpler ECDSA statement. Their scope is narrow, static, and independent of the credential scheme, which eases standardisation.

1.2. Our Techniques

We overview our core techniques: the statement to be proven for device binding, the RoK framework, and the concrete constructions – including a novel *Committed Schnorr* protocol, which we believe to be of independent interest.

1.2.1. ECDSA Device Binding. Device binding is realised by proving possession of the device's secret key during credential presentation. Concretely, the verifier sends a random nonce (n), and the user proves that they can produce a valid ECDSA signature on this nonce under the device's public key, which is attested in the user's credential. The secret key never leaves the SE: the user can only issue signature queries and cannot access the key material directly. The device's public key is the only long-lived value, is attested in the credential, and remains secret.

In the framework of [22], the process corresponds to verifiably disclosing *a commitment to the device public key* and then prove (in ZK) knowledge of a valid signature under this key. This process equates to verifying the ECDSA equation (see Section 2): proving knowledge of z, K s.t. $zK = \mathsf{H}(\mathsf{n}) \cdot F(K)^{-1} G_p + Q$, where Q is the device public key, (z, K) are (deterministically) derived from the ECDSA signature, n must be signed and F is the ECDSA conversion function.

Two observations simplify this statement compared to fully ECDSA-based credential schemes. First, the signed nonce is public, eliminating any need to prove statements about a hash function. Second, because the signature is freshly generated and used only once, we can reveal parts of it – in particular the value K – *without compromising unlinkability*. Hence, we can reduce it to proving knowledge of a secret scalar z such that

$$zK = \alpha G_p + Q \tag{1}$$

where α is computed from public values as $\alpha = \mathsf{H}(\mathsf{n}) \cdot F(K)^{-1}$. Note that Equation (1) is a computation over *the base field of* P-256 which we denote with $\mathbb{F}_p$.

Minimizing non-native operations. Equation (1) must be proven w.r.t. a committed device public key Q; in the case of BBS credentials, the commitment is a Pedersen commitment over BLS12-381. This creates an inherent tension: either the proving system's native field[3] is the scalar field of BLS12-381, which makes the P-256 arithmetic of the ECDSA equation non-native, or it is the base field of P-256, in which case the BLS12-381 commitment opening becomes non-native. We address this tension with two approaches, detailed in the constructions below: *foreign-field arithmetic*, which proves the statement directly over BLS12-381 by emulating the P-256 field operations, and *commitment linking*, which instead "transfers" the BLS12-381 commitments to a scheme whose message space is the base field $\mathbb{F}_p$ of P-256 – the only non-native part of the proof – so that all subsequent operations run natively over $\mathbb{F}_p$.

Committed Schnorr Protocol. Even with this minimiza-

2. In this case the circuit depends on the statement, which adds overhead. We emphasise that this requires no additional trust assumptions for the proving systems we use.

3. Native refers to the field used to arithmetise the statement. Computations over other fields must be emulated, which is significantly more costly.

tion, our circuit-based approaches can remain costly. However, we can extract the secret scalar from the circuit via a preliminary Schnorr-like interaction: the *Committed Schnorr* protocol (CSchnorr).

The idea of CSchnorr is to consider the simplified equation $zK = Z$, where Z is committed and z is secret. The prover first sends a commitment to $R \leftarrow rK$ for a random r, the verifier replies with a short challenge $c \leftarrow \{0,1\}^\lambda$, and the prover responds with $s := cz + r$. In a standard Schnorr proof, the verifier would conclude by checking that $sK = cZ + R$. In our setting, however, the verifier holds only commitments to Z and R, and therefore cannot perform this check directly. Instead, the prover supplies a proof that the verification equation holds over P-256.

Crucially, this transformed equation involves only *public* scalars: both s and c are known to the verifier. Proving a scalar multiplication with a public scalar in a circuit is significantly cheaper than with a secret one, as the double-and-add execution is fixed and not all "branchings" need to be executed. Moreover, since c is sampled from $\{0,1\}^\lambda$ rather than the full field $\mathbb{F}$, the multiplication cZ requires only λ doublings and additions instead of $\log|\mathbb{F}|$, reducing the number of constraints by roughly a factor of two. Hence, CSchnorr trades a single round of interaction for a substantially simpler circuit, which can be safely applied to both our circuit-based constructions.

Our Committed Schnorr protocol is structurally similar to the Sigmabus protocol of [19] – both commit to the statement and the Schnorr first message – but serves a different goal. In their work, the protocol efficiently "witnesses" in-circuit discrete logarithms of public group elements. The committed Schnorr protocol has a different scope: it allows *transforming curve equation statements* in a form that is more efficient to prove in-circuit. For witnessing discrete logarithms we don't rely on the techniques of Sigmabus, but rather use proving systems that allow having (Pedersen-type) commitments as inputs; the corresponding scalar elements are then witnessed for free.

Reductions of knowledge (RoK). To construct and analyse the PoP, we rely on the reductions of knowledge framework [23], [26]. This framework allows reducing the validity of a statement about some relation $\mathcal{R}_1$ to the validity of a statement about a different relation $\mathcal{R}_2$. E.g., the CSchnorr protocol can be viewed as a reduction from knowledge of z and commitment opening Z satisfying $zK = Z$, to knowledge of commitment openings Z, R satisfying $sK = cZ + R$. The final proofs are then expressed as a chain of reductions $\mathcal{R}_1 \to \mathcal{R}_2 \to \cdots \to \mathcal{R}_k$. It is enough to analyse each atomic reduction; the properties of the composed protocol are guaranteed by the framework itself.

1.2.2. Instantiating PoPs. As noted, we propose three approaches to prove ECDSA device binding (Equation (1)): the starting point for all is a Pedersen commitment over BLS12-381 to the device public key Q.

Circuit-based (Π_{Circ}): Our two circuit-based constructions are instantiations of a single generic protocol Π_{Circ}, designed and analysed in the RoK framework. The protocol composes three reductions: (1) the CSchnorr protocol reduces the equation to one involving only public scalars; (2) a commitment transfer re-expresses the commitments to one compatible with the system ($\widetilde{\mathsf{GCS}}$); and (3) a circuit-based proof proves the resulting relation over P-256:

$$\Pi_{\mathsf{Circ}} = \Pi_{\mathcal{C}} \circ \Pi_{\mathsf{GCS}\to\widetilde{\mathsf{GCS}}} \circ \Pi_{\mathsf{CSchnorr}}$$

where $\Pi_2 \circ \Pi_1$ denotes sequential composition, i.e. running Π_2 on the statement/witness pair produced after running Π_1 on the initial pair. The construction is generic in the choice of commitment scheme and proving system, and we propose two instantiations:

(a) *Native instantiation (*PoP-BP*):* The commitment transfer moves the statement to the Tom curve T-256 [20], whose scalar field coincides with $\mathbb{F}_p$, making all P-256 operations native. Here GCS is Pedersen over BLS12-381 and $\widetilde{\mathsf{GCS}}$ is compact Pedersen[4] over T-256. This eliminates the foreign-field overhead entirely but introduces a discrete logarithm assumption over T-256. We instantiate the circuit part with Bulletproofs [27] and rely on [19] for the cross-group commitment transfer. This yields our construction that optimizes for proof size.

(b) *Foreign-field instantiation (*PoP-PLONK*):* The commitment transfer stays within BLS12-381 – with GCS as Pedersen and $\widetilde{\mathsf{GCS}}$ as KZG [28], introducing no new assumptions. Since the circuit operates over BLS12-381 but must verify P-256 arithmetic, the P-256 operations are non-native and must be emulated. We instantiate Π_{Circ} with a variant of Plonk [29], employing the foreign-field techniques of [24] to emulate P-256 arithmetic efficiently.

Schnorr-based (PoP-Σ): Our third construction formalises the Schnorr-based approach of [17], [22], [30] and avoids circuits entirely. It decomposes the ECDSA equation into a scalar multiplication (Π_{SM}) and a point addition (Π_{PA}), each proven with a dedicated Schnorr-type protocol. As in the circuit-based instantiations, commitments are first transferred from BLS12-381 to T-256, and the two proofs then run in parallel:

$$\text{PoP-}\Sigma = (\Pi_{\mathsf{SM,T\text{-}256}} \times \Pi_{\mathsf{PA,T\text{-}256}}) \circ \Pi_{\mathsf{BLS12\text{-}381}\to\mathsf{T\text{-}256}}$$

We present optimised variants of the protocols $\Pi_{\mathsf{SM,T\text{-}256}}$ and $\Pi_{\mathsf{PA,T\text{-}256}}$ which improve existing approaches performance (both in timings and size) by approximately 25-30%. This is the most conservative of our three approaches, as it requires no circuits and relies only on standard Schnorr-type techniques, at the cost of significantly larger proof sizes.

On the choice of the proving system. We focus on proving

4. Compact Pedersen commitment refers to the variant of Pedersen commitment scheme that allows committing to vector of field elements into a single group element.

systems that *minimize proof size*, which is essential for network-constrained applications such as tap-and-go presentations and, to our knowledge, has not been targeted before. Given the inherent tension between proof size and prover time, our constructions match or slightly exceed the proving time of monolithic approaches; thanks to our simpler and shorter circuits, we expect more prover-efficient proving systems to improve all metrics, and leave this to future work.

2. Preliminaries

Notation. We denote with λ the security parameter. We use $x \leftarrow S$ to denote sampling an element from S uniformly at random and assigning it to x. For a probabilistic algorithm $\mathcal{A}$, we denote with $x := \mathcal{A}(\,\cdot\,; r)$ the process of executing $\mathcal{A}$ on some input and randomness r and assign the result to x. We denote $x \leftarrow \mathcal{A}(\cdot)$ the process $r \leftarrow \mathcal{D}; x := \mathcal{A}(\,\cdot\,; r)$ where $\mathcal{D}$ is the uniform distribution over the space of randomness of $\mathcal{A}$. We will use various groups based on Elliptic Curves (EC). We associate each elliptic curve with two fields: the base field, where the elliptic curve point coordinates live, and the scalar field, where scalars lives. In this work we consider the P-256 curve [31]. We denote the corresponding group with $\mathbb{G}_p$, its generator with G_p and its base field with $\mathbb{F}_p$.

ECDSA Signature Scheme. ECDSA is a widely used digital signature scheme defined over ECs. Let $\mathbb{G}$ be a group of order q with generator $G \in \mathbb{G}$ where $\mathbb{G}$ is defined over some EC with base field $\mathbb{F}_p$. A key pair of ECDSA consists of $(x, Q) \in (\mathbb{F}_q, \mathbb{G})$ where x is sampled uniformly at random and $Q = xG$. Let $\mathsf{H} : \{0,1\}^* \to \mathbb{F}_q$ be a hash function and $F : \mathbb{G} \to \mathbb{F}_q$ be a function mapping curve points to scalar field elements. To sign a message m, the signer outputs the signature (r, s) where

$$k \leftarrow \mathbb{F}_q, K := kG,\ r := F(K),\ s := k^{-1}\big(\mathsf{H}(m) + rx\big)$$

To verify the signature, one asserts that $F(K) \stackrel{?}{=} r$ where $K := \mathsf{H}(m)s^{-1}G + rs^{-1}Q$. As shown in [20], one can deterministically transform a signature (r, s) to a pair (K, z), where K is as above and $z := r^{-1}s$. The corresponding verification procedure accepts iff (r, s) is valid; in particular, checking $zK \stackrel{?}{=} \mathsf{H}(m)r^{-1}G + Q$.

Commitment Schemes. A perfectly hiding commitment scheme with canonical opening $\mathsf{CS} = (\mathsf{Setup}, \mathsf{Com})$ consists of two PPT algorithms: $\mathsf{ck} \leftarrow \mathsf{CS.Setup}(pp)$, which on input parameters $pp \leftarrow \mathsf{Gen}(1^\lambda)$[5] outputs a commitment key ck; and $C := \mathsf{CS.Com}(\mathsf{ck}, m; \rho)$, which on input ck, a message $m \in \mathcal{M}$, and randomness ρ outputs a commitment C. Verification is canonical: to open C to m with randomness ρ, the verifier asserts $C \stackrel{?}{=} \mathsf{CS.Com}(\mathsf{ck}, m; \rho)$.

5. We assume all parameters – groups, generators, etc. – are sampled by a common generation algorithm Gen, allowing the same parameters to be shared across primitives.

The security guarantees are *binding*: no PPT adversary $\mathcal{A}(\mathsf{ck})$ can find two valid openings $(m_1, \rho_1), (m_2, \rho_2)$ with $m_1 \neq m_2$ and $\mathsf{CS.Com}(\mathsf{ck}, m_1; \rho_1) = \mathsf{CS.Com}(\mathsf{ck}, m_2; \rho_2)$, except with negligible probability; and *perfectly hiding*: for all $\mathsf{ck} \leftarrow \mathsf{CS.Setup}(pp)$ and $m_1, m_2 \in \mathcal{M}$, their commitments are identically distributed.

Committing to elliptic curve points. Our aim is to prove statements about ECDSA over the P-256 curve. Due to privacy, device public keys need to be committed to preserve user's identity. Such keys are EC points $Q \in \mathbb{G}_p \subseteq \mathbb{F}_p^2$. We will refer to (generic) commitment schemes with message space $\mathbb{G}_p$ in this work. In our instantiations, all our commitment schemes will be variants of the Pedersen commitment scheme whose message space is the scalar field $\mathbb{F}$ of some group $\mathbb{G}$ (or more generally $\mathbb{F}^k$ for some $k \in \mathbb{N}$). Therefore, to commit to points of P-256, we must *encode* $\mathbb{G}_p$ points as vectors over $\mathbb{F}$. When necessary, we will explicitly mention the message space as a vector of $\mathbb{F}$ elements. Finally, the commitment scheme is *native* for P-256 computations if its message space is $\mathbb{F}_p^k$: such commitment scheme can be instantiated with T-256 curve [20].

2.1. Reductions of Knowledge

We analyze and implement our constructions using the *reductions of knowledge* [23], [26] framework. We describe it informally here and defer the formal definitions to Appendix A. A reduction of knowledge (RoK) $\Pi : \mathcal{R} \to \mathcal{R}'$ is an interactive protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ that reduces knowledge of w s.t. $(\mathsf{x}, \mathsf{w}) \in \mathcal{R}$ to knowledge of w' s.t. $(\mathsf{x}', \mathsf{w}') \in \mathcal{R}'$, where x', w' are the respective outputs of $\mathcal{P}$ and $\mathcal{V}$ after their interaction. Informally, the protocol must satisfy:

Completeness: for all $(\mathsf{x}, \mathsf{w}) \in \mathcal{R}$, an honest interaction between $\mathcal{P}$ and $\mathcal{V}$ yields $(\mathsf{x}', \mathsf{w}') \in \mathcal{R}'$.

Knowledge Soundness: if a malicious $\mathcal{P}^*$ knows a valid witness w' for the reduced statement x', it also knows a valid witness w for x. This is formalised by an extractor that *also takes the witness for the resulting statement* and outputs the witness w for the initial statement.

Public Reducibility: the new statement can be efficiently computed from the transcript of the interaction.

Honest Verifier Zero Knowledge: the transcript of an interaction between $\mathcal{P}$ and honest $\mathcal{V}$ is perfectly simulatable.

If a reduction is also public-coin (the verifier's messages are uniformly distributed over some space and independent of the messages sent by the prover) it can be turned to non-interactive using the Fiat-Shamir transform [32].

Roks are more expressive than Proofs of Knowledge (PoK) in the sense that the former can capture the latter. We can view a PoK as a RoK $\Pi : \mathcal{R} \to \mathcal{R}_\top$ where $\mathcal{R}_\top$ is the trivial relation that only accepts $\mathsf{x} = \mathsf{true}$ (capturing that PoK the verifier accepts) and reject all other values (capturing that the PoK the verifier rejects). We use the notation $\Pi : \Pi_\mathcal{R} \to \Pi_\top$ to express PoKs.

As shown in [23], [26], RoKs:

1) *compose sequentially*: when $\mathcal{P}$ and $\mathcal{V}$ engage in an execution $\Pi_1 : \mathcal{R}_1 \to \mathcal{R}_2$ and then an execution of $\Pi_2 : \mathcal{R}_2 \to \mathcal{R}_3$, the combined protocol, denoted $\Pi_2 \circ \Pi_1$, is a RoK from $\mathcal{R}_1$ to $\mathcal{R}_3$,
2) *compose in parallel*: when $\mathcal{P}$ and $\mathcal{V}$ engage in two parallel executions of $\Pi_1 : \mathcal{R}_1 \to \mathcal{R}_2$ and $\Pi_2 : \mathcal{R}_3 \to \mathcal{R}_4$ the combined protocol, denoted $\Pi_1 \times \Pi_2$, is a RoK from $\mathcal{R}_1 \times \mathcal{R}_3 \to \mathcal{R}_2 \times \mathcal{R}_4$, where

$$\mathcal{R} \times \mathcal{R}' = \big\{(\mathsf{x},\mathsf{x}') \mid \exists(\mathsf{w},\mathsf{w}') \text{ s.t. } (\mathsf{x},\mathsf{w}) \in \mathcal{R} \text{ and } (\mathsf{x}',\mathsf{w}') \in \mathcal{R}'\big\}$$

As seen in [23], many existing constructions can be expressed in this framework simplifying analysis.

Remark 1. We consider a universal Gen algorithm that outputs (the same) public parameters for all relations that are composed sequentially and in parallel. When we describe the relations, we abuse notation and only consider the relevant parts of these universal public parameters.

2.2. Relations

2.2.1. Committed Relations $\mathcal{R}_{\mathsf{CS},\mathsf{P}}$. We consider relations that capture that some predicate P holds where some of its inputs are committed. Concretely, let CS be a commitment scheme and P be an arbitrary predicate. We define:

$$\mathcal{R}_{\mathsf{CS},\mathsf{P}} = \left\{ \begin{matrix} pp := \mathsf{ck} \\ \mathsf{x} := (\vec{C}, \vec{x}) \\ \mathsf{w} := (\vec{\rho}, \vec{m}, \vec{w}) \end{matrix} \;\middle|\; \begin{matrix} C_i = \mathsf{CS.Com}_{\mathsf{ck}}(m_i;\rho_i) \wedge \\ \mathsf{P}(\vec{x}, \vec{m}, \vec{w}) \end{matrix} \right\}$$

Note that the above relation "splits" the witness into two parts: some elements are hidden and some are committed. Where needed we express relations as committed relations.

2.2.2. Equality of Committed Values $\mathcal{R}_{\mathsf{EQ}}$. We next show how to construct a RoK to "transfer" commitments from one commitment scheme to another. This can be beneficial in scenarios where the latter commitment scheme is more efficient for proving some computation but we need to start from the former. In our constructions, for example, we start by proving computation about a $\mathsf{P}\text{-}256$ equation with points committed using Pedersen commitment over a pairing curve.

Given a predicate P and two commitment schemes $\mathsf{CS}, \widetilde{\mathsf{CS}}$, the goal is to construct a RoK $\Pi_{\mathsf{CS}\to\widetilde{\mathsf{CS}}} : \mathcal{R}_{\mathsf{CS},\mathsf{P}} \to \mathcal{R}_{\widetilde{\mathsf{CS}},\mathsf{P}}$. We construct the RoK by relying on a (public coin) PoK for the relation

$$\mathcal{R}_{\mathsf{EQ}} = \left\{ \begin{matrix} pp := (\mathsf{ck}, \widetilde{\mathsf{ck}}) \\ \mathsf{x} := (C, \widetilde{C}) \\ \mathsf{w} := (\rho, \widetilde{\rho}, m) \end{matrix} \;\middle|\; \begin{matrix} C = \mathsf{CS.Com}_{\mathsf{ck}}(m;\rho) \wedge \\ \widetilde{C} = \widetilde{\mathsf{CS}}.\mathsf{Com}_{\widetilde{\mathsf{ck}}}(m;\widetilde{\rho}) \end{matrix} \right\}$$

Let $\Pi_{\mathsf{EQ}} : \mathcal{R}_{\mathsf{EQ}} \to \mathcal{R}_\top$ be a PoK. The RoK $\Pi_{\mathsf{CS}\to\widetilde{\mathsf{CS}}}$ consists of $\mathcal{P}$ sending fresh commitments under $\widetilde{\mathsf{CS}}$ for all the committed values, and $\mathcal{P}$ and $\mathcal{V}$ engage in parallel executions of Π_{EQ}. We describe this in Appendix B.

2.2.3. RoKs for Curve Operations $\mathcal{R}_{\mathsf{PA}}, \mathcal{R}_{\mathsf{SM}}$. We rely on RoKs that capture the validity of EC operations, namely point addition ($\mathcal{R}_{\mathsf{PA}}$) and scalar multiplication ($\mathcal{R}_{\mathsf{SM}}$), where some of the points are committed. Concretely, let $\mathbb{G}$ be a group defined over some EC and let GCS be a commitment scheme with message space $\mathbb{G}$. Define the following committed relations:

$$\mathcal{R}_{\mathsf{PA}} = \left\{ \begin{matrix} pp = \mathsf{ck} \\ \mathsf{x} = (C_i)_{i=1}^3 \\ \mathsf{w} = (G_i, \rho_i)_{i=1}^3 \end{matrix} \;\middle|\; \begin{matrix} C_i = \mathsf{GCS.Com}_{\mathsf{ck}}(G_i;\rho_i) \\ \wedge\, G_1 + G_2 = G_3 \end{matrix} \right\}$$

$$\mathcal{R}_{\mathsf{SM}} = \left\{ \begin{matrix} pp = \mathsf{ck} \\ \mathsf{x} = (C, G) \\ \mathsf{w} = (H, \rho, x) \end{matrix} \;\middle|\; \begin{matrix} C = \mathsf{GCS.Com}_{\mathsf{ck}}(G;\rho) \wedge \\ H = xG \end{matrix} \right\} \tag{2}$$

We will use public-coin HVZK proofs of knowledge $\Pi_{\mathsf{PA}} : \mathcal{R}_{\mathsf{PA}} \to \mathcal{R}_\top$ and $\Pi_{\mathsf{SM}} : \mathcal{R}_{\mathsf{SM}} \to \mathcal{R}_\top$.

3. Committed Schnorr Protocol (CSchnorr)

As noted, arithmetizing scalar multiplications to be proven with circuit-based techniques is computationally demanding. Roughly, one has to perform a variant of the double-and-add algorithm which requires $\mathcal{O}(|\mathbb{F}|)$ constraints. Further, both branches of the double-and-add must be computed *independently of the proven statement*. In this section, we propose a technique to alleviate the aforementioned complexity: the Committed Schnorr Protocol.

The core idea of our new approach is as follows: instead of witnessing the (secret) scalar, we first perform an execution of a variant of the Schnorr protocol outside the circuit, where $\mathcal{P}$ sends the first message committed. Then, we arithmetize the verification equation of the Schnorr protocol (over committed values). This allows to (1) reduce the size of the scalar involved in the circuit proof and (2) make the scalar part of the statement, which in turn means we don't need to arithmetize all branches of double-and-add in circuit[6].

The (Plain) Schnorr Protocol. We first recall the *plain* Schnorr protocol for proving knowledge of discrete logarithm. Let $\mathbb{G}$ be a group of order q. The Schnorr protocol allows $\mathcal{P}$ to convince $\mathcal{V}$ (in zero knowledge) that it knows the discrete logarithm of an element $Q = zK$ w.r.t. some element $K \in \mathbb{G}$.

$$\mathcal{P}\colon r \leftarrow \mathbb{F}_q,\ R := rK; \quad \mathcal{V}\colon c \leftarrow \mathbb{F}_q; \quad \mathcal{P}\colon s := r + c \cdot z. \tag{3}$$

$\mathcal{V}$ accepts if $sK = R + cQ$.

Hiding the value Q. Now, consider the case where one would like to keep Q secret: this value can be an *attested*

6. Note that this means we produce a new circuit for each proven statement. In our case this is acceptable since $\mathcal{V}$'s timer is not critical and the produced circuits are small. We emphasize that in our constructions, the circuits can be deterministically derived from a universal structured reference string (srs) and require no trust assumption apart from sampling (once) the srs.

group element derived from some credential presentation. The credential holder might need to prove knowledge of the corresponding discrete logarithm value, but not reveal the element: if the element is a long-lived value, revealing it will break the unlinkability of credential presentation. So, roughly, we aim in proving knowledge of z s.t. $Q = zK$ where Q is committed. We consider a slight generalization of the relation, namely, $Q = H + zK$, where H is a public group element. Let GCS be a (perfectly hiding) commitment scheme with message space $\mathcal{M} = \mathbb{G}$. We define the relation $\mathcal{R}_{\mathsf{C\text{-}DLOG}} =$

$$\left\{ \begin{matrix} pp := \mathsf{ck} \\ \mathsf{x} := \big(C_Q, H, K\big) \\ \mathsf{w} := \big(\rho_Q, Q, z\big) \end{matrix} \;\middle|\; \begin{matrix} C_Q = \mathsf{GCS.Com}_{\mathsf{ck}}(Q; \rho_Q) \wedge \\ zK = H + Q \end{matrix} \right\}$$

Compare this with the (modified) ECDSA verification equation presented in Section 2 ($zK \stackrel{?}{=} \mathsf{H}(m)F(K)^{-1}G + Q$) and note that by setting $H = \mathsf{H}(m)F(K)^{-1}G$, the relation captures knowledge of an ECDSA signature (K, z) on (public) message m under the (committed) public key Q.

CSchnorr Protocol. Let GCS be a commitment scheme with message space $\mathbb{G}$ and consider the Schnorr protocol where instead of sending group elements, we send commitments to them using GCS. $\mathcal{P}$ has input $\mathsf{w} := (\rho_Q, Q, z)$ and interacts with a $\mathcal{V}$ that has input $\mathsf{x} := C_q, H, K$ as follows:

$$\begin{aligned} &r \leftarrow \mathbb{F}_q, \quad R \leftarrow rK, \quad C_R := \mathsf{GCS}_{\mathsf{ck}}.\mathsf{Com}(R; \rho_R), \\ &c \leftarrow \mathbb{F}_q, \\ &s := r + c \cdot z. \end{aligned} \tag{4}$$

The corresponding verification equation in the Schnorr protocol (without committed values) would be

$$sK = (r + c \cdot z)K = R + c(H + Q)$$

or equivalently $T = R + cQ$ where $T = sK - cH$.

$\mathcal{V}$, however, can no longer perform the verification check since it does not know the openings of the commitments. If it was convinced, however, that the openings of C_Q, C_R satisfy $T = R + cQ$ (for example, ignoring zero knowledge for the time being, $\mathcal{P}$ could send directly these openings), it would be convinced that $\mathcal{P}$ knows z s.t. $zK = H + Q$. Put differently, if we define the relation $\mathcal{R}_{\mathsf{CSchnorr}} =$

$$\left\{ \begin{matrix} pp := \mathsf{ck} \\ \mathsf{x} := (C_Q, C_R, T, c), \\ \mathsf{w} := (Q, \rho_Q, R, \rho_R) \end{matrix} \;\middle|\; \begin{matrix} C_Q = \mathsf{GCS.Com}_{\mathsf{ck}}(Q; \rho_Q) \wedge \\ C_R = \mathsf{GCS.Com}_{\mathsf{ck}}(R; \rho_R) \wedge \\ T = R + cQ \end{matrix} \right\}$$

then the protocol defined above is a RoK $\Pi_{\mathsf{CSchnorr}} : \mathcal{R}_{\mathsf{C\text{-}DLOG}} \to \mathcal{R}_{\mathsf{CSchnorr}}$.

There are two important differences between the two relations $\mathcal{R}_{\mathsf{C\text{-}DLOG}}$ and $\mathcal{R}_{\mathsf{CSchnorr}}$ that we will exploit to improve efficiency in some of our constructions:

1) While both relations involve verifying scalar multiplications over $\mathbb{G}$ with committed elements ($zK = H + Q$ and $T = R + cQ$ respectively), the former involves *a secret scalar* while the latter involves *a public one*.
2) The Schnorr protocol has soundness error $1/|\mathbb{F}_q|$ in the general case. However, when $|\mathbb{F}_q|$ is "too large", one can use a smaller challenge space, e.g. $\{0,1\}^\lambda$, to fine tune the knowledge soundness error. Therefore, in the reduced statement, the prover does not need to do a full scalar multiplication.

We formally present the RoK in Figure 1 and give a theorem statement and proof showing that the above protocol (when instantiated with a challenge space $\{0,1\}^\lambda$) is indeed a RoK.

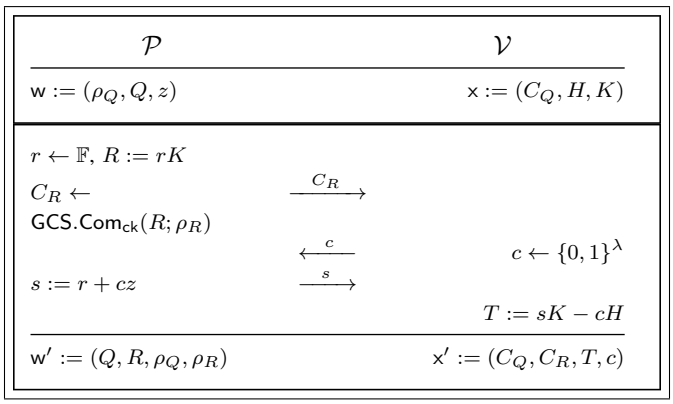


Figure 1: $\Pi_{\mathsf{CSchnorr}} \colon \mathcal{R}_{\mathsf{C\text{-}DLOG}} \to \mathcal{R}_{\mathsf{CSchnorr}}$.

Theorem 1. *Let* GCS *be a perfectly hiding commitment scheme with message space* $\mathbb{G}$*. Then, the protocol* Π_{CSchnorr} *described in Figure 1 is an honest-verifier public-coin RoK* $\Pi_{\mathsf{CSchnorr}} : \mathcal{R}_{\mathsf{C\text{-}DLOG}} \to \mathcal{R}_{\mathsf{CSchnorr}}$*. Furthermore, when* $\mathcal{V}$*'s message is sampled from* $\{0,1\}^\lambda$*, it has knowledge soundness error* $2^{-\lambda} + 2\epsilon_{\mathsf{GCS}}$ *where* ϵ_{GCS} *is the advantage of an adversary against the binding property of* GCS*.*

Proof.

Completeness and public reducibility. Assume $\mathsf{x}, \mathsf{w} \in \mathcal{R}_{\mathsf{C\text{-}DLOG}}$ and consider an honest execution of the protocol. Since $s = r + cz$, we have that

$$sK = (r + cz)K = rK + czK = R + c(H + Q)$$

where the latter equality holds since $zK = H + Q$. Noting that $T = sK - cH$, we conclude that $sK - cH = T = R + cQ$. Finally, since the prover is honest, the values (Q, ρ_Q) and (R, ρ_R) are valid openings of the commitments C_Q, C_R respectively. Since for these values it also holds that $T = R + cQ$, we conclude that the reduced statement/witness pair indeed satisfies $\mathsf{x}', \mathsf{w}' \in \mathcal{R}_{\mathsf{CSchnorr}}$. For public reducibility, it is enough to note that C_Q, C_R, c are parts of the transcript and that T can be constructed by the transcript.

Knowledge Soundness. We next show that the protocol is knowledge sound. We essentially rely on special-soundness of the Schnorr protocol and Lemma 1. Consider two executions with transcripts (C_R, c_b, s_b) and let $\mathsf{w}_b = (Q_b, \rho_{Q,b}, R_b, \rho_{R,b})$ be a valid witness for the reduced statement of each. First, note that by the binding property of the commitment scheme,

$Q_0 = Q_1(= Q)$ and $R_0 = R_1(= R)$ (except with some probability ϵ_H). Since both $\mathsf{w}_0, \mathsf{w}_1$ are valid witnesses, we have that $T_b = s_b K = R + c_b Q$ and therefore $(s_1 - s_0)(c_1 - c_0)^{-1} K = Q$, unless $c_0 = c_1$ which only happens with probability $2^{-\lambda}$. Therefore, the witness $\mathsf{w} := (Q, \rho_Q, (s_1 - s_0)(c_1 - c_0)^{-1})$ is a valid witness for the statement $\mathsf{x} = (C_Q, H, K)$, namely $(\mathsf{x}, \mathsf{w}) \in \mathcal{R}_{\mathsf{C\text{-}DLOG}}$.

Honest Verifier Zero Knowledge. A simulator can sample a commitment to the generator $C_R \leftarrow \mathsf{CS.Com}(G)$, a challenge $c \leftarrow \{0,1\}^\lambda$ and a response $s \leftarrow \mathbb{F}_q$. These values are identically distributed to a transcript derived from an honest execution. Indeed, this is immediate for c, s; for C_R it is enough to note that this is identically distributed to a commitment to $sK - cH$ due to the hiding property of the commitment scheme. □

4. ECDSA Proofs-of-Possession

In this section, we describe our ECDSA based device binding constructions. We start by describing the common parts shared by both the circuit-based and the Schnorr-type proof approaches; in particular we describe the concrete relation that needs to be proven and how we commit to P-256 curve points in a way that is compatible with the framework of [22]. Then, in Section 4.1 we present the blueprint of the circuit-based constructions and our two instantiations and in Section 4.2 we present our Schnorr-type proof approach.

The proven relation. The starting point for all our constructions is the relation $\mathcal{R}_{\mathsf{pECDSA}}$ that captures knowledge of a (partial) ECDSA signature on a public nonce n under a committed device public key. Concretely, let GCS be a commitment scheme with message space elements of P-256. We then define $\mathcal{R}_{\mathsf{pECDSA}} =$

$$\left\{ \begin{matrix} pp := \mathsf{ck} \\ \mathsf{x} := \left(C_Q, \mathsf{n}, K\right) \\ \mathsf{w} := \left(\rho_Q, Q, z\right) \end{matrix} \middle| \begin{matrix} C_Q = \mathsf{GCS.Com}_{\mathsf{ck}}(Q; \rho_Q) \wedge \\ zK = \alpha G_p + Q \end{matrix} \right\}$$

where $\alpha = \mathsf{H}(\mathsf{n}) F(K)$. Our goal is to create PoKs for the above relation. Note that part of the signature (the value K) is part of the statement. Assuming an honest Secure Element [8], this is enough to guarantee proof-of-possession of the device [22], [33].

Committing to P-256 points. Our core constructions Π_{Circ} (Figure 2) and Π_Σ (Figure 4) are agnostic to the instantiation of GCS. In practice, however, the choice of the commitment matters and determines concrete efficiency. We next explain how we instantiate the latter in all our constructions, which also determines the exact shape of $\mathcal{R}_{\mathsf{pECDSA}}$. Our goal is to use ECDSA-based device binding with pairing based credentials over some pairing group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$, all having scalar field $\mathbb{F}$, e.g. based on the BBS signature scheme over the BLS12-381 curve. The latter has attributes that are encoded as elements of $\mathbb{F}$ and can natively prove consistency with the Pedersen commitment scheme over $\mathbb{G}_1$. Since the ECDSA public key consists of a pair of $\mathbb{F}_p$ elements, namely the affine coordinates $Q = (x, y) \in \mathbb{F}_p^2$, we first need to *encode* each coordinate as elements of $\mathbb{F}$ and then use Pedersen commitments to commit to the latter. We encode the coordinates (x, y) as small limbs, i.e. small integers $0 \leq x_i < B$ s.t. $x = \sum_i x_i B^i$. Different values of B can be chosen, yielding also different trade-offs; for our concrete instantiations – which works over BLS12-381 curve – we choose $B = 2^{128}$ which minimizes the size of the commitment (and, therefore, the number of committed disclosure proofs). More concretely, to commit to a P-256 point $Q = (x, y)$, we compute 128-bit limbs $0 \leq x_l, x_h, y_l, y_h < 2^{128}$ s.t. $x = x_l + 2^{128} x_h$ and $y = y_l + 2^{128} y_h$, and we then commit to the four limbs using Pedersen over BLS12-381. The latter constitute the commitment to Q. In the context of AC, the correctness of the encoding of ECDSA's public key is verified at issuance of the credential. In particular, both $0 \leq x_l, x_h, y_l, y_h < 2^{128}$ and the fact that (x, y) is a valid P-256 point are verified: we assume here that they *a priori* hold.

4.1. Π_{Circ}: Circuit-based PoP

Our first two constructions are based on (simple) arithmetic circuits to prove P-256 operations. The constructions proceed in three steps. First, we apply the CSchnorr protocol (Sec. 3) to reduce the complexity of the proven relation. Second, we transfer the commitment to a form compatible with the circuit-based proving system: Pedersen commitments must first be converted before invoking the proving system. Third, we prove the resulting statement using a generic PoK. We call this composition Π_{Circ} and discuss the instantiation of each component below.

Recall from Sec. 3 the relation $\mathcal{R}_{\mathsf{C\text{-}DLOG}} =$

$$\left\{ \begin{matrix} pp := \mathsf{ck} \\ \mathsf{x} := \left(C_Q, H, K\right) \\ \mathsf{w} := \left(\rho_Q, Q, z\right) \end{matrix} \middle| \begin{matrix} C_Q = \mathsf{GCS.Com}_{\mathsf{ck}}(Q; \rho_Q) \wedge \\ zK = H + Q \end{matrix} \right\}$$

and note that a statement of $\mathcal{R}_{\mathsf{pECDSA}}$ is trivially reduced to it by simply setting $H := \alpha G_p$. Our new goal is therefore to efficiently prove the latter.

Step 1: Engaging in Π_{CSchnorr}. After expressing $\mathcal{R}_{\mathsf{ECDSA}}$ as an instance of the relation $\mathcal{R}_{\mathsf{C\text{-}DLOG}}$, $\mathcal{P}$ and $\mathcal{V}$ engage in an execution of Π_{CSchnorr} to reduce the $\mathcal{R}_{\mathsf{C\text{-}DLOG}}$ statement to a $\mathcal{R}_{\mathsf{CSchnorr}}$ statement. Recall that the latter corresponds to $\mathcal{R}_{\mathsf{CSchnorr}} =$

$$\left\{ \begin{matrix} pp := \mathsf{ck} \\ \mathsf{x} := \left(C_Q, C_R, T, c\right) \\ \mathsf{w} := \left(Q, R, \rho_Q, \rho_R\right) \end{matrix} \middle| \begin{matrix} C_Q = \mathsf{GCS.Com}_{\mathsf{ck}}(Q; \rho_Q) \wedge \\ C_R = \mathsf{GCS.Com}_{\mathsf{ck}}(R; \rho_R) \wedge \\ T = R + cQ \end{matrix} \right\}$$

Recall also from Sec. 2.2.2 that we denote with $\mathcal{R}_{\mathsf{GCS,P}}$ a committed relation using the commitment scheme GCS that captures that the committed values satisfy the predicate P. Defining $\mathsf{P}((T, c), (Q, R), \perp) := T \stackrel{?}{=} R + cQ$ – which

essentially captures the final verification test of the Schnorr protocol – we can view $\mathcal{R}_{\mathsf{CSchnorr}}$ as the committed relation $\mathcal{R}_{\mathsf{GCS},\mathsf{P}}$ over GCS.

Step 2: Preparing the circuit statement. The above committed relation $\mathcal{R}_{\mathsf{GCS},\mathsf{P}}$ considers two GCS commitments to the two committed points Q, R. This relation, however, may not be necessarily efficiently provable via a generic circuit-based PoK. In particular, we need to rely on a circuit-based construction that supports *natively committing to (parts of) the witness* so that we do not need to arithmetize the commitment scheme. However, the commitment scheme supported by the circuit can be different than GCS. We therefore first run a commitment transfer protocol to "transfer" the commitments to Q, R to some different commitment scheme $\widetilde{\mathsf{GCS}}$ that is natively supported by the proving system we intend to use (see Section 2.2.2).

Concretely, let $\Pi_{\mathcal{C},\mathsf{P}}$ be a public coin, HVZK PoK for the same committed relation under $\widetilde{\mathsf{GCS}}$, namely, $\Pi_{\mathcal{C},\mathsf{P}} : \mathcal{R}_{\widetilde{\mathsf{GCS}},\mathsf{P}} \to \mathcal{R}_\top$ and recall from Sec. 2.2.2 that $\Pi_{\mathsf{GCS}\to\widetilde{\mathsf{GCS}}}$ reduces validity of a committed relation under GCS to validity of the same relation under a different commitment scheme $\widetilde{\mathsf{GCS}}$. $\mathcal{P}$ and $\mathcal{V}$ engage in an interaction to transfer the commitment from the original commitment scheme GCS to $\widetilde{\mathsf{GCS}}$ which is the one that is supported by the circuit-based PoK (or later in the Schnorr-based PoK).

Step 3: Invoking $\Pi_{\mathcal{C},\mathsf{P}}$. The prover and verifier are left with the task of engaging in a protocol for the validity of $\mathcal{R}_{\widetilde{\mathsf{GCS}},\mathsf{P}}$. Having reduced to using the appropriate commitment scheme, they can rely on the PoK $\Pi_{\mathcal{C},\mathsf{P}}$ to finalize the proof.

Putting it all together. We present in Fig 2 our first construction which is the sequential composition of the RoKs presented above, namely

$$\Pi_{\mathcal{C},\mathsf{P}} \circ \Pi_{\mathsf{GCS}\to\widetilde{\mathsf{GCS}}} \circ \Pi_{\mathsf{CSchnorr}} : \mathcal{R}_{\mathsf{C\text{-}DLOG}} \to \mathcal{R}_\top$$

where the $\mathcal{R}_\top$ is the trivial relation as explained in Sec. 2.1, capturing that the latter protocol is a PoK.

Theorem 2. *Let $\mathsf{GCS}, \widetilde{\mathsf{GCS}}$ be perfectly hiding commitment schemes with message space $\mathbb{G}_p$. Also, let $\Pi_{\mathcal{C},\mathsf{P}} : \mathcal{R}_{\widetilde{\mathsf{GCS}},\mathsf{P}} \to \mathcal{R}_\top$ be a public coin, HVZK PoK, where*

$$\mathsf{P}((T,c),(Q,R),\bot) = T \stackrel{?}{=} R + cQ$$

Then, the protocol Π_{Circ} described in Fig. 2 is a public-coin HVZK PoK for $\mathcal{R}_{\mathsf{pECDSA}}$.

Proof. By construction, Π_{Circ} is a sequential composition of reductions of knowledge. Indeed, computing the statement/witness $\mathsf{x}_{\mathsf{C\text{-}DLOG}}, \mathsf{w}_{\mathsf{C\text{-}DLOG}}$ can be viewed as a trivial (in the sense that requires no interaction) reduction of knowledge $\Pi : \mathcal{R}_{\mathsf{pECDSA}} \to \mathcal{R}_{\mathsf{C\text{-}DLOG}}$ and the protocol as a whole is the composed reduction of knowledge

$$\Pi_{\mathsf{Circ}} = \Pi_{\mathcal{C},\mathsf{P}} \circ \Pi_{\mathsf{GCS}\to\widetilde{\mathsf{GCS}}} \circ \Pi_{\mathsf{CSchnorr}} \circ \Pi : \mathcal{R}_{\mathsf{C\text{-}DLOG}} \to \mathcal{R}_\top$$

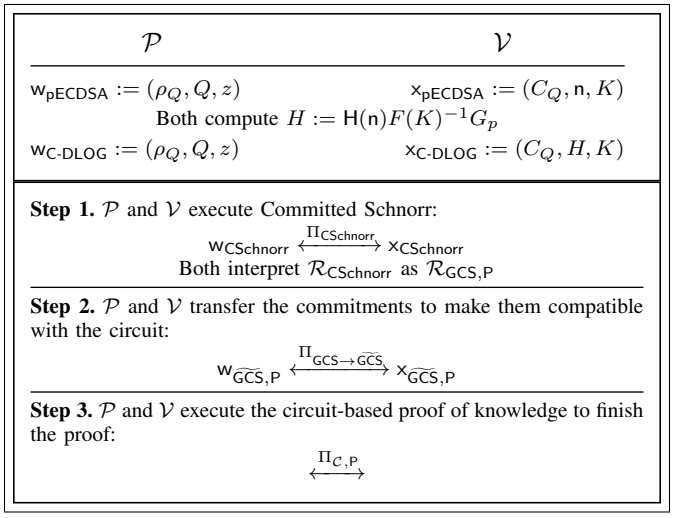


Figure 2: $\Pi_{\mathsf{Circ}} : \mathcal{R}_{\mathsf{pECDSA}} \to \mathcal{R}_\top$. The arrows indicate interactions between $\mathcal{P}$ and $\mathcal{V}$.

Therefore by Theorem 8, Π_{Circ} is a public coin, HVZK proof of knowledge for the relation $\mathcal{R}_{\mathsf{pECDSA}}$. □

We next describe two concrete instantiations of Π_{Circ} with different trade-offs.

Construction PoP-BP. Our first instantiation of Π_{Circ} uses a proving system based on Bulletproofs [27], in particular the constructions described in [34], [35]. The construction essentially consists of a simple reduction of R1CS to an inner-product relation which is in turn proven via the techniques of [27], [36]. We use an instantiation over the T-256 curve so that the predicate $T = R + cQ$ is native and can be efficiently proven with a simple circuit of a small size. The construction supports *committed inputs* when the commitment scheme $\widetilde{\mathsf{GCS}}$ is a *compact Pedersen commitment scheme over* T-256.

For the linking proof $\Pi_{\mathsf{GCS}\to\widetilde{\mathsf{GCS}}}$, let $a_1, \ldots, a_8$ be the 8 limbs of the coordinates of Q, R and $C_1, \ldots, C_8$ the corresponding Pedersen commitments over BLS12-381. Given a commitment key $G_1, \ldots, G_8, H$ over T-256, the prover must compute a compact Pedersen commitment $C = \sum_i a_i G_i + \rho H$ and prove its consistency with C_i. We do this in two steps: first the prover gives 8 plain Pedersen commitments $C'_i = a_i G_i + \rho_i H$ over T-256 curve and proves that for all i, C_i and C'_i have the same opening (as small integers). This is done via the techniques [19, Sec. 3]. Then both the prover and verifier can construct the compact commitment as $C = \sum_i C_i$ and the prover can also construct the corresponding opening $\rho = \sum_i \rho_i$. Note that because (1) we use different generators for each individual commitment and (2) the proof of [19] allows extracting a valid opening, the resulting compact commitment is indeed a commitment to $a_1, \ldots, a_8$. We also implicitly rely on the fact that C_i are guaranteed to have small values since this is required for soundness of [19]. Finally, with the parameters

we chose, in particular limb size of 128 bits, the protocol of [19] gives 112 bits of security.

We make two protocol level optimizations. First, we only consider the x coordinates of the committed P-256 points and let the prover witness the y coordinates in-circuit and prove their correctness. This means that each GCS commitment can be opened to two points, namely (x, y) and $(x, -y)$. This only minimally affects soundness error and reduces the proof size since the size of the commitment and the equality proofs are halved. Second, we run the commitment equality proofs of [19] before the execution of Π_{CSchnorr} so we only need to transfer two commitments (instead of four) corresponding to the limbs of the x-coordinate of Q. Then, after Π_{CSchnorr} the prover simply proves knowledge of opening of the commitment to R to construct the compact commitment. Calling the first step $\Pi_{\mathsf{BLS12\text{-}381}\to\mathsf{T\text{-}256}}$, the resulting construction is $\Pi_{\mathsf{Circ}} \circ \Pi_{\mathsf{BLS12\text{-}381}\to\mathsf{T\text{-}256}}$ and its security follows by the composition properties of RoKs (cf. Thm. 8).

The construction as a whole is based on the binding property of the commitment schemes used: Pedersen over BLS12-381 and T-256. Therefore, the only security assumption is the DLOG in the two groups. No trust/setup assumptions are needed since the proving system is *transparent*; the parameters consist of the Pedersen commitment keys.

Construction PoP-PLONK. We next discuss how to instantiate the components of Sec. 4.1, namely Π_{Circ} to get the FFA based construction PoP-PLONK. We instantiate $\Pi_{\mathcal{C},\mathsf{P}}$ with a variant of Plonk [29] with richer arithmetisation, in particular custom gates [37] and lookup tables [38]. The proving system supports "committed inputs" using the techniques described in [39]; in particular, it supports the KZG polynomial commitment scheme [28]. The commitment key is the same as in standard (compact) Pedersen commitment scheme except that the generators are evaluations of the Lagrange polynomials over a (secret) point encoded as group elements. By construction, the proving system is HVZK if enough blinding factors are used (depending on how the circuit is defined)[7]. The commitment $\widetilde{\mathsf{GCS}}$ is therefore instantiated with compact Pedersen commitment (over BLS12-381 curve) using the aforementioned distribution for the commitment key. We also apply the protocol level optimization we applied in PoP-BP, namely, only commiting to the x-coordinates of points. Regarding the circuit implementation, we rely on the techniques of [24] to implement efficient foreign field arithmetic, in particular emulating $\mathbb{F}_p$ arithmetic over the scalar field of BLS12-381.

For the linking proof $\Pi_{\mathsf{GCS}\to\widetilde{\mathsf{GCS}}}$, we need to show that four (plain) Pedersen commitments and a compact Pedersen with 8 blinding factors – both over BLS12-381 – have the same opening. In particular, given a plain Pedersen commitment key G, H and a compact Pedersen with commitment key $\vec{G}', \vec{H}'$ we need to prove that:

$$\forall i \in \{1, \ldots, 4\} : C_i = a_i G + r_i H, \text{ and}$$

$$C' = \sum_{i=1}^{4} a_i G'_i + \sum_{j=1}^{8} r'_i H'_j$$

where $a_1, \ldots, a_4$ are the limbs of Q, R. These are linear relations and can be proven via Schnorr-type proofs (i.e. [40]).

The construction as a whole *only relies on the standard* BLS12-381 *and* P-256 *curves* and does not need an additional curve. The security assumptions are inherited from the proving system; in this case Plonk, which is secure in the Algebraic Group Model [41]. The construction requires a KZG polynomial commitment key which is universal – i.e. requires no trusted setup per circuit. Also it is updateable [42] which means it can be sampled by simple protocols. A single honest party is enough to guarantee security. Such parameters, sampled by thousands of users, already exist and are deployed in practice (i.e. [43]).

4.2. Π_{Σ}: PoP via Schnorr-Protocols

We next describe our third construction. Unlike the previous ones, however, it requires no circuit-based proofs and is built *exclusively from Schnorr-type protocols* – specifically, PoK for the relations $\mathcal{R}_{\mathsf{PA}}$ (point addition) and $\mathcal{R}_{\mathsf{SM}}$ (scalar multiplication) (Eq. 2) and proofs of opening of Pedersen commitments across different groups. This construction was first implemented in [17], [30] and formalised in [22], but only with an informal security analysis. We provide an analysis in the RoK framework, and improve the construction by optimising the protocols for proving the $\mathcal{R}_{\mathsf{PA}}$ and $\mathcal{R}_{\mathsf{SM}}$ relations, which are the dominant efficiency bottlenecks.

Recall that our starting point is again the relation $\mathcal{R}_{\mathsf{pECDSA}}$ with GCS implemented using Pedersen over BLS12-381 and limb decomposition of the x, y affine coordinates of the committed points.

We start with a commitment transfer RoK (Sec. 2.2.2) to transfer the commitments from BLS12-381 to the T-256 which makes the computation native. To this end, we again use the techniques of [19, Sec. 3] similarly to the PoP-BP construction. Following [19, Sec. 3.2], by the homomorphic properties of Pedersen commitment scheme, the resulting commitment to Q consists of exactly two Pedersen commitments C_x, C_y committing to its coordinates. Therefore, $\widetilde{\mathsf{GCS}}$ has as commitment key for two T-256 points G_t, H_t and to commit to $Q = (x, y)$ we compute $C_x = xG_t + \rho_x H_t$ and $C_y = yG_t + \rho_y H_t$. We then derive the corresponding $\mathcal{R}_{\mathsf{C\text{-}DLOG}}$ statement/witness pair and reduce the problem to proving that $zK = H + Q$, where z is a witness and Q is now committed using Pedersen over T-256. The statement can then be proven in two steps:

1) Prove knowledge of z such that $zK = Z$ where Z is committed using $\widetilde{\mathsf{GCS}}$ and communicated to $\mathcal{V}$,

7. During the proof, the prover reveals evaluations of the committed polynomial which leaks information about the committed input. Using enough blinding factors counters, one for each polynomial evaluation provided by the prover) guarantees no information about the committed inputs is revealed.

2) Prove the predicate $Z = H + Q$ is satisfied where Z, Q are committed.

This can be modeled as a parallel execution of

$$\Pi_{\mathsf{SM}} : \mathcal{R}_{\mathsf{SM}} \to \mathcal{R}_{\top}, \quad \Pi_{\mathsf{PA}} : \mathcal{R}_{\mathsf{PA}} \to \mathcal{R}_{\top}$$

We first present this in Fig. 3 and analyze a inner RoK

$$\Pi_{\mathbb{G}} : \mathcal{R}_{\mathsf{C\text{-}DLOG}} \to \mathcal{R}_{\mathsf{SM}} \times \mathcal{R}_{\mathsf{PA}}$$

that performs the aforementioned reduction. We then present our final protocol Π_{Σ} in Fig. 4 that uses $\Pi_{\mathbb{G}}$.

$\mathcal{P}$		$\mathcal{V}$
$\mathsf{w}_{\mathsf{C\text{-}DLOG}} := (\rho_Q, Q, z)$		$\mathsf{x}_{\mathsf{C\text{-}DLOG}} := (C_Q, H, K)$
$Z := zK$		
$C_Z \leftarrow \mathsf{GCS.Com}_{\mathsf{ck}}(Z; \rho_Z)$		
	$\xrightarrow{\quad C_Z \quad}$	
	Both compute $C_H := \mathsf{GCS.Com}_{\mathsf{ck}}(H; 0)$	
$\mathsf{w}_{\mathsf{SM}} := (Z, \rho_Z, z)$		$\mathsf{x}_{\mathsf{SM}} := (C_Z, K)$
w_{PA}	:=	$\mathsf{x}_{\mathsf{PA}} := (C_Z, C_H, C_Q)$
$(Z, \rho_Z, H, 0, Q, \rho_Q)$		
Output $\mathsf{w}_{\mathsf{SM}}, \mathsf{w}_{\mathsf{PA}}$		Output $\mathsf{x}_{\mathsf{SM}}, \mathsf{x}_{\mathsf{PA}}$

Figure 3: $\Pi_{\mathbb{G}} : \mathcal{R}_{\mathsf{C\text{-}DLOG}} \to \mathcal{R}_{\mathsf{SM}} \times \mathcal{R}_{\mathsf{PA}}$. The arrow indicates the message sent from $\mathcal{P}$ to $\mathcal{V}$.

Theorem 3. *Let* GCS *be a perfectly hiding commitment scheme with message space* $\mathbb{G}$*. Then, the protocol* $\Pi_{\mathbb{G}}$ *described in Fig. 3 is an public-coin HVZK reduction of knowledge* $\Pi_{\mathbb{G}} : \mathcal{R}_{\mathsf{C\text{-}DLOG}} \to \mathcal{R}_{\mathsf{SM}} \times \mathcal{R}_{\mathsf{PA}}$.

Proof.

Completeness and public reducibility. Both properties follow by construction. Indeed, assuming a valid statement/witness pair $\mathsf{x}_{\mathsf{C\text{-}DLOG}}, \mathsf{w}_{\mathsf{C\text{-}DLOG}}$, and that C_Z is indeed a valid commitment to $Z = zK$, the equation $zK = H + Q$ holds which implies that for $Z = zK$

$$zK = Z, \quad Z = H + Q$$

For public reducibility, it is enough to note that the new statement can be deterministically constructed given the commitment C_Z and the statement $\mathsf{x}_{\mathsf{C\text{-}DLOG}}$.

Knowledge Soundness. Let $\mathcal{A}$ and $\mathcal{P}^*$ be PPT adversaries as described in Definition 1. We construct a PPT extractor $\mathcal{E}$ that works as follows: on input $\mathsf{x}_{\mathsf{C\text{-}DLOG}}$ and st, $\mathcal{E}$ runs $\mathsf{w}_{\mathsf{SM}}, \mathsf{w}_{\mathsf{PA}} \leftarrow \mathcal{P}^*(\mathsf{x}_{\mathsf{C\text{-}DLOG}}, \mathsf{st})$[8] and parses it as:

$$\mathsf{w}_{\mathsf{SM}} := Z_1, \rho_{Z_1}, z_1, \qquad \mathsf{w}_{\mathsf{PA}} := Z_2, \rho_{Z,2}, H_2, \rho_{H_2}, Q_2, \rho_{Q_2}$$

It then outputs $\mathsf{w}_{\mathsf{C\text{-}DLOG}} := \rho_{Q_2}, Q_2, z_1$. We claim that the probability that $(\mathsf{x}_{\mathsf{C\text{-}DLOG}}, \mathsf{w}_{\mathsf{C\text{-}DLOG}}) \in \mathcal{R}_{\mathsf{C\text{-}DLOG}}$ is negligibly close to the probability that $((\mathsf{x}_{\mathsf{SM}}, \mathsf{w}_{\mathsf{SM}}), (\mathsf{x}_{\mathsf{PA}}, \mathsf{w}_{\mathsf{PA}})) \in \mathcal{R}_{\mathsf{SM}} \times \mathcal{R}_{\mathsf{PA}}$. Indeed, note that if the latter holds, then by definition $z_1 K = Z_1$ and $Z_2 = H_2 + Q_2$. Now, either $Z_1 = Z_2$ or we break the binding property of GCS which happens only with negligible probability. Assuming $Z_1 = Z_2$, we have that $z_1 K = H_2 + Q_2$. Arguing similarly, it should be the case that $H_2 = H$ or again we break the binding property of GCS. Assuming this is not the case, the extracted witness (1) satisfies $z_1 K = H + Q_2$ and (2) Q_2 is a valid opening of C_Q. Therefore $(\mathsf{x}_{\mathsf{C\text{-}DLOG}}, \mathsf{w}_{\mathsf{C\text{-}DLOG}}) \in \mathcal{R}_{\mathsf{C\text{-}DLOG}}$.

Honest Verifier Zero Knowledge. A simulator can simply output an arbitrary commitment, e.g. a commitment to the generator $C_H \leftarrow \mathsf{GCS.Com}(G)$. The simulated transcript is then indistinguishable from a real one by the hiding property of the commitment scheme.

□

$\mathcal{P}$	$\mathcal{V}$
$\mathsf{w}_{\mathsf{pECDSA}} := (\rho_Q, Q, z)$	$\mathsf{x}_{\mathsf{pECDSA}} := (C_Q, \mathsf{n}, K)$
Both compute $H := \mathsf{H}(\mathsf{n}) F(K)^{-1} G_p$	
$\mathsf{w}_{\mathsf{C\text{-}DLOG}} := (\rho_Q, Q, z)$	$\mathsf{x}_{\mathsf{C\text{-}DLOG}} := (C_Q, H, K)$
Step 1. $\mathcal{P}$ and $\mathcal{V}$ transfer the commitments to $\widetilde{\mathsf{GCS}}$, which is native for P-256:	
$\mathsf{w}'_{\mathsf{pECDSA}} \xleftrightarrow{\Pi_{\mathsf{GCS} \to \widetilde{\mathsf{GCS}}}} \mathsf{x}'_{\mathsf{pECDSA}}$	
Step 2. $\mathcal{P}$ and $\mathcal{V}$ reduce to two statements about scalar multiplication and point addition in P-256:	
$\mathsf{w}_{\mathsf{SM}}, \mathsf{w}_{\mathsf{PA}} \xleftrightarrow{\Pi_{\mathbb{G}}} \mathsf{x}_{\mathsf{SM}}, \mathsf{x}_{\mathsf{PA}}$	
Step 3. $\mathcal{P}$ and $\mathcal{V}$ prove the latter statements to finish the proof:	
$\xleftrightarrow{\Pi_{\mathsf{SM}} \times \Pi_{\mathsf{PA}}}$	

Figure 4: $\Pi_{\Sigma} : \mathcal{R}_{\mathsf{pECDSA}} \to \mathcal{R}_{\top}$. The arrows indicate interactions between $\mathcal{P}$ and $\mathcal{V}$.

Theorem 4. *Let* $\mathsf{GCS}, \widetilde{\mathsf{GCS}}$ *be perfectly hiding commitment schemes with message space* $\mathbb{G}$ *encoded as* $\mathbb{F}^{\kappa}, \mathbb{F}_p^{\ell}$ *respectively. Then, the protocol* Π_{Σ} *described in Fig. 2 is a public-coin HVZK PoK for* $\mathcal{R}_{\mathsf{pECDSA}}$.

Proof. We argue similarly to the proof of Theorem 2. Let $\Pi : \mathcal{R}_{\mathsf{ECDSA}} \to \mathcal{R}_{\mathsf{C\text{-}DLOG}}$ be the trivial reduction as in the proof of Theorem 2. Then, Π_{Σ} is by construction the composition of reductions of knowledge $(\Pi_{\mathsf{SM}} \times \Pi_{\mathsf{PA}}) \circ \Pi_{\mathbb{G}} \circ \Pi_{\mathsf{GCS} \to \widetilde{\mathsf{GCS}}} : \mathcal{R}_{\mathsf{pECDSA}} \to \mathcal{R}_{\top}$ and the theorem follows by Theorem 8. □

Remark 2. Note that we do not use the Π_{CSchnorr} protocol in the construction. The reason for this is that we instantiate the protocol Π_{SM} with a "bit-challenge" Schnorr-type protocol

8. Note that in this case, the protocol in non-interactive since the prover simply needs to send C_Z.

and then use parallel repetitions to boost soundness. Therefore, the soundness error is already fine-tuned and Π_{CSchnorr} yields no efficiency gains.

We next describe how to concretely instantiate Π_Σ to derive the construction PoP-Σ. In particular we show how to concretely instantiate $\mathcal{R}_{\mathsf{SM}}$ and $\mathcal{R}_{\mathsf{PA}}$. Our instantiations significantly optimize the approaches of [20], [21].

4.2.1. RoK for EC Scalar Multiplication. In this section we recall the construction of [20], [21] that we use to implement the RoK $\Pi_{\mathsf{SM}} : \mathcal{R}_{\mathsf{SM}} \to \mathcal{R}_\top$. We slightly modify the construction – which considered commitments to the scalar as well – to improve its efficiency. The relation we want to prove is

$$\mathcal{R}_{\mathsf{SM}} = \left\{ \begin{matrix} pp = \mathsf{ck} \\ \mathsf{x} = (C, K) \\ \mathsf{w} = (Z, \rho, z) \end{matrix} \;\middle|\; \begin{matrix} C = \widetilde{\mathsf{GCS}}.\mathsf{Com}_{\mathsf{ck}}(Z; \rho) \wedge \\ Z = zK \end{matrix} \right\}$$

where $\widetilde{\mathsf{GCS}}$ is as described above. To prove the relation the prover samples a random statement in the relation and the "combined" statement and reveals one depending on a challenge bit of the verifier. To prove correctness of the combined statement shared with the verifier, it invokes $\Pi_{\mathsf{PA},\{0,1\}}$ in parallel. We describe the protocol in Fig. 5.

$\mathcal{P}$	$\mathcal{V}$
$\mathsf{w} := (Z, z, \rho)$	$\mathsf{x} := (C, K)$
$\omega \leftarrow \mathbb{F}_q \setminus \{0, z, 2z\}$	
$Z' := \omega Z$	
$Z'' := (\omega - z)Z$	
$C' = \widetilde{\mathsf{GCS}}.\mathsf{Com}(\mathsf{ck}, Z'; \rho')$	
C'' = $\widetilde{\mathsf{GCS}}.\mathsf{Com}(\mathsf{ck}, Z''; \rho'')$	
$\mathsf{x}_{\mathsf{PA}} = (C, C'', C')$	
w_{PA} = $(Z, \rho, Z'', \rho'', Z', \rho')$	
Compute m_1	
$\xrightarrow{\quad C', C'', m_1 \quad}$	
$\xleftarrow{\quad b \leftarrow \{0,1\} \quad}$	
Compute m_2	
If $b = 0$: $(\alpha = z', \tau = \rho')$	
If $b = 1$: $(\alpha = z'', \tau = \rho'')$	
$\xrightarrow{\quad \alpha, \tau, m_2 \quad}$	
	Verify $\Pi_{\mathsf{PA},\{0,1\}}$
	If $b = 0$:
	$C' \stackrel{?}{=} \widetilde{\mathsf{GCS}}.\mathsf{Com}(\alpha K; \tau)$
	If $b = 1$:
	$C'' \stackrel{?}{=} \widetilde{\mathsf{GCS}}.\mathsf{Com}(\alpha K; \tau)$

Figure 5: $\Pi_{\mathsf{SM}} : \mathcal{R}_{\mathsf{SM}} \to \mathcal{R}_\top$. In the above, m_1, m_2 denote the first and second prover message in an execution of $\Pi_{\mathsf{PA},\{0,1\}}$.

Theorem 5. *The protocol Π_{SM} described in Fig. 5 is a public-coin HVZK RoK $\Pi_{\mathsf{SM}} : \mathcal{R}_{\mathsf{SM}} \to \mathcal{R}_\top$. The notation $\mathbb{F}_q$ refers to the scalar field of* P-256.

Proof. Completeness and public reducibility follow by construction. We next focus on the HVZK and knowledge soundness.

HVZK. We describe a simulator that produces transcripts that are statistically close to an actual protocol execution[9]. The simulator first samples the random bit b. We consider two cases:

$b = 0$**:** it samples $\alpha \leftarrow \mathbb{F}_q \setminus \{0\}$ and computes:

$$C' = \widetilde{\mathsf{GCS}}.\mathsf{Com}(\mathsf{ck}, \alpha K; \tau)$$

and a random commitment C'' (in the case of Pedersen over T-256 this corresponds to two random field elements). It invokes the simulator for $\Pi_{\mathsf{PA},\{0,1\}}$ to create a transcript for the case of $b = 0$. Finally, it sets:

$$\mathsf{tr} := (C', C'', m_1, b, \alpha, \tau, m_2)$$

$b = 1$**:** it samples $\alpha \leftarrow \mathbb{F}_q \setminus \{0\}$ and computes:

$$C'' = \widetilde{\mathsf{GCS}}.\mathsf{Com}(\mathsf{ck}, \alpha K; \tau)$$

and a random commitment C' (in the case of Pedersen over T-256 this corresponds to two random field elements). It invokes the simulator for $\Pi_{\mathsf{PA},\{0,1\}}$ to create a transcript for the case of $b = 1$. Finally, it sets:

$$\mathsf{tr} := (C', C'', m_1, b, \alpha, \tau, m_2)$$

In both cases, the transcript corresponds to uniformly distributed elements conditioned on the verifier accepting. This is the same as in the real execution except that in that case the value ω is sampled from a $\mathbb{F} \setminus \{z, 2z\}$[10] and therefore the statistical distance of the real and the simulated transcript is $\frac{2}{|\mathbb{F}_p| - 1}$.

Knowledge Soundness. We rely on Lemma 1 which in this case reduces our problem to proving special soundness. Assume we have two accepting transcripts with for $b = 0$ and $b = 1$ with the same first message. Let

$$\mathsf{tr}_0 = (C', C'', m_1, 0, \alpha_0, \tau_0, m_{2,0})$$
$$\mathsf{tr}_1 = (C', C'', m_1, 1, \alpha_1, \tau_1, m_{2,1})$$

Recall that each of C, C'', C' is a pair of Pedersen commitments to the affine x, y coordinates of the corresponding point; following the indexing of the $\Pi_{\mathsf{PA},\{0,1\}}$ extractor, the statement $\mathsf{x}_{\mathsf{PA}} = (C, C'', C')$ corresponds to

$$C = (C_1, C_2), \quad C'' = (C_3, C_4), \quad C' = (C_5, C_6),$$

9. As noted in [21] the protocol achieves statistical (and not perfect) HVZK since the message ω is sampled from $\mathbb{F}_q \setminus \{0, z, 2z\}$ instead of $\mathbb{F}_q$.

10. Note that this assumes some a priori knowledge about the statement. If the commitment to Z is simply a uniformly distributed element, no information is leaked.

committing to the points Z, Z'', Z' respectively. Since the $\Pi_{\mathsf{PA},\{0,1\}}$ verifier accepts in both cases, these committed points satisfy the point-addition relation $Z + Z'' = Z'$ with overwhelming probability. Moreover, the second verification check guarantees that the commitments C' and C'' are honestly computed, i.e.

$$C' = \widetilde{\mathsf{GCS}}.\mathsf{Com}(\alpha_0 K; \tau_0), \quad C'' = \widetilde{\mathsf{GCS}}.\mathsf{Com}(\alpha_1 K; \tau_1),$$

We claim that the discrete logarithm of Z w.r.t. K is $z = \alpha_0 - \alpha_1$ except with negligible probability. Indeed, we consider two cases:

1) If $Z' \neq \alpha_0 K$ or $Z'' \neq \alpha_1 K$ we have two openings for either Z' or Z'' and we break the binding property of $\widetilde{\mathsf{GCS}}$.
2) If $Z' = \alpha_0 K$ and $Z'' = \alpha_1 K$ then

$$Z = Z' - Z'' = \alpha_0 K - \alpha_1 K = (\alpha_0 - \alpha_1) K$$

and therefore $z = \alpha_0 - \alpha_1$ is the discrete logarithm of Z w.r.t K.

It remains to recover the randomness $\rho = (r_1, r_2)$ for which $C = \widetilde{\mathsf{GCS}}.\mathsf{Com}(Z; \rho)$. The x-coordinate randomness r_1 is recovered directly by the point-addition extractor. The y-coordinate randomness r_2, however, is *not* determined by the point-addition extractor alone. Since the opening proof of C_2 is dropped (Optimisation 3), the extractor may recover the openings to $C_{p_1} = C_4 - C_2$ and $C_{p_3} = C_2 + C_6$, which fix $r_4 - r_2$ and $r_2 + r_6$ – two equations with three unknowns, leaving r_2 undetermined. The missing information is supplied by the second verification check of Π_{SM}: for $b = 1$ it opens $C'' = (C_3, C_4)$ in full and thereby reveals the randomness r_4 of C_4, which together with $r_4 - r_2$ fixes r_2 (similarly, for $b = 0$ the opening of C' reveals r_6 which together with $r_2 + r_6$ fixes r_2). The extractor thus outputs the witness (Z, ρ, z) with $Z = (\alpha_0 - \alpha_1)K$ and $z = \alpha_0 - \alpha_1$. □

4.2.2. RoK for EC Point Addition. In this section, we optimise Π_{PA} of [21] (CDLS), whose repeated invocation within Π_{SM} – to prove commitment consistency – is the main overhead of PoP-Σ. Our optimisations improve its performance without sacrificing security.

First, we recall the protocols of [20], [21] used to implement the original RoK $\Pi_{\mathsf{SM}} : \mathcal{R}_{\mathsf{SM}} \to \mathcal{R}_\top$. Given non-identity points $P_1 = (a_x, a_y)$ and $P_2 = (b_x, b_y)$ s.t. $P_1 \neq P_2$ and $P_1 \neq -P_2$, the point $P_3 = P_1 + P_2$ is computed as

$$t_x = \left(\frac{b_y - a_y}{b_x - a_x}\right)^2 - a_x - b_x$$
$$t_y = \frac{b_y - a_y}{b_x - a_x}(a_x - t_x) - a_y$$

Setting $\tau = (b_y - a_y)(b_x - a_x)^{-1}$ we can express the above equations as a simple R1CS system over the base field of P-256 defined by the equations:

$$\begin{aligned} \tau(b_x - a_x) &= b_y - a_y \\ \tau^2 &= a_x + b_x + t_x \\ \tau(a_x - t_x) &= a_y + t_y \end{aligned} \tag{5}$$

If these equations are satisfied and additionally $b_x \neq a_x$[11] then indeed $P_3 = (t_x, t_y) = P_1 + P_2$. This can be turned to a PoK for $\mathcal{R}_{\mathsf{PA}}$ by having the prover send a Pedersen commitment of the value τ and prove knowledge of openings for all commitments that satisfy the above equations. We can express this as a RoK. In particular, let $G_t, H_t \in \mathbb{G}_t$, where $\mathbb{G}_t$ is the group of T-256, be a Pedersen commitment key and consider the following relations:

$$\mathcal{R}_{\mathsf{PD}} = \left\{ \begin{matrix} pp := (G_t, H_t) \\ \mathsf{x} := C,\ \mathsf{w} := (m, \rho) \end{matrix} \;\middle|\; C = mG_t + \rho H_t \right\}$$

$$\mathcal{R}_{\mathsf{MUL}} = \left\{ \begin{matrix} pp := (G_t, H_t) \\ \mathsf{x} := (C_1, C_2, C_3) \\ \mathsf{w} := \begin{pmatrix} m_1, \rho_1, \\ m_2, \rho_2, \rho_3 \end{pmatrix} \end{matrix} \;\middle|\; \begin{matrix} C_1 = m_1 G_t + \rho_1 H_t \\ C_2 = m_2 G_t + \rho_2 H_t \\ C_3 = m_1 m_2 G_t + \rho_3 H_t \end{matrix} \right\}$$

$$\mathcal{R}_{\mathsf{NZ}} = \left\{ \begin{matrix} pp := (G_t, H_t) \\ \mathsf{x} := C,\ \mathsf{w} := (m, \rho) \end{matrix} \;\middle|\; \begin{matrix} C = mG_t + \rho H_t \\ m \neq 0 \end{matrix} \right\}$$

We also define the relation $\mathcal{R}_{\mathsf{PA}^*}$ which is the same as $\mathcal{R}_{\mathsf{PA}}$ but also (1) guarantees that $P_1 \neq P_2$ and $P_1 \neq -P_2$ and (2) assumes that $P_1, P_2 \in \mathbb{G}_t$. In our use-case the latter "promise" is guaranteed to hold[12].

$$\mathcal{R}_{\mathsf{PA}^*} = \left\{ \begin{matrix} pp = \mathsf{ck} \\ \mathsf{x} = (C_i)_{i=1}^3 \\ \mathsf{w} = (P_i, \rho_i)_{i=1}^3 \end{matrix} \;\middle|\; \begin{matrix} (pp, \mathsf{x}, \mathsf{w}) \in \mathcal{R}_{\mathsf{PA}} \wedge \\ P_1 \neq P_2 \wedge P_1 \neq -P_2 \vee \\ P_1 \notin \mathbb{G}_p \ \vee \ P_2 \notin \mathbb{G}_p \end{matrix} \right\}$$

The work of [21] implies a reduction of knowledge[13] for

$$\Pi_{\mathsf{PA}^*} : \mathcal{R}_{\mathsf{PA}^*} \to \mathcal{R}_{\mathsf{MUL}}^3 \times \mathcal{R}_{\mathsf{PD}} \times \mathcal{R}_{\mathsf{NZ}}$$

where we denote $\mathcal{R}_{\mathsf{MUL}}^3 = \mathcal{R}_{\mathsf{MUL}} \times \mathcal{R}_{\mathsf{MUL}} \times \mathcal{R}_{\mathsf{MUL}}$.

Optimisation 1: A proof of knowledge for $\mathcal{R}_{\mathsf{Sq}}$. Observe that in the multiplication proof for proving consistency of the second equation of equations (5) we prove knowledge of a squaring relation. Rather than using a proof of a multiplicative identity $\mathcal{R}_{\mathsf{MUL}}$, as in [21], we can instead use a PoK (or RoK) for a (special case) of the relation, $\mathcal{R}_{\mathsf{Sq}}$, defined as:

$$\mathcal{R}_{\mathsf{Sq}} = \left\{ \begin{matrix} pp := (G_t, H_t) \\ \mathsf{x} := (C_1, C_2) \\ \mathsf{w} := (m, \rho_1, \rho_2) \end{matrix} \;\middle|\; \begin{matrix} C_1 = mG_t + \rho_1 H_t \\ C_2 = (m^2) G_t + \rho_2 H_t \end{matrix} \right\}$$

We define a protocol Π_{Sq} for proving $\mathcal{R}_{\mathsf{Sq}}$ in Fig. 6. The protocol is a simple modification of the standard proof of knowledge for $\mathcal{R}_{\mathsf{MUL}}$.

Theorem 6. *The protocol Π_{Sq} described in Fig. 6 is a public coin HVZK reduction of knowledge for $\mathcal{R}_{\mathsf{Sq}} \to \mathcal{R}_\top$, where $\mathbb{F}$ is the field of scalars of G and H.*

Proof. We prove the protocol is Σ-protocol for $\mathcal{R}_{\mathsf{Sq}}$.

Completeness. By the verification equations, an honest prover will be accepted by the verifier with probability 1.

11. This can also be expressed as the constraint $(b_x - a_x)r = 1$.
12. In Π_Σ the points correspond to the public key Q and a public point $\alpha\mathbb{G}_p$ committed by the verifier as well.
13. But explictly given as an AND-composition of Σ-protocols.

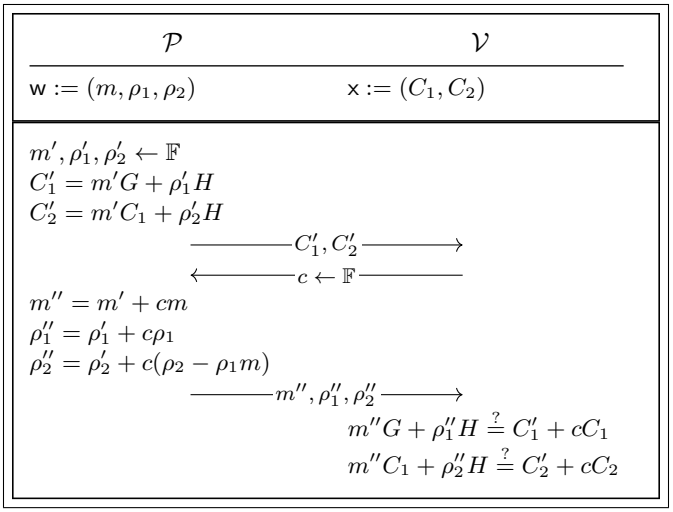


Figure 6: $\Pi_{\mathsf{Sq}} : \mathcal{R}_{\mathsf{Sq}} \to \mathcal{R}_\top$ for $\mathcal{R}_{\mathsf{Sq}} = \{(\mathsf{x}, \mathsf{w}) : C_1 = mG + \rho_1 H \wedge C_2 = m^2 G + \rho_2 H\}$.

Special Soundness. Given two colliding transcripts $(C_1', C_2', c, m'', \rho_1'', \rho_2'')$ and $(C_1', C_2', \widetilde{c}, \widetilde{m''}, \widetilde{\rho_1''}, \widetilde{\rho_2''})$ with $c \neq \widetilde{c}$, we can extract a witness m, ρ_1, ρ_2 for $\mathcal{R}_{\mathsf{Sq}}$ as follows:

Derive $m = \frac{m'' - \widetilde{m''}}{c - \widetilde{c}}$, $\rho_1 = \frac{\rho_1'' - \widetilde{\rho_1''}}{c - \widetilde{c}}$ and $\rho_2 = \frac{\rho_2'' - \widetilde{\rho_2''}}{c - \widetilde{c}} + m \cdot \rho_1$. We first show that (m, ρ_1) is a valid opening of C_1. Recall that the verification equations

$$m''G + \rho_1'' H = C_1' + cC_1$$
$$\widetilde{m''}G + \widetilde{\rho_1''} H = C_1' + \widetilde{c} C_1$$

hold. By substracting the second equation from the first, we have

$$(m'' - \widetilde{m''})G + (\rho_1'' - \widetilde{\rho_1''})H = (c - \widetilde{c})C_1.$$

Multiplying by $(c - \widetilde{c})^{-1}$ (since $c \neq \widetilde{c}$) gives $mG + \rho_1 H = C_1$. Next, we show that (m^2, ρ_2) is a valid opening of C_2. By the verification equations

$$m'' C_1 + \rho_2'' H = C_2' + cC_2$$
$$\widetilde{m''} C_1 + \widetilde{\rho_2''} H = C_2' + \widetilde{c} C_2$$

hold. By substracting the second equation from the first, we have

$$(m'' - \widetilde{m''})C_1 + (\rho_2'' - \widetilde{\rho_2''})H = (c - \widetilde{c})C_2.$$

Multiplying by $(c - \widetilde{c})^{-1}$ gives $mC_1 + (\rho_2 - \rho_1 m)H = C_2$. Since $C_1 = mG + \rho_1 H$, we have $C_2 = m^2 G + \rho_2 H$. Therefore, we have extracted a valid witness m, ρ_1, ρ_2 for $\mathcal{R}_{\mathsf{Sq}}$.

Honest Verifier Zero Knowledge. We construct a simulator $\mathcal{S}$ for Π_{Sq} as follows: on input $\mathsf{x} = (C_1, C_2)$ and challenge c, the simulator samples $m'', \rho_1'', \rho_2'' \leftarrow \mathbb{F}$ and $c \leftarrow \mathbb{F}$, and computes $C_1' = m''G + \rho_1'' H - cC_1$ and $C_2' = m'' C_1 + \rho_2'' H - cC_2$. The simulator outputs the transcript $(C_1', C_2', c, m'', \rho_1'', \rho_2'')$, which passes the verification equations by construction. Finally, we show that the real and simulated transcripts are identically distributed. Observe that both real (resp. simulated) transcripts are determined solely by the random coins of the prover (resp. the simulator). The prover's random coins are m', ρ_1', ρ_2' and the simulator's random coins are m'', ρ_1'', ρ_2''. Consider the mapping from the simulator's random coins to the prover's random coins defined by:

$$(m', \rho_1', \rho_2') \mapsto (m' + cm, \rho_1' + c\rho_1, \rho_2' + c\rho_2).$$

This clearly defines a bijection. Both the prover's and simulator's random coins are uniformly distributed, and thus, for every possible instance-challenge pair, both the honest prover and the simulator produce a given transcript with equal probability. □

We note that the above protocol reduces the proof size of the point addition proof by 1 EC point and 2 scalars.

Optimisation 2: Reusing commitment masking across multiplication proofs. In [21]'s point addition protocol, the prover performs three multiplication proofs Π_{MUL} (or two multiplication proofs and one squaring proof Π_{Sq}) with a common challenge to prove the point addition formulae in Equations (5) are satisfied. In each of these multiplication proofs (and also in the squaring proof), the prover samples fresh, independent random masks for the input commitments. However, it is possible to reuse the same random mask across the proofs with common input, corresponding to the commitment to τ. This reduces the proof size of Π_{PA} (resp. Π_{SM}) by a further 2 EC points and 4 scalars.

In order to prove security, we construct a direct RoK $\Pi_{\mathsf{PA}} : \mathcal{R}_{\mathsf{PA}} \to \mathcal{R}_\top$, which removes redundancy in the protocol transcript, and then show that the protocol is secure.

Optimisation 3: Removing Π_{PD} from the RoK. We note that the relation $\mathcal{R}_{\mathsf{PD}}$ on the commitment C_2 is only required in generality when extracting the witness for $\mathcal{R}_{\mathsf{PA}}$, but in the specific case of composition in Π_{SM} we can avoid the need for $\mathcal{R}_{\mathsf{PD}}$ entirely. In particular, we can directly extract the witness from the openings of a colliding transcript from the openings to the commitments C', C''.

The Optimised Protocol. Combining the three optimisations above yields the Σ-protocol Π_{PA} described in Fig. 7. The statement consists of Pedersen commitments $C_1, \ldots, C_6$ to the affine coordinates of the points $A = (a_x, a_y)$, $B = (b_x, b_y)$ and $T = (t_x, t_y)$, and the prover shows $A + B = T$ with $A \neq \pm B$ and $A, B \neq \mathcal{O}$.

Following the equations (5), we use the derived commitments $C_{f_1} = C_3 - C_1$, $C_{p_1} = C_4 - C_2$, $C_{p_2} = C_1 + C_3 + C_5$, $C_{f_3} = C_1 - C_5$ and $C_{p_3} = C_2 + C_6$ that both parties recompute from the statement; here C_{f_i} and C_{p_i} denote the commitments to (non-τ) *factor* and the resulting *product* of the i-th equation of (5). The squaring proof for the second equation realises Optimisation 1, the mask (a_τ, a_{r_τ}) of C_τ is shared across the three coordinate proofs realising Optimisation 2, and the proof of opening Π_{PD} on C_2 is written in grey as in Optimisation 3. Finally, $\Pi_{\mathsf{NZ}}(C_{f_1})$

(following [21, Construction 2.4]) proves $b_x - a_x \neq 0$, i.e. $A \neq \pm B$.

Theorem 7. *The protocol Π_{PA} described in Fig. 7 is a public coin HVZK RoK for $\mathcal{R}_{\mathsf{PA}} \to \mathcal{R}_{\top}$, where $\mathbb{F}$ is the field of scalars of G and H.*

Proof. We prove Π_{PA} is a Σ-protocol for $\mathcal{R}_{\mathsf{PA}}$.

Completeness. Both parties recompute the derived commitments, which for an honest witness open as

$$\begin{aligned} C_{f_1} &= (b_x - a_x)G + (r_3 - r_1)H, \\ C_{p_1} &= (b_y - a_y)G + (r_4 - r_2)H, \\ C_{f_3} &= (a_x - t_x)G + (r_1 - r_5)H, \\ C_{p_3} &= (a_y + t_y)G + (r_2 + r_6)H, \end{aligned}$$

together with $C_{p_2} = (a_x + b_x + t_x)G + (r_1 + r_3 + r_5)H$ and $C_\tau = \tau G + r_\tau H$. Substituting the prover's responses into each check:

Opening checks $(1), (2), (5), (7)$ are linear; e.g. for (1),

$$z_\tau G + z_{r_\tau} H = (a_\tau + c\tau)G + (a_{r_\tau} + c r_\tau)H = T_1 + cC_\tau,$$

and identically $(2), (5), (7)$ open C_{f_1}, C_{f_3}, C_2 at $(b_x - a_x, r_3 - r_1)$, $(a_x - t_x, r_1 - r_5)$, (a_y, r_2).

Product checks $(3), (4), (6)$, we know that equations (5) hold, and hence for check (3),

$$\begin{aligned} z_\tau C_{f_1} + z_{e_1} H &= a_\tau C_{f_1} + c\tau\big((b_x - a_x)G + (r_3 - r_1)H\big) \\ &\quad + a_{e_1} H + c\big((r_4 - r_2) - \tau(r_3 - r_1)\big)H \\ &= T_3 + c\big((b_y - a_y)G + (r_4 - r_2)H\big) \\ &= T_3 + cC_{p_1}, \end{aligned}$$

where the $c\tau(r_3 - r_1)H$ terms cancel and $\tau(b_x - a_x) = b_y - a_y$. Checks $(4), (6)$ follow identically from $\tau \cdot \tau = a_x + b_x + t_x$ and $\tau(a_x - t_x) = a_y + t_y$.

Non-zero checks (8)–(10)*:* as $A \neq \pm B$ we have $b_x - a_x \neq 0$, so $U_1 = \beta_1(b_x - a_x)G \neq \mathcal{O}$ for $\beta_1 \neq 0$, giving (8); check (9) is $cU_1 + U_3 = (c\beta_1(b_x - a_x) + \beta_4)G = v_3 G$; and check (10) is

$$cU_1 + U_2 = c\beta_1(b_x - a_x)G + \beta_2 C_{f_1} + \beta_3 H = v_1 C_{f_1} + v_2 H.$$

Hence, all checks are satisfied and the verifier accepts with probability 1.

Special Soundness. Let $(\mathsf{com}, c, \mathsf{resp})$ and $(\mathsf{com}, c', \mathsf{resp}')$ be accepting with the same first message and $\Delta c := c - c' \neq 0$. For ease of notation, for eachresponse symbol z, we write $z^* := (z - z')/\Delta c$.

Step 1 (openings, checks $(1), (2), (5)$). Subtracting paired verification equations, and multiplying by $(\Delta c)^{-1}$, we have that

$$C_\tau = \tau G + r_\tau H, \quad C_{f_1} = f_1 G + \rho_{f_1} H, \quad C_{f_3} = f_3 G + \rho_{f_3} H,$$

with $\tau = z^*_\tau$, $r_\tau = z^*_{r_\tau}$, $f_1 = z^*_{f_1}$, $\rho_{f_1} = z^*_{r_{f_1}}$, $f_3 = z^*_{f_3}$, $\rho_{f_3} = z^*_{r_{f_3}}$. As $C_{f_1} = C_3 - C_1$ and $C_{f_3} = C_1 - C_5$, this fixes the extracted values $b_x - a_x := f_1$, $r_3 - r_1 := \rho_{f_1}$, $a_x - t_x := f_3$, $r_1 - r_5 := \rho_{f_3}$.

$\mathcal{P}$	$\mathcal{V}$
$\mathsf{w} := \begin{pmatrix} a_x, a_y, b_x, b_y, t_x, t_y \\ r_1, r_2, r_3, r_4, r_5, r_6 \end{pmatrix}$	$\mathsf{x} := \begin{pmatrix} C_1, C_2, C_3 \\ C_4, C_5, C_6 \end{pmatrix}$

Both parties compute

$$C_{f_1} = C_3 - C_1, \quad C_{p_1} = C_4 - C_2, \quad C_{p_2} = C_1 + C_3 + C_5,$$
$$C_{f_3} = C_1 - C_5, \quad C_{p_3} = C_2 + C_6$$

$\tau = \frac{b_y - a_y}{b_x - a_x}$, $\quad r_\tau \leftarrow \mathbb{F}$

$C_\tau = \tau G + r_\tau H$

$a_\tau, a_{r_\tau}, a_{f_1}, a_{r_{f_1}}, a_{e_1}, a_{e_2} \leftarrow \mathbb{F}$

$a_{f_3}, a_{r_{f_3}}, a_{e_3}, a_2, a_{r_2}, \beta_2, \beta_3, \beta_4 \leftarrow \mathbb{F}$

$\beta_1 \leftarrow \mathbb{F}^*$ ▷ β_1 must be nonzero for perfect completeness

$T_1 = a_\tau G + a_{r_\tau} H$, $\quad T_2 = a_{f_1} G + a_{r_{f_1}} H$

$T_3 = a_\tau C_{f_1} + a_{e_1} H$, $\quad T_4 = a_\tau C_\tau + a_{e_2} H$

$T_5 = a_{f_3} G + a_{r_{f_3}} H$, $\quad T_6 = a_{f_3} C_\tau + a_{e_3} H$

$T_7 = a_2 G + a_{r_2} H$, $\quad U_3 = \beta_4 G$

$U_1 = \big(\beta_1(b_x - a_x)\big)G$, $\quad U_2 = \beta_2 C_{f_1} + \beta_3 H$

$\mathsf{com} := (C_\tau, T_1, T_2, T_3, T_4, T_5, T_6, T_7, U_1, U_2, U_3)$

$\longrightarrow \mathsf{com} \longrightarrow$

$\longleftarrow c \leftarrow \mathbb{F} \longleftarrow$

$z_\tau = a_\tau + c\tau$, $\quad z_{r_\tau} = a_{r_\tau} + c\, r_\tau$

$z_{f_1} = a_{f_1} + c(b_x - a_x)$, $\quad z_{r_{f_1}} = a_{r_{f_1}} + c(r_3 - r_1)$

$z_{e_1} = a_{e_1} + c\big((r_4 - r_2) - (r_3 - r_1)\tau\big)$

$z_{e_2} = a_{e_2} + c\big((r_1 + r_3 + r_5) - r_\tau \tau\big)$

$z_{f_3} = a_{f_3} + c(a_x - t_x)$, $\quad z_{r_{f_3}} = a_{r_{f_3}} + c(r_1 - r_5)$

$z_{e_3} = a_{e_3} + c\big((r_2 + r_6) - r_\tau(a_x - t_x)\big)$

$z_2 = a_2 + c\, a_y$, $\quad z_{r_2} = a_{r_2} + c\, r_2$

$v_1 = \beta_2 + c\,\beta_1$, $\quad v_2 = \beta_3 - c\,\beta_1(r_3 - r_1)$

$v_3 = \beta_4 + c\,\beta_1(b_x - a_x)$

$\mathsf{resp} := \begin{pmatrix} z_\tau, z_{r_\tau}, z_{f_1}, z_{r_{f_1}}, z_{e_1}, z_{e_2}, z_{f_3}, \\ z_{r_{f_3}}, z_{e_3}, z_2, z_{r_2}, v_1, v_2, v_3 \end{pmatrix}$

$\longrightarrow \mathsf{resp} \longrightarrow$

(1) $z_\tau G + z_{r_\tau} H \stackrel{?}{=} T_1 + cC_\tau$

(2) $z_{f_1} G + z_{r_{f_1}} H \stackrel{?}{=} T_2 + cC_{f_1}$

(3) $z_\tau C_{f_1} + z_{e_1} H \stackrel{?}{=} T_3 + cC_{p_1}$

(4) $z_\tau C_\tau + z_{e_2} H \stackrel{?}{=} T_4 + cC_{p_2}$

(5) $z_{f_3} G + z_{r_{f_3}} H \stackrel{?}{=} T_5 + cC_{f_3}$

(6) $z_{f_3} C_\tau + z_{e_3} H \stackrel{?}{=} T_6 + cC_{p_3}$

(7) $z_2 G + z_{r_2} H \stackrel{?}{=} T_7 + cC_2$

(8) $U_1 \stackrel{?}{\neq} \mathcal{O}$

(9) $cU_1 + U_3 \stackrel{?}{=} v_3 G$

(10) $cU_1 + U_2 \stackrel{?}{=} v_1 C_{f_1} + v_2 H$

Figure 7: $\Pi_{\mathsf{PA}} : \mathcal{R}_{\mathsf{PA}} \to \mathcal{R}_{\top}$, the optimised point-addition Σ-protocol, where C_{f_i}, C_{p_i} are the (non-τ) factor and product of the i-th equation of (5). Checks (1)–(6) prove the multiplicative (resp. square) relations of equations; check (7) is the opening $\Pi_{\mathsf{PD}}(C_2)$; and checks (8)–(10) are the non-zero proof $\Pi_{\mathsf{NZ}}(C_{f_1})$. The grey elements realise $\Pi_{\mathsf{PD}}(C_2)$ and may be dropped when Π_{PA} is composed in Π_{SM}, where the witness is extracted directly from colliding transcripts (Optimisation 3).

Step 2 (products, checks $(3),(4),(6)$*).* Subtracting paired equations and substituting the openings of Step 1,

$$C_{p_1} = \tau C_{f_1} + e_1 H = \tau f_1\, G + (\tau\rho_{f_1} + e_1)H,$$
$$C_{p_2} = \tau C_\tau + e_2 H = \tau^2 G + (\tau r_\tau + e_2)H,$$
$$C_{p_3} = f_3 C_\tau + e_3 H = \tau f_3\, G + (f_3 r_\tau + e_3)H,$$

with $e_1 = z^*_{e_1}$, $e_2 = z^*_{e_2}$, $e_3 = z^*_{e_3}$. *Crucially, the value* τ *here is the one from check* (1): the same response z_τ occurs in checks $(1),(3),(4)$, so the τ extracted from the two product relations coincides with the opening of C_τ *by construction*. Reading off $C_{p_1} = C_4 - C_2$, $C_{p_2} = C_1 + C_3 + C_5$, $C_{p_3} = C_2 + C_6$ certifies

$$b_y - a_y := \tau f_1, \quad a_x + b_x + t_x := \tau^2, \quad a_y + t_y := \tau f_3,$$

together with the randomness values $r_4 - r_2 := \tau\rho_{f_1} + e_1$, $r_1 + r_3 + r_5 := \tau r_\tau + e_2$, $r_2 + r_6 := f_3 r_\tau + e_3$. In particular the three equations of (5) hold for the extracted values.

Step 3 (x-coordinates). From $b_x - a_x$, $a_x - t_x$ and $a_x + b_x + t_x$ we may recover a_x, b_x, t_x via solving a linear system of 3 equations with 3 unknowns. The identical system on the randomness triples $r_3 - r_1$, $r_1 - r_5$, $r_1 + r_3 + r_5$ recovers r_1, r_3, r_5. Clearly, by the additive homomorphic property of Pedersen commitments, C_1, C_3, C_5 open to a_x, r_1, b_x, r_3 and t_x, r_5 respectively.

Step 4 (y-coordinates, check (7)*).* Subtracting the paired opening equations for C_2 recovers $a_y := z^*_2$, $r_2 := z^*_{r_2}$ with $C_2 = a_y G + r_2 H$. Then $C_4 = C_{p_1} + C_2$ and $C_6 = C_{p_3} - C_2$ open to

$$b_y := a_y + \tau f_1, \quad r_4 := r_2 + \tau\rho_{f_1} + e_1$$
$$t_y := \tau f_3 - a_y, \quad r_6 := f_3 r_\tau + e_3 - r_2.$$

Composition in Π_{SM}. When Π_{PA} is composed in Π_{SM} (Figure 5), the opening proof $\Pi_{\mathsf{PD}}(C_2)$—check (7), the components in the commitment response phases—may be dropped (Optimisation 3). In Π_{SM}, the points $A = Z$, $B = Z''$, $T = Z'$ are committed by C, C'', C', and the final Π_{SM} check opens $C' = (C_5, C_6)$ and $C'' = (C_3, C_4)$ in full, revealing $Z' = \alpha_0 K$, $Z'' = \alpha_1 K$ and their randomness across the two challenge transcripts. The opening of C_4 then recovers that of C_2 from Step 2 via $C_2 = C_4 - C_{p_1}$, determining the y-coordinates without the components of the protocol written in grey.

Step 5 (non-zero, checks (8)–(10)*), following [21, Constr. 2.4].* Both transcripts share U_1, U_2, U_3. Subtracting the paired check (9) gives $\Delta c\, U_1 = (v_3 - v'_3)G$, i.e. $U_1 = \delta G$ with $\delta := v^*_3$. Subtracting the paired check (10) gives $\Delta c\, U_1 = (v_1 - v'_1)C_{f_1} + (v_2 - v'_2)H$, i.e. $U_1 = \mu C_{f_1} + \nu H$ with $\mu := v^*_1$, $\nu := v^*_2$. Equating, $\delta G = \mu C_{f_1} + \nu H$. If $\mu = 0$ then $\delta G = \nu H$, where either $\delta = \nu = 0$ (so $U_1 = \mathcal{O}$, contradicting check (8)) or a non-trivial relation $\log_G H = \delta/\nu$ is found, contradicting the discrete-logarithm assumption in $\mathbb{G}_t$. Hence $\mu \neq 0$ and $C_{f_1} = \mu^{-1}\delta\, G - \mu^{-1}\nu\, H$ opens to value $x := \mu^{-1}\delta$; since $U_1 = \delta G \neq \mathcal{O}$ we have $\delta \neq 0$, so $x \neq 0$. By binding this opening agrees (e.w.n.p.) with that of Step 1, so $b_x - a_x = x \neq 0$, i.e. $A \neq \pm B$.

Conclusion. The extractor outputs $(a_x, a_y, b_x, b_y, t_x, t_y, r_1, \ldots, r_6)$, valid openings of $C_1, \ldots, C_6$ satisfying equations (5); therefore $A + B = T$, with $A \neq \pm B$ (Step 5) and $A, B \neq \mathcal{O}$ (affine coordinates), so $(\mathsf{x}, \mathsf{w}) \in \mathcal{R}_{\mathsf{PA}}$. The only non-algebraic step invokes the discrete-logarithm/binding assumption (Step 5), exactly as in [21, Thm 4] and Theorem 6.

Honest Verifier Zero Knowledge. We construct a simulator $\mathcal{S}$ for Π_{PA} as follows: on input $\mathsf{x} = (C_1, \ldots, C_6)$ and challenge c, the simulator recomputes the derived commitments $C_{f_1}, C_{p_1}, C_{p_2}, C_{f_3}, C_{p_3}$ from x and samples

$$C_\tau \leftarrow \mathbb{G}_t, \quad \gamma \leftarrow \mathbb{F}^*,$$
$$(z_\tau, z_{r_\tau}, z_{f_1}, z_{r_{f_1}}, z_{e_1}, z_{e_2}, z_{f_3}, z_{r_{f_3}},$$
$$z_{e_3}, z_2, z_{r_2}, v_1, v_2, v_3) \leftarrow \mathbb{F}^{14}.$$

It sets $U_1 = \gamma G$ and computes the remaining first-message elements from the verification equations:

$$T_1 = z_\tau G + z_{r_\tau} H - cC_\tau, \qquad T_2 = z_{f_1} G + z_{r_{f_1}} H - cC_{f_1},$$
$$T_3 = z_\tau C_{f_1} + z_{e_1} H - cC_{p_1}, \quad T_4 = z_\tau C_\tau + z_{e_2} H - cC_{p_2},$$
$$T_5 = z_{f_3} G + z_{r_{f_3}} H - cC_{f_3}, \quad T_6 = z_{f_3} C_\tau + z_{e_3} H - cC_{p_3},$$
$$T_7 = z_2 G + z_{r_2} H - cC_2, \qquad U_2 = v_1 C_{f_1} + v_2 H - cU_1,$$
$$U_3 = v_3 G - cU_1.$$

The simulator outputs the transcript with first message $(C_\tau, T_1, \ldots, T_7, U_1, U_2, U_3)$, challenge c, and response resp. Checks (1)–(7), (9) and (10) hold by construction, and check (8) holds since $U_1 = \gamma G \neq \mathcal{O}$ as $\gamma \neq 0$.

We show that, for every $(\mathsf{x}, \mathsf{w}) \in \mathcal{R}_{\mathsf{PA}}$ and every challenge c, the real and simulated transcripts are identically distributed. As for Π_{Sq}, both transcripts are determined solely by the random coins of the prover (resp. the simulator): once C_τ, U_1 and resp are fixed, each T_i is determined by checks (1)–(7) and U_2, U_3 by checks $(9),(10)$. The simulator's coins are

$$\big(C_\tau, \gamma,\ z_\tau, z_{r_\tau}, z_{f_1}, z_{r_{f_1}}, z_{e_1}, z_{e_2}, z_{f_3}, z_{r_{f_3}}, z_{e_3},$$
$$z_2, z_{r_2}, v_1, v_2, v_3\big) \in \mathbb{G}_t \times \mathbb{F}^* \times \mathbb{F}^{14},$$

and the prover's coins are

$$\big(\beta_1, r_\tau,\ a_\tau, a_{r_\tau}, a_{f_1}, a_{r_{f_1}}, a_{e_1}, a_{e_2}, a_{f_3}, a_{r_{f_3}}, a_{e_3},$$
$$a_2, a_{r_2},\ \beta_2, \beta_3, \beta_4\big) \in \mathbb{F}^* \times \mathbb{F}^{15}.$$

Consider the mapping from the prover's coins to the simulator's coins given by

$$C_\tau = \tau G + r_\tau H, \qquad \gamma = \beta_1(b_x - a_x),$$

together with linear maps corresponding to the honest response definitions $z_\tau = a_\tau + c\tau$, $z_{r_\tau} = a_{r_\tau} + cr_\tau$, $\ldots$, $z_2 = a_2 + ca_y$, $z_{r_2} = a_{r_2} + cr_2$, $v_1 = \beta_2 + c\beta_1$, $v_2 = \beta_3 - c\beta_1(r_3 - r_1)$, $v_3 = \beta_4 + c\beta_1(b_x - a_x)$. Since H generates $\mathbb{G}_t$, the map $r_\tau \mapsto C_\tau$ is a bijection $\mathbb{F} \to \mathbb{G}_t$ for the fixed τ; since $b_x - a_x \neq 0$, the map $\beta_1 \mapsto \gamma$ is a bijection $\mathbb{F}^* \to \mathbb{F}^*$; and each remaining coordinate is an additive bijection of $\mathbb{F}$ in its own mask (the coins r_τ, β_1 that recur across several responses are fixed first, after which each response is shifted by a distinct fresh uniformly at random value). Hence the mapping is a bijection. Both

coin vectors are uniformly distributed and the bijection is measure-preserving, so for every instance-challenge pair the honest prover and the simulator produce a given transcript with equal probability. When Π_{PA} is composed in Π_{SM} and check (7) is dropped (Optimisation 3), the coins a_2, a_{r_2}, the responses z_2, z_{r_2} and T_7 are removed and the argument restricts to the remaining coordinates unchanged. □

5. Evaluation

In this section, we discuss implementation details and experimental evaluation of our three protocols.

5.1. Implementation

We provide an open-source Rust implementation of all three protocols in [25]. The experiments can be reproduced using the accompanying artifact archived on Zenodo [44]. All constructions are implemented as small modular components linked together via the reduction of knowledge framework. To this end, we implement a separate Rust crate that defines types and the composition operations for reductions of knowledge.

There are two common parameters in our constructions: the device's ECDSA curve and the credential signature scheme. We instantiate the former as Secp256r1 (NIST's P-256), as used in standard ECDSA deployments. For the credential scheme – the message space of BBS signatures and of the credential-layer commitments – we use BLS12-381, the curve commonly used in the academic BBS literature. The Tom curve, used in PoP-BP and PoP-Σ to enable native arithmetic over the base field of Secp256r1, is instantiated as T-256 [20], whose scalar field coincides with that base field. We set the CSchnorr length to $\lambda = 128$ in all constructions; the equality-across-groups protocol of [19] used in PoP-BP and PoP-Σ is instantiated at $\lambda = 112$. We emphasise that these choices reflect our concrete instantiation: the constructions themselves are agnostic to the specific curves and apply to any suitable choice of ECDSA or BLS curves, and (for PoP-BP and PoP-Σ) a chain-compatible curve. We describe each implementation in turn.

PoP-BP. For the circuit part of the proof $\Pi_{\mathcal{C},\mathsf{P}}$, we use the Bulletproofs-based proving system of [45], which we extend to support committed inputs. Combined with the optimizations of the CSchnorr reduction, the resulting circuit has $760\,\mathrm{R1CS}$ constraints on average. The circuit depends on the statement, but as discussed in Sec. 1.2, this incurs no additional trust assumptions.

PoP-PLONK. For the circuit part of the proof $\Pi_{\mathcal{C},\mathsf{P}}$ we use the plonk-based implementation of [46] as the proving system, which provides support for highly performant emulation of P-256 over BLS12-381 using the techniques of [24]. We do not specialise the circuit to a specific CSchnorr challenge: this would simplify the constraint system, but it is offset by the cost of per-statement circuit setup, which dominates and harms the verifier. A single a priori fixed circuit is therefore used for all statements. The resulting circuit has $12\,457\,\mathrm{rows}$, $9\,\mathrm{advice_columns}$, and $39\,\mathrm{fixed_columns}$.

PoP-Σ. We extend the CDLS proof system of [21], [30] with several optimisations at the protocol level as noted in Section 4.2. Together, said optimisations reduce the PoP-Σ proof size by approximately 29% (from $172\,\mathrm{kB}$ to $122\,\mathrm{kB}$ at $\lambda = 128$) and prover time by approximately 24%, with comparable verifier time. At the implementation level, we based our implementation on [47], and further apply standard micro-optimisations: batched Montgomery inversions for projective-to-affine conversions across rounds and shared computations across the per-equation verifier checks. These contribute small but consistent improvements to verifier time. We report a breakdown of this protocol's operations in Table 2, measured single-threaded for clarity.

5.2. Experimental Analysis

A core motivation of our work is credential presentation on *consumer* devices. To assess practical reach, we benchmark across three classes of commodity user hardware spanning the installed base, on both x86-64 and ARM64 architectures. Our Rust implementation uses no architecture-specific assembly or SIMD tuning, so the numbers reflect raw CPU capability. The devices used are:

Old. A 2017 MacBook Pro (2.3 GHz dual-core Intel Core i5, $8\,\mathrm{GB}$, x86-64), representative of devices several generations old still in widespread use.

Medium. An Intel Core i5-1345U laptop ($16\,\mathrm{GB}$, no dedicated GPU), a mid-range commodity laptop.

New. A 2024 MacBook Air (Apple M3, $24\,\mathrm{GB}$ unified memory, arm64), representative of recent hardware.

We present the end-to-end benchmark results (bandwidth and timings), which correspond to averages of 100 executions, in Table 1 (multi-threaded) and Table 3 (single-threaded). The benchmarks for PoP-BP, PoP-Σ, and PoP-PLONK correspond to Claims 1, 2, and 3, respectively, in the accompanying artifact [44]. All benchmarks were run with multi-threading enabled.

Comparison with Prior Work. Prior approaches to anonymous credentials with legacy device binding are *monolithic*: a single construction handles (1) ECDSA credential verification, (2) parsing, (3) attribute checks, and (4) device binding, making them hard to analyse and to upgrade. Our work instead confines ECDSA to device binding, yielding a fundamentally simpler statement that can be combined in a modular way with any pairing-based anonymous credential scheme. This modularity, however, limits direct comparison: prior systems report aggregate costs over all four sub-goals, so the device-binding overhead our work isolates cannot be separated out. We nevertheless compare along two axes: (i) against monolithic systems for anonymous credentials over legacy ECDSA, and (ii) against the CDLS protocol [21] underlying Ubique [17].

TABLE 1: Average efficiency measures over 100 executions across three device classes with multi-threading on. Proof size is device-independent.

	Old		Medium		New		
Construction	$\mathcal{P}$ (ms)	$\mathcal{V}$ (ms)	$\mathcal{P}$ (ms)	$\mathcal{V}$ (ms)	$\mathcal{P}$ (ms)	$\mathcal{V}$ (ms)	Proof (kB)
PoP-BP	1247.90	219.31	330.68	52.40	208.20	35.70	1.47
PoP-PLONK	3621.10	17.46	1324.90	9.01	696.27	6.42	4.28
PoP-Σ	283.04	277.98	122.82	114.87	67.58	62.86	122.11

The end-to-end performance of the circuit-based constructions (PoP-PLONK, PoP-BP) and the Schnorr-based construction (PoP-Σ). The results reported are the average of 100 iterations across old, medium and new devices (see Section 5.2). The best results per device are highlighted in green.

TABLE 2: Per-layer single-threaded timing on the New device of PoP-Σ and Ubique's CDLS work [17]. Averages over 100 executions.

(a) Our PoP-Σ

Layer	Prover (ms)	Verifier (ms)
C. transfer (BLS $\rightarrow$ T-256)	4.17	3.56
PoK ($\mathcal{R}_{\mathsf{SM}}$ and $\mathcal{R}_{\mathsf{PA}}$)	301.06	287.15
Total proof size: 122 kB		

(b) Ubique

Layer	Prover (ms)	Verifier (ms)
C. transfer (BLS $\rightarrow$ T-256)	4.17	3.56
PoK	455.76	312.40
Total proof size: 179 kB		

For (i), we consider Longfellow [12], Crescent [33], and Vega [13], which prove a *full* credential presentation including CBOR parsing and SHA-256 hashing in addition to device binding. Absent an open-source implementation, Vega's numbers are taken as published [13]; the others run on the New device. Vega's best variant achieves 237 ms online proving (plus 390 ms precomputation) at 108 kB for a 1920 B mDL, albeit on a 16-vCPU server. Longfellow runs in 483 ms with 325 kB for a CBOR credential, and Crescent in 303 ms with 15.08 kB for a JWT credential. Crescent attains compact proofs by relying on Groth16 [48]. Groth16, however, needs a circuit-specific trusted setup – difficult to organise and problematic in regulated settings where every component must be standardised. For our size-optimized variant we therefore build on Bulletproofs instead. Of the three, both Crescent and Longfellow expose device binding separately: under 286.70 ms for the first; and under 22.42 ms on the New device for the second. The latter, however, only considers a "witnessed" public key and is therefore unclear how to efficiently "bridge" the public key with a committed one that emerges via committed disclosure of an efficient multi-message signature scheme. Achieving such an efficient bridging is an interesting open problem that will result in a prover-optimized modular construction via a lightweight adaptation of [12].

Across both axes, our device-binding proofs (Table 1) match or improve on the proving times while drastically reducing proof size, down to 1.47 kB for PoP-BP. The remaining credential logic can be handled by the BBS layer at negligible cost, composed through the committed-disclosure interface of [22].

We stress, however, that competing on raw performance is not the goal of this work. Our aim is to demonstrate a structurally simpler and more flexible route to device-bound anonymous credentials: rather than arithmetising an entire credential in one monolithic circuit, we isolate device binding as a small, fixed sub-problem – provable, in our most conservative variant, without any circuit at all. Beyond the efficiency gains above, this yields a simpler code base, fewer opportunities for implementation bugs, and a more tractable target for standardisation.

6. Discussion & Conclusion

We presented three constructions for device-binding of anonymous credentials to legacy ECDSA secure elements, achieving proof sizes from approximately 1.5 kB to 122 kB with proving times consistently under 500 ms for newer devices. The resulting constructions can be combined in a plug-and-play manner with any native pairing-based credential scheme, e.g., based on BBS, allowing deployments to select the most appropriate variant depending on the use case and regulatory requirements. As such native schemes only add marginal costs, the combination of BBS-credentials with one of our two circuit-based constructions, PoP-BP and PoP-PLONK, improves proof size by two orders of magnitude over existing practical approaches. This gain stems from adopting the modular framework of [22], which isolates device binding from the credential system and exposes a much simpler ECDSA statement amenable to dedicated optimisations.

Extensions & Open Problems. We explore two further constructions in Appendix D. The first, building on Desmoulins et al. [49], produces signatures under a randomized device public key, so that only correct key randomization – rather than the full ECDSA relation – needs to be proven. While this does not yield significant efficiency gains over our main constructions, it has an interesting structural advantage: the proof is independent of the verifier's nonce and can therefore be precomputed offline, opening the possibility of trading precomputation time for smaller proof

TABLE 3: Average efficiency measures over 100 executions across three device classes with single-threading on. Proof size is device-independent.

	Old		Medium		New		
Construction	$\mathcal{P}$ (ms)	$\mathcal{V}$ (ms)	$\mathcal{P}$ (ms)	$\mathcal{V}$ (ms)	$\mathcal{P}$ (ms)	$\mathcal{V}$ (ms)	Proof (kB)
PoP-BP	1375.40	223.63	673.84	105.40	647.96	100.58	1.47
PoP-PLONK	3960.20	17.78	2513.30	9.57	1515.5	6.42	4.28
PoP-Σ	479.42	465.91	351.90	332.66	285.12	321.97	122.11

The end-to-end performance of the circuit-based constructions (PoP-PLONK, PoP-BP) and the Schnorr-based construction (PoP-Σ). The results reported are the average of 100 iterations across old, medium and new devices (see Section 5.2) with single-threading on. The best results per device are highlighted in green.

sizes. However, the approach requires raw ECDSA signing support from the secure element – a capability currently being deprecated on major platforms [50] – and its security model warrants further analysis.

The second extension addresses the threat of subverted secure elements, recently studied in [8]. Our constructions reveal part of the ECDSA signature, which is safe when the secure element behaves honestly but can break unlinkability if the element is compromised. By keeping the entire signature hidden, one obtains a proof of possession that remains unlinkable even against a subverted device, at the cost of a more complex circuit involving two full scalar multiplications. Adapting our optimisation techniques – in particular the Committed Schnorr protocol – to this setting remains an interesting open problem.

Modularity & Further Improvements. A central contribution beyond the concrete efficiency gains is the demonstration that the *reductions of knowledge* framework is a powerful tool for cryptographic engineering. By decomposing the device-binding proof into a chain of self-contained reductions, each step can be independently analysed, optimised, and replaced without invalidating the overall security. Future improvements to individual components – such as more efficient scalar multiplication proofs, tighter foreign-field emulation, or faster commitment transfer – can be integrated without redesigning the protocol.

Acknowledgements

We thank Miguel Ambrona and Inigo Querejeta-Azurmendi for implementing the foreign field arithmetic circuit and the P-256 emulation parameters over BLS12-381. We thank Greg Zaverucha for suggesting using the lightweight Bulletproofs over T-256 for device binding and Christian Mouchet for usefull feedback on an earlier draft of this work. This research was partially funded by SPRIND (the German Federal Agency for Breakthrough Innovation) and Common Codes. The authors are solely responsible for the content of this publication; the positions presented here do not reflect the views of SPRIND or Common Codes.

References

[1] T. Looker, V. Kalos, A. Whitehead, and M. Lodder, "The BBS Signature Scheme," Internet-Draft draft-irtf-cfrg-bbs-signatures-09, Internet Engineering Task Force, July 2025. Work in Progress.

[2] "Iso/iec wd 24843: Information security - attribute-based credentials," Working Draft ISO/IEC WD 24843, International Organization for Standardization, 2025. Under development. A working group has prepared a draft.

[3] Electronic Signatures and Trust Infrastructures (ESI), "ETSI TR 119 476 V1.2.1: Analysis of selective disclosure and zero-knowledge proofs applied to Electronic Attestation of Attributes," 2024.

[4] European Commission, "Eu digital identity wallet – architecture and reference framework." https://eudi.dev/2.4.0/, 2026. Accessed: 2026-04-27.

[5] "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." https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32014R0910, July 2014. Accessed: 2026-04-27.

[6] E. F. Brickell, J. Camenisch, and L. Chen, "Direct anonymous attestation," in *ACM CCS 2004* (V. Atluri, B. Pfitzmann, and P. McDaniel, eds.), pp. 132–145, ACM Press, Oct. 2004.

[7] E. Brickell, L. Chen, and J. Li, "A new direct anonymous attestation scheme from bilinear maps," in *Trusted Computing - Challenges and Applications (TRUST 2008)*, Lecture Notes in Computer Science, Springer, 2008.

[8] K. Friedrichs, F. Harding, A. Lehmann, and A. Lysyanskaya, "Device-bound anonymous credentials with(out) trusted hardware." Cryptology ePrint Archive, Report 2025/1995, 2025.

[9] Trusted Computing Group, "TPM 2.0 Library Specification." https://trustedcomputinggroup.org/resource/tpm-library-specification/, 2013. Accessed: 2026-04-27.

[10] P. D. Rosa, "Discussion comment on Cryptographers' Feedback on the EU Digital Identity's ARF #211." github.com, 2024.

[11] M. Rosenberg, J. D. White, C. Garman, and I. Miers, "zk-creds: Flexible anonymous credentials from zkSNARKs and existing identity infrastructure," in *2023 IEEE Symposium on Security and Privacy*, pp. 790–808, IEEE Computer Society Press, May 2023.

[12] M. Frigo and a. shelat, "Anonymous credentials from ECDSA." Cryptology ePrint Archive, Report 2024/2010, 2024.

[13] D. Kaviani and S. Setty, "Vega: Low-latency zero-knowledge proofs over existing credentials." Cryptology ePrint Archive, Report 2025/2094, 2025.

[14] L. Eagen, H. Ngo, V. Rushi, Y. Tong, M. Tsai, and J. Xia, "Openac: Open design for transparent and lightweight anonymous credentials," *IACR Cryptol. ePrint Arch.*, vol. 2026, p. 251, 2026.

[15] ZKProof.org, "Zkproof charter." https://docs.zkproof.org/general, 2018.

[16] 0xPARC, "ZK Bug Tracker." https://github.com/0xPARC/zk-bug-tracker, 2024. A community-maintained collection of bugs, vulnerabilities, and exploits in applications using zero-knowledge cryptography. Accessed 27 April 2026.

[17] P. Amrein. https://github.com/UbiqueInnovation/zkattest-rs, 2025.

[18] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *CRYPTO 2004* (M. Franklin, ed.), vol. 3152 of *LNCS*, pp. 41–55, Springer, Berlin, Heidelberg, Aug. 2004.

[19] M. Orrù, G. Kadianakis, M. Maller, and G. Zaverucha, "Beyond the circuit: How to minimize foreign arithmetic in ZKP circuits," *CiC*, vol. 2, no. 1, p. 23, 2025.

[20] A. Faz-Hernández, W. Ladd, and D. Maram, "ZkAttest: Ring and group signatures on top of existing ECDSA keys." Cryptology ePrint Archive, Report 2021/1183, 2021.

[21] S. Celi, S. Levin, and J. Rowell, "CDLS: Proving knowledge of committed discrete logarithms with soundness," in *AFRICACRYPT 24* (S. Vaudenay and C. Petit, eds.), vol. 14861 of *LNCS*, pp. 69–93, Springer, Cham, July 2024.

[22] A. Lehmann, A. Sidorenko, and A. Zacharakis, "Vision: A modular framework for anonymous credential systems." Cryptology ePrint Archive, Report 2025/1981, 2025.

[23] A. Kothapalli and B. Parno, "Algebraic reductions of knowledge," in *CRYPTO 2023, Part IV* (H. Handschuh and A. Lysyanskaya, eds.), vol. 14084 of *LNCS*, pp. 669–701, Springer, Cham, Aug. 2023.

[24] M. Ambrona, D. Firsov, and I. Querejeta-Azurmendi, "Efficient foreign-field arithmetic in PLONK." Cryptology ePrint Archive, Report 2025/695, 2025.

[25] "ecdsa_pops." https://github.com/hpicrypto/ecdsa_pops, 2026.

[26] A. Kothapalli, *A Theory of Composition for Proofs of Knowledge*. PhD thesis, Carnegie Mellon University, USA, 2024.

[27] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *2018 IEEE Symposium on Security and Privacy*, pp. 315–334, IEEE Computer Society Press, May 2018.

[28] A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in *ASIACRYPT 2010* (M. Abe, ed.), vol. 6477 of *LNCS*, pp. 177–194, Springer, Berlin, Heidelberg, Dec. 2010.

[29] A. Gabizon, Z. J. Williamson, and O. Ciobotaru, "PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge." Cryptology ePrint Archive, Report 2019/953, 2019.

[30] L. Harchandani. https://github.com/docknetwork/crypto/tree/main/equality_across_groups, 2025.

[31] C. for Research on Cryptography and Security, "P-256." https://std.neuromancer.sk/nist/P-256/. Accessed 2026-04-16.

[32] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *CRYPTO'86* (A. M. Odlyzko, ed.), vol. 263 of *LNCS*, pp. 186–194, Springer, Berlin, Heidelberg, Aug. 1987.

[33] C. Paquin, G.-V. Policharla, and G. Zaverucha, "Crescent: Stronger privacy for existing credentials." Cryptology ePrint Archive, Report 2024/2013, 2024.

[34] B. Bünz, *Improving the Privacy, Scalability, and Ecological Impact of Blockchains*. Doctor of philosophy dissertation, Stanford University, June 2023.

[35] G. Segev, "Bulletproofs for R1CS: Bridging the completeness-soundness gap and a ZK extension." Cryptology ePrint Archive, Report 2025/327, 2025.

[36] J. Bootle, A. Cerulli, P. Chaidos, J. Groth, and C. Petit, "Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting," in *EUROCRYPT 2016, Part II* (M. Fischlin and J.-S. Coron, eds.), vol. 9666 of *LNCS*, pp. 327–357, Springer, Berlin, Heidelberg, May 2016.

[37] A. Gabizon and Z. J. Williamson, "The turbo-plonk program syntax for specifying snark programs," 2019. Preprint.

[38] A. Gabizon and Z. J. Williamson, "plookup: A simplified polynomial protocol for lookup tables." Cryptology ePrint Archive, Report 2020/315, 2020.

[39] M. Ambrona, A.-L. Schmitt, R. R. Toledo, and D. Willems, "New optimization techniques for PlonK's arithmetization." Cryptology ePrint Archive, Report 2022/462, 2022.

[40] M. Orru, L. Tulloch, V. Snyder-Graf, and I. Goldberg, "sigma-rs: A modular approach for keyed-verification anonymous credentials." Cryptology ePrint Archive, Paper 2026/794, 2026.

[41] G. Fuchsbauer, E. Kiltz, and J. Loss, "The algebraic group model and its applications," in *CRYPTO 2018, Part II* (H. Shacham and A. Boldyreva, eds.), vol. 10992 of *LNCS*, pp. 33–62, Springer, Cham, Aug. 2018.

[42] J. Groth, M. Kohlweiss, M. Maller, S. Meiklejohn, and I. Miers, "Updatable and universal common reference strings with applications to zk-SNARKs," in *CRYPTO 2018, Part III* (H. Shacham and A. Boldyreva, eds.), vol. 10993 of *LNCS*, pp. 698–728, Springer, Cham, Aug. 2018.

[43] Ethereum Foundation, "Kzg summoning ceremony," 2023. Accessed: 2026-04-21.

[44] S. Celi, A. Lehmann, S. Levin, and A. Zacharakis, "Device binding for anonymous credentials on legacy phones." https://doi.org/10.5281/zenodo.22807192, 2026. Version 1.0.

[45] Greg Zaverucha, "signature-proof." https://github.com/zaverucha/signature-proof, 2026. accessed 2026-04-22.

[46] midnightntwrk, "midnight-zk: proofs." https://github.com/midnightntwrk/midnight-zk/tree/next/proofs, 2026. accessed 2026-04-21.

[47] J. R. Sofia Celi, Shai Levin, "CDLS: A Rust implementation." https://github.com/brave-experiments/CDLS, 2024.

[48] J. Groth, "On the size of pairing-based non-interactive arguments," in *EUROCRYPT 2016, Part II* (M. Fischlin and J.-S. Coron, eds.), vol. 9666 of *LNCS*, pp. 305–326, Springer, Berlin, Heidelberg, May 2016.

[49] N. Desmoulins, A. Dumanois, S. Kane, and J. Traoré, "Making BBS anonymous credentials eIDAS 2.0 compliant." Cryptology ePrint Archive, Report 2025/619, 2025.

[50] Android Open Source Project, "Features." https://source.android.com/docs/security/features/keystore/features. Last updated 2026-04-10 UTC; accessed 2026-04-16.

[51] T. Attema, S. Fehr, and M. Klooß, "Fiat-Shamir transformation of multi-round interactive proofs," in *TCC 2022, Part I* (E. Kiltz and V. Vaikuntanathan, eds.), vol. 13747 of *LNCS*, pp. 113–142, Springer, Cham, Nov. 2022.

[52] J. Bootle, A. Chiesa, and K. Sotiraki, "Sumcheck arguments and their applications," in *CRYPTO 2021, Part I* (T. Malkin and C. Peikert, eds.), vol. 12825 of *LNCS*, (Virtual Event), pp. 742–773, Springer, Cham, Aug. 2021.

[53] M. Bellare and T. Kohno, "A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications," in *EUROCRYPT 2003* (E. Biham, ed.), vol. 2656 of *LNCS*, pp. 491–506, Springer, Berlin, Heidelberg, May 2003.

Appendix A. Reductions of Knowledge

A reduction of knowledge (RoK) $\Pi : \mathcal{R} \to \mathcal{R}'$ is an interactive protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ that reduces knowledge of w s.t. $(pp, \mathsf{x}, \mathsf{w}) \in \mathcal{R}$ to knowledge of w' s.t. $(pp, \mathsf{x}', \mathsf{w}') \in \mathcal{R}'$, where x', w' are the respective

outputs of $\mathcal{P}$ and $\mathcal{V}$ after their interaction. We next recall the definition of [23], [26][14].

Definition 1 (Reductions of Knowledge)**.** Let $\mathcal{R}, \mathcal{R}'$ be ternary relations taking as input public parameters, a statement and a witness. A RoK $\Pi : \mathcal{R} \to \mathcal{R}'$ is a tuple $(\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ where

- $pp \leftarrow \mathsf{Gen}(1^\lambda)$ is a PPT algorithm that on input the security parameter outputs parameters pp.
- $(\mathsf{w}', \mathsf{x}') \leftarrow \langle \mathcal{P}(pp, \mathsf{x}, \mathsf{w}), \mathcal{V}(pp, \mathsf{x}) \rangle$ is an interactive protocol between PPT (interactive) algorithms $\mathcal{P}, \mathcal{V}$, where $\mathcal{P}$ takes as input the parameters pp, a statement x and a witness w and after the interaction outputs a fresh witness w', and $\mathcal{V}$ takes as input the parameters pp and a statement x and after the interaction outputs statement x',

that satisfies the following properties:

Completeness. For all $\lambda \in \mathbb{N}$ and PPT algorithms $\mathcal{A}$:

$$\Pr\left[\begin{array}{c|c} (pp, \mathsf{x}, \mathsf{w}) \in \mathcal{R} & pp \leftarrow \mathsf{Gen}(1^\lambda) \\ \wedge & (\mathsf{x}, \mathsf{w}) \leftarrow \mathcal{A}(pp) \\ (pp, \mathsf{x}', \mathsf{w}') \notin \mathcal{R}') & (\mathsf{w}', \mathsf{x}') \leftarrow \langle \mathcal{P}(pp, \mathsf{x}, \mathsf{w}), \mathcal{V}(pp, \mathsf{x}) \rangle \end{array}\right]$$

is zero.

Knowledge Soundness. For all $\lambda \in \mathbb{N}$ and all expected polynomial time adversaries $\mathcal{P}^*, \mathcal{A}$ there exists a polynomial time extractor $\mathcal{E}$ such that

$$\Pr\left[\begin{array}{c|c} (pp, \mathsf{x}, \mathsf{w}) \in \mathcal{R} & pp \leftarrow \mathsf{Gen}(1^\lambda) \\ & (\mathsf{x}, \mathsf{st}) \leftarrow \mathcal{A}(pp) \\ & \mathsf{w} \leftarrow \mathcal{E}(pp, \mathsf{x}, \mathsf{st}) \end{array}\right] + \mathsf{negl}(\lambda) \geq$$

$$\Pr\left[\begin{array}{c|c} (pp, \mathsf{x}', \mathsf{w}') \in \mathcal{R}' & pp \leftarrow \mathsf{Gen}(1^\lambda) \\ & (\mathsf{x}, \mathsf{st}) \leftarrow \mathcal{A}(pp) \\ & (\mathsf{w}', \mathsf{x}') \leftarrow \langle \mathcal{P}^*(pp, \mathsf{x}, \mathsf{st}), \mathcal{V}(pp, \mathsf{x}) \rangle \end{array}\right]$$

Public Reducibility. For all $\lambda \in \mathbb{N}$, there exists a polynomially computable function ϕ such that for all polynomial time adversaries $\mathcal{P}^*, \mathcal{A}$:

$$\Pr\left[\begin{array}{c|c} \phi(pp, \mathsf{x}, \mathsf{tr}) \neq \mathsf{x}' & pp \leftarrow \mathsf{Gen}(1^\lambda) \\ & (\mathsf{x}, \mathsf{st}) \leftarrow \mathcal{A}(pp) \\ & (\mathsf{w}', \mathsf{x}') \leftarrow \langle \mathcal{P}^*(pp, \mathsf{x}, \mathsf{st}), \mathcal{V}(pp, \mathsf{x}) \rangle \\ & \mathsf{tr} \leftarrow \mathsf{tr}\left(\langle \mathcal{P}^*(pp, \mathsf{x}, \mathsf{st}), \mathcal{V}(pp, \mathsf{x}) \rangle\right) \end{array}\right]$$

is zero, where $\mathsf{tr}(\cdot)$ denotes the transcript of the interaction of $\mathcal{P}^*, \mathcal{V}$ with their respective inputs.

Honest Verifier Zero Knowledge (HVZK). For all $\lambda \in \mathbb{N}$ there exists a PPT algorithm $\mathcal{S}$ s.t. for all $pp \leftarrow \mathsf{Gen}(1^\lambda)$ and all x, w that satisfy $(pp, \mathsf{x}, \mathsf{w}) \in \mathcal{R}$, the distributions

$$\left\{ \ \mathsf{tr} \ \middle| \ \mathsf{tr} \leftarrow \mathcal{S}(pp, \mathsf{x}) \ \right\}$$

$$\left\{ \ \mathsf{tr} \ \middle| \ \mathsf{tr} \leftarrow \mathsf{tr}\left(\langle \mathcal{P}(pp, \mathsf{x}, \mathsf{w}), \mathcal{V}(pp, \mathsf{x}) \rangle\right) \ \right\}$$

are statistically close, where $\mathsf{tr}(\cdot)$ denotes the transcript of the interaction of $\mathcal{P}, \mathcal{V}$ with their respective inputs.

We write $\Pi : \mathcal{R} \to \mathcal{R}'$ to denote that Π is a RoK from relation $\mathcal{R}$ to relation $\mathcal{R}'$.

If the messages sent by $\mathcal{V}$ are random messages, independent of the interaction, we say that the RoK is *public coin*. Such protocols can be transformed to non-interactive via the Fiat-Shamir transform [32], [51].

We also recall the results for abstracting knowledge soundness for a large family of algebraic public coin protocols. These results are taken verbatim from [23], which in turn is an adaptation of the results of [52].

Definition 2 (Tree of accepting transcripts)**.** Consider an m-round public-coin interactive protocol $(\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ that satisfies the interface described in Definition 1. A $(n_1, \ldots, n_m)$-tree of accepting transcripts for statement x is a tree of depth m where each node at layer i has n_i outgoing edges such that (1) each node in layer $i \in [m]$ is labeled with $\mathcal{P}$'s message for round i; (2) each outgoing edge from layer $i \in [m]$ is labeled with a different choice of $\mathcal{V}$ randomness for round i; (3) each leaf is labeled with an accepting statement-witness pair output by $\mathcal{P}$ and $\mathcal{V}$ corresponding to the interaction along the path.

Lemma 1. *Consider an m-round public-coin interactive protocol $(\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ that satisfies the interface described in Definition 1 and completeness. Then $(\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ is a RoK if there exists a PPT extractor χ that, for all instances x_1 outputs a satisfying witness w_1 with probability $1 - \mathsf{negl}(\lambda)$, given an $(n_1, \ldots, n_m)$-tree of accepting transcripts for x_1 where the $\mathcal{V}$'s randomness is sampled from space Q such that $|Q| = O(2^\lambda)$, and $\prod_i n_i = \mathsf{poly}(\lambda)$.*

We note that a special case of the above lemma is for Σ-protocols, which are (as per this definition), a 1-round public-coin protocol that satisfies 2-special soundness, i.e., it possesses an extractor taking a (2)-tree of accepting transcripts.

Composing reductions of knowledge. We now recall the results of [23], [26] about composition of RoKs. We first present the definition for a relation pair.

Definition 3. Let $\mathcal{R}_1, \mathcal{R}_2$ be ternary relations. We define $\mathcal{R}_1 \times \mathcal{R}_2 =$

$$\{(\mathsf{x}_1, \mathsf{w}_1), (\mathsf{x}_2, \mathsf{w}_2) \mid (\mathsf{x}_1, \mathsf{w}_1) \in \mathcal{R}_1 \ \wedge \ (\mathsf{x}_2, \mathsf{w}_2) \in \mathcal{R}_2\}$$

We now define sequential composition for RoKs.

Definition 4. Let $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4$ be ternary relations.

1) Let $\Pi_1 : \mathcal{R}_1 \to \mathcal{R}_2 = (\mathsf{Gen}, \mathcal{P}_1, \mathcal{V}_1)$ and $\Pi_2 : \mathcal{R}_2 \to \mathcal{R}_3 = (\mathsf{Gen}, \mathcal{P}_2, \mathcal{V}_2)$ be RoKs. We define the sequential and parallel composition of Π_1, Π_2 denoted as $\Pi_s = \Pi_2 \circ \Pi_1$ as the tuple $(\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ where $\langle \mathcal{P}(pp, \mathsf{x}_1, \mathsf{w}_1), \mathcal{V}(pp, \mathsf{x}_1) \rangle$ is the following protocol:
   - $(\mathsf{w}_2, \mathsf{x}_2) \leftarrow \langle \mathcal{P}_1(pp, \mathsf{x}_1, \mathsf{w}_1), \mathcal{V}_1(pp, \mathsf{x}_1) \rangle$
   - $(\mathsf{w}_3, \mathsf{x}_3) \leftarrow \langle \mathcal{P}_2(pp, \mathsf{x}_2, \mathsf{w}_2), \mathcal{V}_2(pp, \mathsf{x}_2) \rangle$
   - $\mathcal{P}$ outputs w_3 and $\mathcal{V}$ outputs $(\mathsf{x}_3, \mathsf{x}_4)$.

14. We slightly modify the HVZK definition of [26] to capture the property against unbounded adversaries.

2) Let $\Pi_1 : \mathcal{R}_1 \to \mathcal{R}_2 = (\mathsf{Gen}, \mathcal{P}_1, \mathcal{V}_1)$ and $\Pi_2 : \mathcal{R}_3 \to \mathcal{R}_4 = (\mathsf{Gen}, \mathcal{P}_2, \mathcal{V}_2)$ be RoKs. We define the parallel composition of Π_1, Π_2 denoted as $\Pi_p = \Pi_1 \times \Pi_2$ the tuple $(\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ where $\langle \mathcal{P}(pp, (\mathsf{x}_1, \mathsf{x}_3), (\mathsf{w}_1, \mathsf{w}_3))), \mathcal{V}(pp, (\mathsf{x}_1, \mathsf{x}_3)) \rangle$ is the following protocol:
   - $(\mathsf{w}_2, \mathsf{x}_2) \leftarrow \langle \mathcal{P}_1(pp, \mathsf{x}_1, \mathsf{w}_1), \mathcal{V}_1(pp, \mathsf{x}_1) \rangle$
   - $(\mathsf{w}_4, \mathsf{x}_4) \leftarrow \langle \mathcal{P}_2(pp, \mathsf{x}_3, \mathsf{w}_3), \mathcal{V}_2(pp, \mathsf{x}_3) \rangle$
   - $\mathcal{P}$ outputs $(\mathsf{w}_2, \mathsf{w}_4)$ and $\mathcal{V}$ outputs $(\mathsf{x}_2, \mathsf{x}_4)$.

Theorem 8. *Let $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4$ be ternary relations.*

1) Let $\Pi_1 : \mathcal{R}_1 \to \mathcal{R}_2$ and $\Pi_2 : \mathcal{R}_2 \to \mathcal{R}_3$ be public coin RoKs. Then, the sequential composition $\Pi = \Pi_2 \circ \Pi_1$ is a RoK from $\mathcal{R}_1$ to $\mathcal{R}_3$.

2) Let $\Pi_1 : \mathcal{R}_1 \to \mathcal{R}_2$ and $\Pi_2 : \mathcal{R}_3 \to \mathcal{R}_4$ be public coin RoK. Then, the parallel composition $\Pi = \Pi_2 \times \Pi_1$ is a RoK from $\mathcal{R}_1 \times \mathcal{R}_3$ to $\mathcal{R}_2 \times \mathcal{R}_4$.

Proof. For completeness, knowledge soundness and public reducibility we refer the reader to [23, Thm. 5, 6]. We next present the case for HVZK.

1) Let $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ be ternary relations and $\Pi_1 : \mathcal{R}_1 \to \mathcal{R}_2$ and $\Pi_2 : \mathcal{R}_2 \to \mathcal{R}_3$ be HVZK public coin reductions of knowledge. Let $\mathcal{S}_1, \mathcal{S}_2$ be the HVZK simulators of Π_1, Π_2 respectively. Also let ϕ_1 be the polynomially computable function that can produce the reduced statement from the transcript for Π_1 (guaranteed to exist by the public reducibility property). We next describe the simulator $\mathcal{S}$ for $\Pi = \Pi_2 \circ \Pi_1$. The simulator $\mathcal{S}$ on input pp, x_1 proceeds as follows:

   $\underline{\mathcal{S}(pp, \mathsf{x}_1)}$
   $\mathsf{tr}_1 \leftarrow \mathcal{S}_1(pp, \mathsf{x}_1)$
   $\mathsf{x}_2 \leftarrow \phi_1(pp, \mathsf{x}_1, \mathsf{tr}_1)$
   $\mathsf{tr}_2 \leftarrow \mathcal{S}_2(pp, \mathsf{x}_2)$
   Output $\mathsf{tr} := \mathsf{tr}_1 \| \mathsf{tr}_2$

   By construction of $\Pi = \Pi_2 \circ \Pi_1$ we can view the messages exchanged by $\mathcal{P}$ and $\mathcal{V}$ as two parts: the execution of Π_1 and the execution of Π_2. Consider an actual transcript $\widetilde{\mathsf{tr}} = \widetilde{\mathsf{tr}_1} \| \widetilde{\mathsf{tr}_1}$ where $\widetilde{\mathsf{tr}_1}$, $\widetilde{\mathsf{tr}_2}$ correspond to the executions of Π_1, Π_2 respectively. By the HVZK property of Π_1, Π_2 we know that for all $\mathsf{x}_1, \mathsf{x}_2$ the distributions

$$\left\{ \ \mathsf{tr}_\mathsf{i} \mid \mathsf{tr}_\mathsf{i} \leftarrow \mathcal{S}_i(pp, \mathsf{x}_i) \ \right\}$$

$$\left\{ \ \widetilde{\mathsf{tr}_\mathsf{i}} \mid \widetilde{\mathsf{tr}_\mathsf{i}} \leftarrow \mathsf{tr} \left(\langle \mathcal{P}_i(pp, \mathsf{x}_i), \mathcal{V}_i(pp, \mathsf{x}_i) \rangle \right) \ \right\}$$

   are statistically close for $i \in \{1, 2\}$. Finally, the produced "intermediate" statement x_2 the simulator computes is statistically close to the one produced by an honest verifier during the execution of Π. Indeed, this follows from the fact that $\widetilde{\mathsf{tr}_1}$ and tr_1 are statistically close and the fact that $\mathsf{x}_2 := \phi_1(pp, \mathsf{x}_1, \mathsf{tr}_1)$. We therefore conclude that the distributions

$$\left\{ \ \mathsf{tr}_1 \| \mathsf{tr}_2 \mid \mathsf{tr}_1 \| \mathsf{tr}_2 \leftarrow \mathcal{S}_i(pp, \mathsf{x}_1) \ \right\}$$

$$\left\{ \ \widetilde{\mathsf{tr}_1} \| \widetilde{\mathsf{tr}_2} \mid \widetilde{\mathsf{tr}_1} \| \widetilde{\mathsf{tr}_2} \leftarrow \mathsf{tr} \left(\langle \mathcal{P}(pp, \mathsf{x}_1), \mathcal{V}(pp, \mathsf{x}_1) \rangle \right) \ \right\}$$

   are statistically close.

2) The proof is the same as in the first case except that we don't need to argue about an "intermediate" statement. Instead, the two produced transcripts in this case are independent.

□

Appendix B.
RoK for Transfering Committed Relations

We present the generic RoK $\Pi_{\mathsf{CS} \to \widetilde{\mathsf{CS}}}$ that allows transfering a committed relation over P w.r.t. some commitment scheme CS to the same relation under a different commitment scheme $\widetilde{\mathsf{CS}}$. The construction relies on a PoK Π_{EQ} to prove knowledge of two openings of a committed value under different commitment schemes. We present the construction in Figure 8.

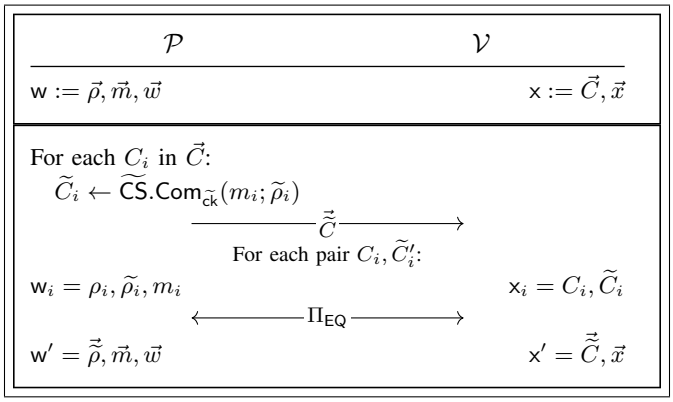


Figure 8: $\Pi_{\mathsf{CS} \to \widetilde{\mathsf{CS}}} : \mathcal{R}_{\mathsf{CS},\mathsf{P}} \to \mathcal{R}_{\widetilde{\mathsf{CS}},\mathsf{P}}$.

Theorem 9. *Let $\mathsf{CS}, \widetilde{\mathsf{CS}}$ be a perfectly hiding commitment schemes with the same message space and $\mathcal{M}$ and let $\Pi_{\mathsf{EQ}} : \mathcal{R}_{\mathsf{EQ}} \to \mathcal{R}_\top$ be a public coin HVZK PoK. Then, the protocol $\Pi_{\mathsf{CS} \to \widetilde{\mathsf{CS}}}$ described in Figure 8 is a public-coin HVZK RoK $\Pi_{\mathsf{CS} \to \widetilde{\mathsf{CS}}} : \mathcal{R}_{\mathsf{CS},\mathsf{P}} \to \mathcal{R}_{\widetilde{\mathsf{CS}},\mathsf{P}}$.*

Proof. Completeness and public reducibility follow by construction. We next describe the proof for knowledge soundness and HVZK

Knowledge Soundness. Let $\mathcal{A}$ and $\mathcal{P}^*$ be PPT adversaries as described in Definition 1. Let $\Pi^\ell_{\mathsf{EQ}} = \Pi_{\mathsf{EQ}} \times \ldots \times \Pi_{\mathsf{EQ}} : \mathcal{R}^\ell_{\mathsf{EQ}} \to \mathcal{R}_\top$ be the parallel composition capturing the second part of the protocol, where ℓ is the number of input commitments for $\mathcal{R}_{\mathsf{CS},\mathsf{P}}$. We first construct a pair of adversaries $\mathcal{A}_{\mathsf{EQ}}, \mathcal{P}^*_{\mathsf{EQ}}$ against Π^ℓ_{EQ} by $\mathcal{A}$ and $\mathcal{P}^*$. Let $\mathcal{P}^* = (\mathcal{P}^*_1, \mathcal{P}^*_2)$ where $\mathcal{P}^*_1$ computes the first message (the fresh commitments), and $\mathcal{P}^*_2$ computes the messages for the rest of the protocol.

$\underline{\mathcal{A}_{\mathsf{EQ}}(pp)}$
$\mathsf{x}, \mathsf{st} \leftarrow \mathcal{A}(pp)$
$\vec{\widetilde{C}} \leftarrow \mathcal{P}^*_1(\mathsf{x}, \mathsf{st}; r)$
Parse $\mathsf{x} := C_1, \ldots, C_\ell, \vec{x}$

Define $\mathsf{x}_{\mathsf{EQ}} := (\vec{\widetilde{C}}, \vec{C})$, $\mathsf{st}_{\mathsf{EQ}} := (\mathsf{st}, r)$
Output $\mathsf{x}_{\mathsf{EQ}}, \mathsf{st}_{\mathsf{EQ}}$

$\underline{\mathcal{P}^*_{\mathsf{EQ}}(pp, \mathsf{x}_{\mathsf{EQ}}, \mathsf{st}_{\mathsf{EQ}})}$
Parse $\mathsf{st}_{\mathsf{EQ}} := (\mathsf{st}, r)$
Interact with $\mathcal{V}$ as follows:

- Start an execution of $\mathcal{P}^*(\mathsf{x}, \mathsf{st}; r)$ until outputting the commitments $\vec{\widetilde{C}}$
- Use $\mathcal{P}^*$ with the same state to reply all the next messages

By knowledge soundness of Π_{EQ} and by Theorem 8, Π^{ℓ}_{EQ} is also knowledge sound and there exists an extractor $\mathcal{E}_{\mathsf{EQ}}$ corresponding to the pair $(\mathcal{A}_{\mathsf{EQ}}, \mathcal{P}^*_{\mathsf{EQ}})$ that outputs a valid witness w_{EQ} for x_{EQ} with probability negligibly close to that of $(\mathcal{A}_{\mathsf{EQ}}, \mathcal{P}^*_{\mathsf{EQ}})$ making the $\mathcal{V}_{\mathsf{EQ}}$ verifier accept. We next define our extractor $\mathcal{E}$.

$\underline{\mathcal{E}(pp, \mathsf{x}, \mathsf{st})}$
$\mathsf{x}_{\mathsf{EQ}}, \mathsf{st}_{\mathsf{EQ}} \leftarrow \mathcal{A}_{\mathsf{EQ}}(pp)$
Parse $\mathsf{st}_{\mathsf{EQ}} := (\mathsf{st}, r)$
$\mathsf{w}', \mathsf{x}' \leftarrow \langle \mathcal{P}^*(pp, \mathsf{x}, \mathsf{st}; r), \mathcal{V}(pp, \mathsf{x}) \rangle$
$\mathsf{w}_{\mathsf{EQ}} \leftarrow \mathcal{E}_{\mathsf{EQ}}(\mathsf{x}_{\mathsf{EQ}}, \mathsf{st}_{\mathsf{EQ}})$
Parse $\mathsf{w}' := \vec{m}', \vec{\rho}', \vec{w}$
Parse $\mathsf{w}_{\mathsf{EQ}} := \vec{m}_{\mathsf{EQ}}, \vec{\rho}_{\mathsf{EQ}}, \vec{\widetilde{\rho}}_{\mathsf{EQ}}$
Output $\vec{m}_{\mathsf{EQ}}, \vec{\rho}_{\mathsf{EQ}}, \vec{w}$

Now, if $\mathcal{P}^*$ outputs a valid statement $(pp, \mathsf{x}', \mathsf{w}') \in \mathcal{R}_{\widetilde{\mathsf{CS}}, \mathsf{P}}$:

1) the values m'_i, ρ'_i satisfy $\widetilde{C_i} = \widetilde{\mathsf{CS}}.\mathsf{Com}_{\widetilde{ck}}(m'_i; \rho'_i)$, and
2) $\mathsf{P}(\vec{x}, \vec{m}', \vec{w})$ is true

Since the $\widetilde{\mathsf{GCS}}$ commitments in $\mathsf{x}', \mathsf{x}_{\mathsf{EQ}}$ are the same, the openings satisfy $(m_{\mathsf{EQ},i}, \widetilde{\rho}_{\mathsf{EQ},i}) = (m'_i, \rho'_i)$ except with negligible probability, by the binding property of $\widetilde{\mathsf{GCS}}$. Further, since the verifier $\mathcal{V}_{\mathsf{EQ}^{\ell}}$ is invoked by $\mathcal{V}$ by construction of $\Pi_{\mathsf{CS} \to \widetilde{\mathsf{CS}}}$ and it accepts, the witness output by $\mathcal{E}_{\mathsf{EQ}}$ must be (e.w.n.p.) a valid witness. Therefore (1) $\vec{m}' = \vec{m}_{\mathsf{EQ}}$ and (2) $C_i = \mathsf{CS}.\mathsf{Com}_{\mathsf{ck}}(m_{\mathsf{EQ}_i}; \rho_{\mathsf{EQ}})$ which in turn implies that $\mathsf{P}(\vec{x}, \vec{m}_{\mathsf{EQ}}, \vec{w})$ is true and therefore $\mathsf{w} = (\vec{m}_{\mathsf{EQ}}, \vec{\rho}_{\mathsf{EQ}}, \vec{w})$ is a valid witness for x.

Honest Verifier Zero Knowledge. We describe a simulator that produces transcripts that are statistically close to real ones. The simulator works as follows: it uses an arbitrary message $m \in \mathcal{M}$ and computes for each i a fresh commitment $\widetilde{C_i} \leftarrow \widetilde{\mathsf{CS}}_{\widetilde{\mathsf{ck}}}(m)$. Then it creates for each i the statement $\mathsf{x}_i = (C_i, \widetilde{C_i})$ and invokes the simulator of Π_{EQ} on input x_i to get a simulated transcript tr_i. It outputs the transcript

$$\mathsf{tr} := \left(\left\{ \widetilde{C_i}, \right\}_i, \left\{ \widetilde{\mathsf{tr}_i}, \right\}_i \right)$$

The simulated transcript tr is statistically close to a real transcript by the perfect hiding property of $\widetilde{\mathsf{CS}}$ and the HVZK property of Π_{EQ}.

□

Remark 3. The construction also captures proving equality of commitment under the same commitment scheme with different commitment keys.

Appendix C. Equality of Commitments

We present two constructions for proving knowlege of openings of committed values under different commitment schemes. The first one is a simple construction to show that multiple (plain) Pedersen commitments and a compact pedersen commitment have the same opening. The second is the construction of [19] for proving knowledge of the same opening of two Pedersen commitments over different groups.

C.1. Equality of Compact and Plain Pedersen Commitment Schemes

Let $\mathbb{G}$ be a group with scalar field $\mathbb{F}$ and $\mathsf{ck}_1 = (G, H)$ be a Pedersen commitment key and $\mathsf{ck}_2 = (\vec{G} \in \mathbb{G}^n, \vec{H} \in \mathbb{G}_k)$ be a compact Pedersen commitment scheme with k blinding factors. We define the relation:

$$\mathcal{R}_{\mathsf{PD}} = \left\{ \begin{matrix} pp := (\mathsf{ck}_1, \mathsf{ck}_2) \\ \mathsf{x} := \vec{C}, C, \\ \mathsf{w} := (\vec{m}, \vec{\rho}, \vec{\rho}_c) \end{matrix} \middle| \begin{matrix} C = \vec{m}^{\top} \vec{G} + \vec{\rho}_c^{\top} \vec{H} \wedge \\ \forall i \; C_i = m_i G + \rho_i H \end{matrix} \right\}$$

We present a simple public coin HVZK proof of knowledge for the above relation in Figure 9. The protocol is based on standard techniques and its security properties are well understood.

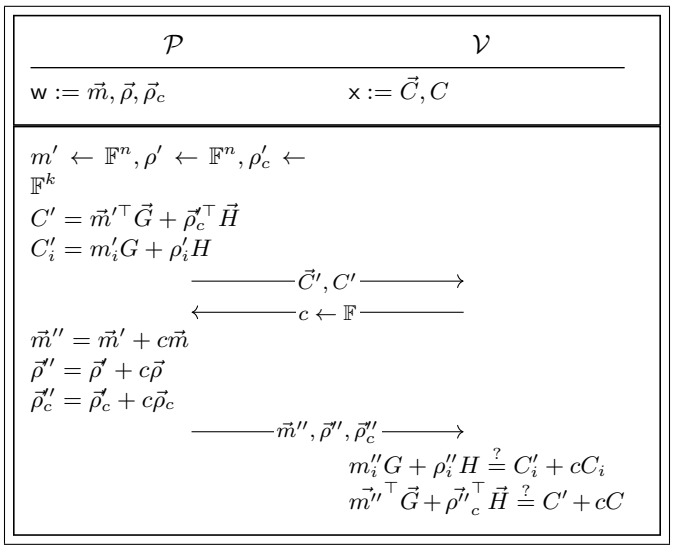


Figure 9: $\Pi_{\mathsf{PD}} : \mathcal{R}_{\mathsf{EQ}} \to \mathcal{R}_{\top}$.

C.2. Equality of Pedersen Commitments Across Groups

We recall the protocol of [19] for proving knowledge of the same opening of two Pedersen commitments across different groups. In particular, let $\mathbb{G}_1, \mathbb{G}_2$ be two groups with corresponding scalar field $\mathbb{F}_1, \mathbb{F}_2$, and let $\mathsf{ck}_1 = (G_1, H_1)$

and $\mathsf{ck}_2 = (G_2, H_2)$ be Pedersen commitment keys for each respectively. The goal is to prove the relation

$$\mathcal{R}_{\mathsf{PD},\mathbb{G}_1,\mathbb{G}_2} = \left\{ \begin{matrix} pp := (\mathsf{ck}_1, \mathsf{ck}_2) \\ \mathsf{x} := C_1, C_2, \\ \mathsf{w} := \big(m, \rho_1, \rho_2\big) \end{matrix} \;\middle|\; \begin{matrix} C_1 = mG_1 + \rho_1 H_1 \wedge \\ C_2 = mG_2 + \rho_2 H_2 \end{matrix} \right\}$$

where m is considered an integer that is canonically embedded in $\mathbb{F}_1, \mathbb{F}_2$. The proposed construction requires m to be bounded, namely $0 \leq m < 2^{b_x}$ for some b_x which parameterized the system. This guarantee can be achieved either by applying a range proof to one of the commitments or it can be assumed to hold in a larger protocol. In our case, we use this to decompose the device public key to two limbs Q_1, Q_2, whose well formness is guaranted during credential issuance.

Roughly, the protocol follows the standard approach to prove equality of commitments *in the same group*, except that it takes care to bound all relevant values, so that no "wraparounds" happen in either field and the equality holds over the integers. The protocol is parameterized by three parameters that determine the efficiency and the security guarantees:

- b_x which is the bound of the committed value m,
- b_c which determines the challenge space and therefore the soundness error, and
- b_f which determines the efficiency of the prover.

The values must be set to satisfy

$$b_x + b_c + b_f < \lceil \log_2(\min(|\mathbb{F}_1|, |\mathbb{F}_2|)) \rceil$$

We describe the protocol in Figure 10 as a reduction of knowledge

$$\Pi_{\mathsf{PD},\mathbb{G}_1,\mathbb{G}_2} : \mathcal{R}_{\mathsf{PD},\mathbb{G}_1,\mathbb{G}_2} \to \mathcal{R}_{\mathsf{RP}}$$

where

$$\mathcal{R}_{\mathsf{RP}} = \left\{ \begin{matrix} pp := (\mathsf{ck}_1) \\ \mathsf{x} := C_1, \\ \mathsf{w} := \big(m, \rho_1\big) \end{matrix} \;\middle|\; \begin{matrix} C_1 = mG_1 + \rho_1 H_1 \wedge \\ 0 \leq m < 2^{b_x} \end{matrix} \right\}$$

and refer the reader to [19] for the security analysis.

Appendix D. Alternative Constructions and Extensions

We next describe two alternative potential approaches to ECDSA proofs-of-possession with different features. The first one allows *efficient precomputation* of proofs and the second aims at privacy against subverted secure elements as defined in [8].

Blinding the public key and precomputation of PoPs. Desmoulins et al. [49] propose an alternative approach for doing ECDSA based proofs-of-possession in the keyed-verification anonymous credential setting. Their construction relies on getting an ECDSA signature under a *randomized*

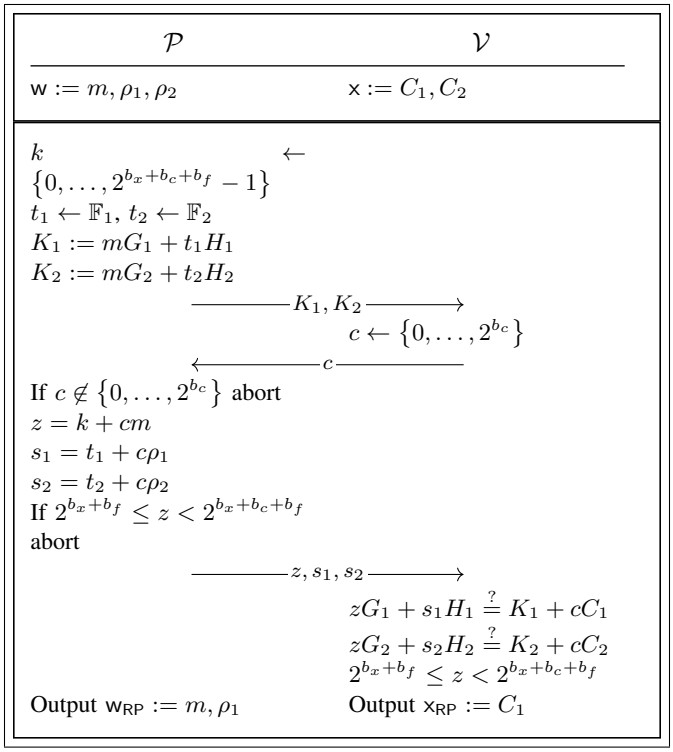


Figure 10: $\Pi_{\mathsf{PD},\mathbb{G}_1,\mathbb{G}_2} : \mathcal{R}_{\mathsf{EQ},\mathbb{G}_1,\mathbb{G}_2} \to \mathcal{R}_{\mathsf{RP}}$.

public key by only accessing the secure element in a black-box way. Intuitively, consider the ECDSA verification equation[15,16]

$$zK = \mathsf{H}(\mathsf{n}, Q)F(K)^{-1}\mathbb{G}_p + Q$$

and multiply both sides by a random $r \leftarrow \mathbb{F}_q$ uniformly sampled from the P-256 scalar field:

$$\begin{aligned} rzK &= r(\mathsf{H}(\mathsf{n}, Q)F(K)^{-1}\mathbb{G}_p + Q) \\ &= r\mathsf{H}(\mathsf{n}, Q)F(K)^{-1}\mathbb{G}_p + rQ \end{aligned}$$

The pair (K, rz) is *almost* a valid ECDSA signature of n under the public key $Q_B = rQ$. The issue is that (1) the digest contains Q instead of Q_B and (2) the scalar of $\mathbb{G}_p$ also contains r. One can address these by non-standard use of the black box: we can query for a (prehashed) signature on the message $\mathsf{h} = r^{-1}\mathsf{H}(\mathsf{n}, Q_B)$. The SE will then reply with two values (K, z) that satisfy $zK = \mathsf{h}F(K, Q_B)^{-1}\mathbb{G}_p + Q$; multiplying both sides by r we get:

$$\begin{aligned} (rz)K &= rr^{-1}\mathsf{H}(\mathsf{n}, Q_B)F(K)^{-1}\mathbb{G}_p + rQ \\ &= \mathsf{H}(\mathsf{n}, Q_B)F(K)^{-1}\mathbb{G}_p + rQ \\ &= \mathsf{H}(\mathsf{n}, Q_B)F(K)^{-1}\mathbb{G}_p + Q_B \end{aligned}$$

15. We present their approach from the lens of the alternative ECDSA verification algorithm presented in Section 2.

16. As noted in [49], to achieve proof-of-possession with this technique, the public key must be included in the digest to prevent related key attacks [53]. Indeed, if it is not included, a malicious user can trivially create many such proofs with only a single query to the SE. We emphasize that this is not an issue in our previous constructions.

Now (z_B, K) is a valid ECDSA signature on n under Q_B, where $z_B = rz$ and $Q_B = rQ$. To turn this into a proof-of-possession, the user can share the randomized signature – which can be verified in the clear – and a proof of knowledge of r s.t. $Q_B = rQ$ where the device public key Q is committed. The latter guarantees that the re-randomized signature is not produced with a fresh key, but rather depends on the attested device public key Q, whose corresponding secret key is only known to the SE. Since the blinded public key Q_B is random and it does not leak information about the device public key Q. Formally, the user and verifier should engage in a reduction-of-knowledge for the relation: $\mathcal{R}_{\mathsf{re\text{-}rand}} =$

$$\left\{ \begin{matrix} \mathsf{x} := \left(C_Q, Q_B\right) \\ \mathsf{w} := \left(\rho_Q, Q, r\right) \end{matrix} \;\middle|\; \begin{matrix} C_Q = \mathsf{GCS.Com}_{\mathsf{ck}}(Q; \rho_Q) \wedge \\ rQ = Q_B \end{matrix} \right\}$$

Noting that the proven predicate is equivalent to $r^{-1}Q_B = Q$, the above relation is a subset of $\mathcal{R}_{\mathsf{C\text{-}DLOG}}$ and therefore the techniques described in the previous sections can be applied to prove it.

This alternative approach does not come with significant efficiency improvements. While it saves a point addition, the dominant cost is the scalar multiplication: the circuit-based implementations essentially rely on variants of the double-and-add algorithm, which involve roughly $\mathcal{O}(\lambda)$ point additions. Similarly, the Schnorr-type proof approach also needs to prove $\mathcal{O}(\lambda)$ point additions as part of the scalar multiplication proof.

That said, this approach has the advantage that it admits *pre-computation*. A user can sample various re-randomized public keys and pre-compute the proofs for $\mathcal{R}_{\mathsf{re\text{-}rand}}$, which do not depend on the nonce or any other value produced during the presentation. Later, when making a presentation, it can query the SE for a signature on the nonce using one of the randomized public keys and present the signature along with the pre-computed proof. While the constructions of the previous sections don't have a prohibitive proving time, it is worthwhile exploring different proving mechanisms that minimize the proof size at the cost of a more costly prover. The more costly prover can be countered via the offline precomputation.

The main drawback of this approach is that it is unclear whether SEs will support "raw" (i.e. without hashing) ECDSA[17]. We also note that the proposed mechanism for proof-of-possession using this technique is more involved since (1) it does not simply produce a proof-of-knowledge of a signature and (2) precomputation might come with additional complications, for example the Fiat-Shamir challenges are not tight to the specific presentation since the proofs are produced in advance. We believe that a more thorough analysis is necessary both to assess the potential efficiency advantages and to get more confidence in the security of the approach, for example by analyzing it in the model of [8]. We defer such analysis to future work.

17. While android supports it, it is marked as deprecated and it will be removed [50].

A PoP secure against subverted secure elements. All previous constructions rely on revealing part of the signature (the value K) as part of the proof-of-possession. While this does not harm privacy when the secure element is honest, it can fully break unlinkability when it is *subverted*. For example, it could sample the k value of the ECDSA signature not uniformly, but rather from a distribution known to an adversary who could exploit this to link "unlinkable" presentations.

A solution to this is to simply not reveal K and increase the complexity of the proven circuit. Concretely, we can define a relation capturing knowledge of an ECDSA signature as $\mathcal{R}_{\mathsf{ECDSA}} =$

$$\left\{ \begin{matrix} \mathsf{x} := \left(C_Q, C_K, \mathsf{n}\right) \\ \mathsf{w} := \left(\rho_Q, Q, \rho_K, K, z\right) \end{matrix} \;\middle|\; \begin{matrix} C_Q = \mathsf{GCS.Com}_{\mathsf{ck}}(Q; \rho_Q) \wedge \\ C_K = \mathsf{GCS.Com}_{\mathsf{ck}}(K; \rho_K) \wedge \\ zK = \mathsf{H}(\mathsf{n})F(K)^{-1}G_p + Q \end{matrix} \right\}$$

Note that the nonce value n is public. Therefore, a circuit for asserting the predicate $zK = \mathsf{H}(\mathsf{n})F(K)^{-1}G_p + Q$ is dominated by two (full) scalar multiplications over P-256. Indeed, the remaining operations are the evaluation of the conversion function F and a point addition. Importantly, no hash computation needs to be proven. It is unclear, however, how to apply the concrete improvements of the CSchnorr protocol in this case, since they inherently rely on the generator K being public.

While one could extend the circuits both in the foreign-field and in the native-field cases to handle this more complex circuit in a straightforward manner, the prover efficiency is expected to increase significantly; first, the optimizations we introduced are no longer applicable and second, two scalar multiplications need to be proven. An interesting open problem is to (1) batch-prove the two scalar multiplications and (2) adapt the committed Schnorr or find new techniques to minimize the circuit complexity in the case where also the base is committed.