Device Binding for Anonymous Credentials on Legacy Phones

Anja Lehmann
anja.lehmann@hpi.de
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany

Alexandros Zacharakis
alexandros.zacharakis@hpi.de
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany

Abstract

Digital identity systems are currently build around the globe, aiming to enable secure, usable, but also privacy-preserving user authentication. Concretely, the EUDI Wallet developed in Europe requires to ensure selective attribute disclosure and unlinkable authentication. This essentially mandates the use of anonymous credentials, that have been developed for this exact purpose over the last 20 years. However, they are not integrated in the current solutions as they lack an essential feature: *device binding*. That is, binding credentials stored on the users' phones to a secure hardware element therein, in order to prevent credential cloning or sharing. Device binding is typically done through encoding a device public key into the user's credential and requiring a fresh signature under the corresponding and hardware-protected secret key – the proof-of-possession (PoP) — when presenting the credential. While academic solutions exist that realize efficient device binding for anonymous credentials, they are not compatible with the secure hardware currently available in consumer phones. The main challenge lies in the underlying curves: all efficient anonymous credentials, (and their native device binding protocols) require the use of pairing-friendly curves, whereas existing phones are essentially restricted to ECDSA signatures and classic P256 curves.

In this work, we show how to bridge these two systems, enabling device-binding for pairing-based credentials on legacy phones, i.e., relying solely on standard ECDSA signatures for the PoP. We present three different constructions with different trade-offs in efficiency and in protocol complexity. Our most efficient solution generates unlinkable bridging proofs of size ~1.5KB in less than ~500ms by relying on a (very simple) arithmetic circuit, whereas the most conservative approach (without circuits) takes as well ~500ms and comes with proof size of ~175KB. All our solutions share a common blueprint, and we express them in the *reductions of knowledge* framework (Crypto 2023) to reflect this is in our protocols' design. This framework allows to modularly construct complex zero-knowledge proofs in an elegant and intuitive manner, greatly facilitating the security analysis and the 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 complex real-world systems.

1 Introduction

Anonymous credentials enable privacy-preserving user authentication. Therein, users hold digital credentials, which are attested attributes from a trusted issuer, and can present them in a decentralized manner towards Relying Parties (RP). 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 [40], ISO [2] and ETSI [20]. They are also being considered for the EUDI, Europe's upcoming digital identity system [22].

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 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 [22]. At the same time, the eIDAS regulation mandates that the EUDI Wallet be privacy-preserving, requiring selective attribute disclosure and unlinkable authentication [1], capabilities that essentially call for anonymous credentials.

While device-binding mechanisms for anonymous credentials exist [13, 14, 26] and have been integrated into secure hardware as part of the TPM 2.0 standard [48], the Secure Elements currently 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. This incompatibility between anonymous credentials and the non-transferability requirement on legacy hardware was a key 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 [44].

Anonymous Credentials for Legacy Systems. To address the challenge of compatibility with legacy hardware, circuit-based solutions have gained significant attention in the digital identity community. Most prominently, zkCred [45], Longfellow [27], Vega [36] or OpenAC [19] provide constructions where an anonymous credential scheme can be build entirely on top of legacy ECDSA credentials. The main advantage of that approach is to be fully legacy compliant with already existing digital identity systems. However, it also comes with inherent disadvantages in terms of complexity and efficiency: the circuits and proof systems need to be highly optimized such that they can perform in (somewhat) practical parameters: e.g. Longfellow takes 500ms to run on a user phone, and

takes 300 KB proof size, While this may suffice for occasional, online credential presentations, it falls short for high-frequency authentication or proximity-based use cases – such as authentication over BLE or NFC, where bandwidth is strictly limited and interaction times must remain below perceptible latency thresholds.

Limits of Monolithic Circuits. Beyond performance, these solutions pose major challenges regarding standardisation and certification. In the context of the European Digital Identity (EUDI) framework, interoperability across member states and wallet providers is a hard requirement. This means that every component of the cryptographic stack – from the signature scheme and the proof system to the proven circuit – must be specified in sufficient detail to ensure that independently developed wallets and verifiers can interact seamlessly. Yet standardisation efforts for circuit-based proof systems within the zkProofs community have been ongoing for over eight years without convergence – suggesting that a deployment path through this route remains distant and uncertain [52].

A further structural limitation of monolithic circuit designs is their rigidity. Extending a large, tightly integrated circuit to support additional features – such as pseudonyms, cross-credential proofs, or deniability – requires redesigning and re-optimising the entire construction. Each such extension would itself need to undergo standardisation and certification anew. Moreover, the complexity of these circuits is a challenge for security, and subtle errors in the circuit design have repeatedly led to critical vulnerabilities in deployed systems (see for example the list of reported bugs in [4]).

BBS with ECDSA Device Binding. An alternative approach, explored by Ubique [7] in a EUDI innovation competition, takes a different route. Rather than forcing the entire credential scheme through a circuit over ECDSA, their proposal leverages a natively ZK-friendly signature scheme – specifically BBS [11] — for the attribute credential, while confining ECDSA solely to device binding. This design acknowledges that legacy constraints are rooted in user devices, where secure elements are already deployed and cannot be updated, while issuers – operating server-side infrastructure – can more easily adopt modern signature schemes. By separating these concerns, the construction benefits from the full expressive power and efficiency of BBS for the credential layer, yielding a design that is both simpler and naturally extensible.

Since BBS and ECDSA operate over different and incompatible elliptic curves – BLS12-381 and P-256 – and the PoP involves proving statements about P-256 elements, the central technical challenges lie in bridging between the two efficiently proving P-256 operations while keeping the device public key hidden. Ubique addresses the former via the techniques of [42] and the latter via the techniques zkAttest/CDLS scheme [17, 23], which solely rely on Schnorr-type prooofs. This is considerably simpler and more conservative than full-fledged SNARKs over ECDSA. Their prototype allows to present a BBS credential with ECDSA device binding in 700ms and produces proofs of around 200 KB. Crucially, the BBS component itself contributes only a few milliseconds and negligible proof size; the costs are dominated by the ECDSA Proof of Possession.

Towards More Efficient Proofs of Possession. While the hybrid BBS–ECDSA approach represents a significant architectural improvement over fully monolithic circuit constructions, the performance of the bridging layer remains a bottleneck. The construction is still impractical, e.g., for proximity-based scenarios such as NFC tap-and-go authentication, where message sizes should ideally remain under a few kilobytes. Schnorr-proof based techniques, however, are inherently limited to proof sizes that are linear to the size of the computation proven, offering limited room for further compression. This motivates the following question:

How can we realize the ECDSA device-binding with a more efficient proof system – e.g., a compact circuit-based proof?

To address this question, we build on the *modular credential framework* proposed in [39]. In this framework, 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 in our case – 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 two new constructions that significantly improve upon the state of the art for enabling pairing-based anonymous credentials on legacy devices, constrained to ECDSA signatures. Both employ lightweight arithmetic circuits, but differ in their trade-offs between proof size and protocol complexity. We additionally formalise and analyse the Schnorr-type proof approach of Ubique within our framework (which lacked a formal analysis), yielding a unified treatment 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 [38]. This framework enables modular composition of zero-knowledge proofs in a way that greatly facilitates both security analysis and implementation. While previously employed mainly in theoretical contexts, our work demonstrates its utility for designing and implementing complex real-world protocols.

The core technical challenge is performing in-circuit P-256 operations while commitments live in BLS12-381. We address this in two ways: either via efficient *foreign-field arithmetic* using the techniques of [5], or by relying on a curve that forms a chain with P-256 [23], nearly eliminating the overhead of non-native computation. In both cases, we further reduce the 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.[1]

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

A family of constructions. Our techniques, combined with the Schnorr-based approach of [7], yield a family of constructions with trade-offs not only in efficiency but also in implementation complexity, standardisation effort, and acceptable cryptographic assumptions – such as whether to rely on the non-standard curve forming a chain with P-256, or on proving systems with more advanced features. 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 [27] 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 [3]. Our most efficient variant produces unlinkable device-binding proofs of approximately 1.5 KB in under 500 ms, while the most conservative – avoiding circuits entirely – achieves comparable proving time at approximately 175 KB proof size. Importantly, our circuits are far simpler than those of fully monolithic approaches: they encode *only the fixed 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 now present the core techniques used in our work, starting with the statement we have to prove for device-binding, to the use of the reductions of knowledge framework and our concrete constructions – including a novel Committed Schnorr Protocol, which we believe to be of independent interest.

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, and the user proves that they can produce a valid ECDSA signature on this nonce under a public key attested in the user's credential. The secret key never leaves the secure element: the user can only issue signature queries and cannot access the key material directly.

In the framework of [39], this corresponds to verifiably disclosing *a commitment to the device public key* and then prove (in ZK) knowledge of a valid signature under this key. Note that the device public key is the only long-lived value and must "live" in the credential and be kept secret. Verifying the ECDSA equation can be translated to proving knowledge of z, K s.t. $zK = \mathsf{H(n)}F(K)G_p + Q$ In the above, Q is the device public key, (z, K) are (deterministically) derived from the ECDSA signature, n is the nonce that must be signed and F the ECDSA conversion function.

Two observations significantly 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.* Together, these reduce the core problem to proving knowledge of a secret scalar z such that

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

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

Minimizing non-native operations. Eq. 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[2] 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:

(1) *Efficient foreign-field arithmetic:* We prove the statement directly over BLS12-381 by emulating the P-256 field operations, using the techniques of [5] over variants of the Plonk proving system [31].
(2) *Native arithmetic via commitment linking:* Instead of emulating foreign-field operations, we "transfer" the BLS12-381 commitments to a commitment scheme whose message space is the base field $\mathbb{F}_p$ of P-256. This transfer is *the only non-native* part of the proof and can be realised at cost comparable to a simple Schnorr-type proof using the technique of [42]. All subsequent operations then work natively over $\mathbb{F}_p$.

In both cases, the vast majority of the cost of proving Eq. 1 reduces to *proving a scalar multiplication* over P-256 – specifically, zK where the scalar z is secret. Proving such a multiplication in a circuit requires encoding the full double-and-add chain with a secret scalar, which dominates the circuit size. This cost is particularly pronounced in the foreign-field case due to the overhead of emulating P-256 arithmetic. We next introduce a new technique that is very simple, yet significantly reduces this cost.

Committed Schnorr Protocol. Our key idea for reducing the aforementioned cost is to extract the secret scalar from the circuit by means of a preliminary Schnorr-like interaction – the *Committed Schnorr Protocol* (CSchnorr).

The idea of CSchnorr is as follows: 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.

The crucial observation is that 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. In summary, the committed Schnorr step trades a

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

single round of interaction for a substantially simpler circuit, and this reduction applies to both of our circuit-based constructions.

Reductions of knowledge (RoK). To construct and analyse the full proof of possession we rely on the reductions of knowledge framework [37, 38]. 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 $\mathcal{R}_2$. As an example, 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.

Proof of possession constructions. We propose three approaches to prove ECDSA device binding as captured by Eq. 1, all starting from a Pedersen commitment over BLS12-381 to the device public key Q. We provide benchmarks for all in Sec. 5.

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 relation to one involving only public scalars; (2) a commitment transfer re-expresses the commitments under a scheme $\widetilde{\mathsf{GCS}}$ compatible with the proving system; and (3) a circuit-based proof proves the resulting relation over P-256:

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

The construction is generic in the choice of commitment scheme and proving system, and we propose two instantiations:

(1) *Foreign-field instantiation (*PoP-PLONK*):* The commitment transfer stays within BLS12-381 – with GCS as Pedersen and $\widetilde{\mathsf{GCS}}$ as KZG – 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 [31], employing the foreign-field techniques of [5]. We instantiate $\Pi_{\mathcal{C},\mathsf{BLS12\text{-}381}}$ with a variant of the Plonk proving system [31], employing the foreign-field techniques of [5] to emulate P-256 arithmetic efficiently. When the committed Schnorr reduction is instantiated with $\lambda = 128$, the corresponding circuit implementation produces proofs of size 3.2KB with proving time on a moderate laptop of around 2.6s and negligible verifaction time[3].

(2) *Native instantiation (*PoP-BP*):* The commitment transfer moves the statement to the Tom curve T-256 [23], 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 [16] and rely on [42] for the cross-group commitment transfer. This yields our most efficient construction: when the committed Schnorr protocol is instantiated with $\lambda = 128$, the proof size is 1.46KB with average proving and verifying time 344ms and 54ms respectively.

Schnorr-based (PoP-Σ): Our third construction formalises the Schnorr-based approach of [7, 34, 39] and avoids circuits entirely. It decomposes the ECDSA relation into a scalar multiplication and a point addition, each proven with a dedicated Schnorr-type protocol. As in the native instantiation, commitments are first transferred from BLS12-381 to T-256, and the two proofs then run in parallel:

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

This is the most conservative of our three approaches – requiring no circuits and relying only on standard Schnorr-type techniques – at the cost of significantly larger proof sizes. When the security parameter for Π_{SM} is set to $\lambda = 128$, the construction's efficiency measures are 172KB proof size, 356ms prover time and 680ms verification time.

On the choice of the proving system. Our generic construction can be instantiated with various proving systems. We focus on proving systems that *minimize proof size*. Small proofs are essential for tap-and-go presentations and – to the best of our knowledge – no such solution has been proposed so far. Since there is inherent "tension" between proof size and prover time, our constructions have similar or in some case worse proving time compared to "monolithic" approaches[5]. We expect that our techniques combined with prover-efficient proof systems such as the ones used in the former will yield significant improvements in the proof size and the prover size, since the circuit used in our work are far simpler and smaller. We leave exploring such approaches as future work.

1.3 Related Work

We survey the most closely related works and discuss how our approach differs. The key distinction is that we do not attempt to optimise the monolithic approach of proving an entire ECDSA credential in a single circuit, but instead exploit the structural simplifications available when ECDSA is confined to device binding.

Most proposed anonymous credential solutions for legacy devices prove possession of an ECDSA-based credential entirely within a single zero-knowledge circuit. Notable examples include Longfellow [27], OpenAC [19], Vega [36], Crescent [43], and Woo et al. [50]. With the exception of Crescent, all produce proofs exceeding 100 KB with prover times in the hundreds of milliseconds. Crescent achieves approximately 13 KB by combining Groth16 [32] for the credential presentation with Spartan [47] for device binding. While Groth16 yields compact proofs, it requires a circuit-specific

[3] In this case we don't hardcode the challenge to the circuit since the efficiency gains are countered by the relatively expensive circuit setup.

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

[5] We emphasize, however, that we intentionally performed our benchmarks on less capable hardware – a commodity laptop with an Intel i5-1345U CPU and 16GB RAM vs 16 vCPUs and 64GB RAM used in [36], from which we borrow the relevant numbers – since this is more in-line with the targeted use-cases.

trusted setup – a ceremony that is difficult to organise in practice and particularly problematic in regulated settings where cryptographic components must be standardised and approved. Moreover, Groth16's prover is computationally expensive, making it less suited for resource-constrained devices. Thus, despite aiming for compact proof sizes, we did not choose Groth16 as a base, but used Plonk, Bulletproofs or standard Schnorr protocols instead.

Further, we stress that our work is not a sub-problem of the monolithic setting. The constructions above must arithmetise hash functions, data parsing, and the complete ECDSA verification equation – including over hidden messages – within a single circuit. By contrast, confining ECDSA to device binding yields a fundamentally simpler statement: the signed nonce is public and the signature is single-use. This admits dedicated optimisations – such as the Committed Schnorr protocol – that are inapplicable in the general case. The result is not merely a smaller circuit but a structurally different construction whose proofs compose in a black-box manner with the credential layer and can be designed, standardised, and replaced independently.

Finally, the Sigmabus protocol of [42] is structurally similar to our Committed Schnorr protocol, in the sense that they also use commitments for the statement and the first message of the Schnorr protocol, but serves a different goal. In their work, the protocol allows to efficiently "witness" in-circuit discrete logarithms of public group elements, while the committed Schnorr protocol *transforms* the initial statement to a simpler one – which can be proven with or without circuits.

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. 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. Throughout this work we consider the P-256 curve [25]. We denote the corresponding group with $\mathbb{G}_p$, its generator with G_p and its base field with $\mathbb{F}_p$.

2.1 ECDSA Signature Scheme

ECDSA is a widely used digital signature scheme defined over elliptic curves. Let $\mathbb{G}$ be a group of order q with generator $G \in \mathbb{G}$ where $\mathbb{G}$ is defined over some elliptic curve 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

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

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 [23], 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 $zK \stackrel{?}{=} \mathsf{H}(m)r^{-1}G + Q$.

2.2 Commitment Schemes

Definition 2.1 (Commitment Scheme). A perfectly hiding commitment scheme with canonical opening $\mathsf{CS} = (\mathsf{Setup}, \mathsf{Com})$ consists of two PPT algorithms:

- $\mathsf{ck} \leftarrow \mathsf{CS.Setup}(pp)$: on input some parameters $pp \leftarrow \mathsf{Gen}(1^\lambda)$[6] it outputs a commitment key ck.
- $C := \mathsf{CS.Com}(\mathsf{ck}, m; \rho)$: on input the commitment scheme and a message m belonging to some messagespace $\mathcal{M}$ and some randomness ρ outputs a commitment C.

Verification is canonical, i.e. to verify a commitment C to a message m given the randomness ρ, the verifier asserts $C \stackrel{?}{=} \mathsf{CS.Com}(\mathsf{ck}, m; \rho)$.

The security guarantees are (1) *binding*, i.e. no PPT adversary $\mathcal{A}(\mathsf{ck})$ can compute two valid openings $(m_1, \rho_1), (m_2, \rho_2)$ such that $m_1 \neq m_2$ and $\mathsf{CS.Com}(\mathsf{ck}, m_1; \rho_1) = \mathsf{CS.Com}(\mathsf{ck}, m_2; \rho_2)$ except with negligible probability, and (2) *perfectly hiding*, i.e. for all $\mathsf{ck} \leftarrow \mathsf{KeyGen}(pp)$ and $m_1, m_2 \in \mathcal{M}$ the distributions

$$\{\ C_1 \mid C_1 \leftarrow \mathsf{CS.Com}(pp, m_1)\ \}, \{\ C_2 \mid C_2 \leftarrow \mathsf{CS.Com}(pp, m_2)\ \}$$

are perfectly indistinguishable

Committing to elliptic curve points. Our aim is to prove statements about ECDSA over the P-256 curve. Due to privacy, we need to commit to device public keys which is information that identifies a user. Such keys are elliptic curve points $Q \in \mathbb{G}_p \subseteq \mathbb{F}_p^2$. We will refer to (generic) commitment schemes with message space $\mathbb{G}_p$ throughout 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 the T-256 curve [23].

2.3 Reductions of Knowledge

We analyze and implement our constructions using the *reductions of knowledge* [37, 38] framework. We describe them next informally and defer the formal definitions and relevant theorems in App.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 $(\mathsf{x}, \mathsf{w}) \in \mathcal{R}$, after an honest interaction between $\mathcal{P}$ and $\mathcal{V}$ the output $(\mathsf{x}', \mathsf{w}')$ belongs in $\mathcal{R}'$.
- *Knowledge Soundness*: if a malicious prover $\mathcal{P}^*$ knows a valid witness w' for the reduced statement x', it must also know a

[6] We assume all parameters such as groups, generators etc are sampled by a common public parameter generation algorithm Gen, allowing us to use the same parameters across different primitives.

valid witness w for x. This is formalized by a knowledge extractor algorithm that *also takes as input the witness for the resulting statement* and outputs the witness w for the initial statement.
- *Public Reducibility*: the new statement output at the end of the protocol can be efficiently computed from the transcript of the interaction between $\mathcal{P}$ and $\mathcal{V}$.
- *Honest Verifier Zero Knowledge*: the transcript of the interaction between $\mathcal{P}$ and (an honest) $\mathcal{V}$ is perfectly simulatable.

If a reduction is also public-coin –meaning that 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 [24].

Reductions of knowledge are more expressive than proofs of knowledge in the sense that the former can capture the latter. Indeed, we can view a proof of knowledge as a reduction of knowledge $\Pi : \mathcal{R} \to \mathcal{R}_\top$ where $\mathcal{R}_\top$ is the trivial relation that only accepts $\mathsf{x} = \mathsf{true}$ (capturing that the proof of knowledge verifier accepts) and reject all other values (capturing that the proof of knowledge verifier rejects). We use the notation $\Pi : \Pi_{\mathcal{R}} \to \Pi_\top$ to express proofs of knowledge.

As shown in [37, 38], reductions of knowledge:

(1) compose sequentially, in the sense that 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 reduction of knowledge from $\mathcal{R}_1$ to $\mathcal{R}_3$,
(2) compose in parallel, in the sense that 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 reduction of knowledge from $\mathcal{R}_1 \times \mathcal{R}_3 \to \mathcal{R}_2 \times \mathcal{R}_4$, where

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

As demonstrated in [38], many existing constructions can be expressed in this framework with a significantly simplified analysis.

Remark 1. *In this work 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.*

2.4 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 the relation:

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

Note that the above relation "splits" the witness to two parts: some elements are hidden and some are committed. All relations in this work can be expressed in this way. When needed we explicitly express relations as committed relations.

2.5 Equality of Committed Values $\mathcal{R}_{\mathsf{EQ}}$

We next show how to construct a reduction of knowledge 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. Looking ahead, in our proof-of-possession construction, we start by proving a computation about a P-256 equation with points committed using Pedersen commitment over a pairing curve.

Given a predicate P and two commitment schemes 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) proof of knowledge for the relation

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

Let $\Pi_{\mathsf{EQ}} : \mathcal{R}_{\mathsf{EQ}} \to \mathcal{R}_\top$ be a proof of knowledge. The RoK $\Pi_{\mathsf{CS}\to\widetilde{\mathsf{CS}}}$ simply consists of $\mathcal{P}$ sending fresh commitments under $\widetilde{\mathsf{CS}}$ for all the committed values and the prover and verifier engaging in parallel executions of Π_{EQ}. We present the construction in App. B.

2.6 PoKs for Curve Operations $\mathcal{R}_{\mathsf{PA}}, \mathcal{R}_{\mathsf{SM}}$

We rely on proofs of knowledge that capture the validity of Elliptic Curve 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 elliptic curve and let GCS be a commitment scheme with message space $\mathbb{G}$. Define the following committed relations:

$$\begin{aligned} \mathcal{R}_{\mathsf{PA}} &= \left\{ \begin{matrix} pp = \mathsf{ck} \\ \mathsf{x} = (C_i)_{i=1}^3 \\ \mathsf{w} = (G_i, \rho_i)_{i=1}^3 \end{matrix} \;\middle|\; \begin{matrix} C_i = \mathsf{GCS.Com}_{\mathsf{ck}}(G_i; \rho_i) \wedge \\ G_1 + G_2 = G_3 \end{matrix} \right\} \\ \mathcal{R}_{\mathsf{SM}} &= \left\{ \begin{matrix} pp = \mathsf{ck} \\ \mathsf{x} = (C, G) \\ \mathsf{w} = (H, \rho, x) \end{matrix} \;\middle|\; \begin{matrix} C = \mathsf{GCS.Com}_{\mathsf{ck}}(G; \rho) \wedge \\ H = xG \end{matrix} \right\} \end{aligned} \tag{2}$$

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)

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 the prover 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[7].

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

- $\mathcal{P}$ first samples $r \leftarrow \mathbb{F}_q$ and sends to the verifier $R \leftarrow rK$,
- $\mathcal{V}$ sends a random challenge $c \leftarrow \mathbb{F}_q$,
- $\mathcal{P}$ responds with $s := r + c \cdot z$.

The verifier accepts if $sK = R + cQ$.

Hiding the value Q. Now, consider the case where one would like to keep Q secret. Looking ahead, this value can be an *attested group element* derived from some credential presentation. The credential holder might need to prove knowledge of the corresponding discrete log value, but not reveal the element. Indeed, 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:

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

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

The Committed Schnorr 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. The prover $\mathcal{P}$ has input $\mathsf{w} := (\rho_Q, Q, z)$ and interacts with a verifier that has input $\mathsf{x} := C_q, H, K$ as follows:

- $\mathcal{P}$ first samples $r \leftarrow \mathbb{F}_q$, $R \leftarrow rK$ and sends to the verifier $C_R := \text{GCS}_{\mathsf{ck}}.\text{Com}(R; \rho_R)$,
- $\mathcal{V}$ sends a random challenge $c \leftarrow \mathbb{F}_q$,
- $\mathcal{P}$ responds with $s := r + c \cdot z$.

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 $T = sK - cH$.

The verifier, 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

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

then the protocol defined above is a reduction of knowledge $\Pi_{\text{CSchnorr}} : \mathcal{R}_{\text{C-DLOG}} \rightarrow \mathcal{R}_{\text{CSchnorr}}$.

There are two important differences between the two relations $\mathcal{R}_{\text{C-DLOG}}$ and $\mathcal{R}_{\text{CSchnorr}}$ 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 $1/\left|\mathbb{F}_q\right|$ in the general case. However, when $\left|\mathbb{F}_q\right|$ 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 above reduction of knowledge in Fig. 1 and give a theorem statement and proof showing that the above protocol (when instantiated with a challenge space $\{0,1\}^\lambda$) is a reduction of knowledge.

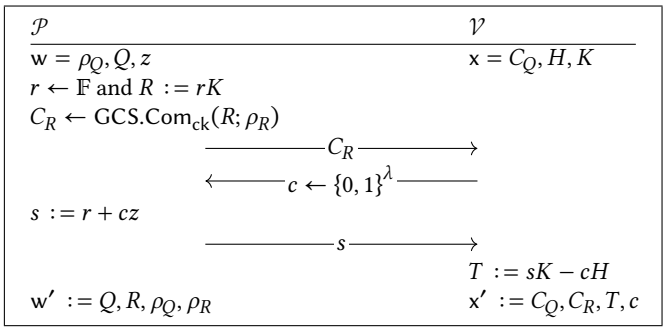

Figure 1: $\Pi_{\text{CSchnorr}} : \mathcal{R}_{\text{C-DLOG}} \rightarrow \mathcal{R}_{\text{CSchnorr}}$.

Theorem 3.1. *Let* GCS *be a perfectly hiding commitment scheme with message space* $\mathbb{G}$. *Then, the protocol* Π_{CSchnorr} *described in Fig. 1 is an honest-verifier public-coin reduction of knowledge* $\Pi_{\text{CSchnorr}} : \mathcal{R}_{\text{C-DLOG}} \rightarrow \mathcal{R}_{\text{CSchnorr}}$. *Furthermore, when the verifier's message is sampled from* $\{0,1\}^\lambda$, *it has knowledge soundness error* $2^{-\lambda} + 2\epsilon_{\text{GCS}}$ *where* ϵ_{GCS} *is the advantage of an adversary against the binding property of* GCS.

Proof.

Completeness and public reducibility. Assume $\mathsf{x}, \mathsf{w} \in \mathcal{R}_{\text{C-DLOG}}$ 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

[7] Note that this means we produce a new circuit for each proven statement. In our case this is acceptable since the verifier 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.

values it also holds that $T = R + cQ$, we conclude that the reduced statement/witness pair indeed satisfies $\mathsf{x}', \mathsf{w}' \in \mathcal{R}_{\text{CSchnorr}}$. 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 A.3. 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 $(\mathsf{x}, \mathsf{w}) \in \mathcal{R}_{\text{C-DLOG}}$.

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

We now present our two generic constructions for ECDSA device binding. Both take as input a Pedersen commitment over BLS12-381 to the device public key and prove knowledge of a valid ECDSA signature under the committed key. The first construction, Π_{Circ}, is circuit-based, whereas the second construction, Π_Σ, formalises the Schnorr-based approach of [7] and requires no circuits.

We express both constructions in the reductions of knowledge framework, which simplifies the security analysis and allows the individual components to be instantiated independently. We leverage this modularity in Sec. 5, where we give two instantiations of Π_{Circ}: one requires optimised foreign-field arithmetic (PoP-PLONK), and another that links commitments to the additional curve T-256 – whose scalar field coincides with the base field $\mathbb{F}_p$ of P-256 – making all circuit operations native (PoP-BP).

ECDSA PoP. Our starting point is a relation $\mathcal{R}_{\text{pECDSA}}$ 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:

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

where $\alpha = \mathsf{H}(\mathsf{n}) F(K)$, and our goal is to prove (in ZK) statements for this relation. Note that part of the signature (the value K) is part of the statement. Assuming an honest Secure Element [26], this is enough to guarantee proof-of-possession of the device [39, 43].

Recall also the committed discrete logarithm relation:

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

which is the relation proven by the Committed Schnorr protocol of 3. Note that $\mathcal{R}_{\text{pECDSA}}$ is trivially reduced to it by simply setting $H := \alpha G_p$. Our goal is essentially to construct efficient protocols for proving this statement. In all our constructions the commitment scheme used in $\mathcal{R}_{\text{pECDSA}}$ is the Pedersen commitment scheme over BLS12-381. To commit to points we commit to the coordinates of the points decomposed in two 128-bits limb.

4.1 Π_{Circ}: Circuit-based PoP

Our first construction is based on (simple) arithmetic circuits to prove the P-256 operations. The construction proceeds in three steps. First, we apply the Committed Schnorr 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. Third, we prove the resulting statement using a generic proof of knowledge. We call this composition Π_{Circ} and discuss two the instantiation of each component below.

In more detail, after expressing $\mathcal{R}_{\text{ECDSA}}$ as an instance of the relation $\mathcal{R}_{\text{C-DLOG}}$, capturing validity of the equation $zK = H + Q$ (over P-256), the prover and verifier engage in three reductions of knowledge to prove the latter. First the prover $\mathcal{P}$ and verifier $\mathcal{V}$ engage in an execution of Π_{CSchnorr} to reduce the $\mathcal{R}_{\text{C-DLOG}}$ statement to a $\mathcal{R}_{\text{CSchnorr}}$ statement. Recall that the latter corresponds to

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

Recall also from Sec. 2.5 that we denote with $\mathcal{R}_{\text{GCS,P}}$ 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 $\mathcal{R}_{\text{CSchnorr}}$ as the committed relation $\mathcal{R}_{\text{GCS,P}}$ over GCS. Next, let $\Pi_{\mathcal{C},\text{P}}$ be a public coin, HVZK proof of knowledge for the same relation under a different commitment scheme $\widetilde{\text{GCS}}$, namely, $\Pi_{\mathcal{C},\text{P}} : \mathcal{R}_{\widetilde{\text{GCS}},\text{P}} \to \mathcal{R}_\top$ and recall from Sec. 2.5 that $\Pi_{\text{GCS} \to \widetilde{\text{GCS}}}$ reduces validity of a committed relation under GCS to validity of the same relation under a different commitment scheme $\widetilde{\text{GCS}}$. In the second step of the protocols $\mathcal{P}$ and $\mathcal{V}$ "transfer" the commitment from the original commitment scheme GCS to $\widetilde{\text{GCS}}$ which is the one that is supported by the circuit-based proof of knowledge. Finally, they engage in an execution of $\Pi_{\mathcal{C},\text{P}}$ to prove the reduced statement. Therefore, our first construction, presented in Fig 2, is essentially a composed reduction of knowledge

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

where the $\mathcal{R}_\top$ is the trivial relation as explained in Sec. 2.3, capturing that the latter protocol is a proof of knowledge.

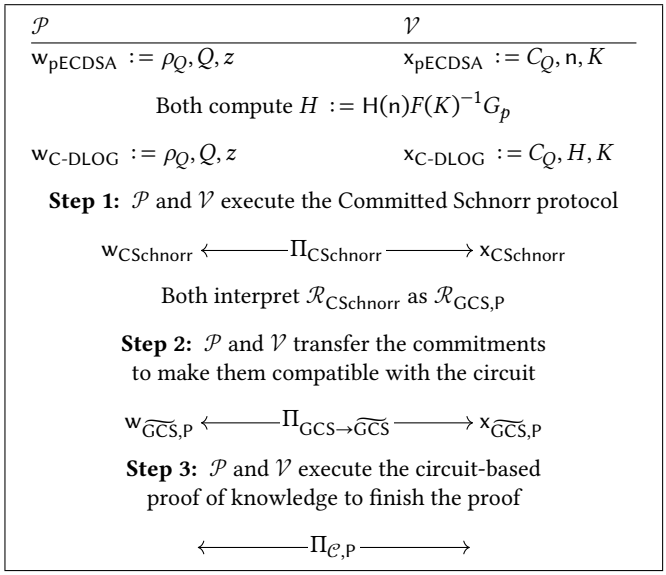

Figure 2: $\Pi_{\mathsf{Circ}} : \mathcal{R}_{\mathsf{pECDSA}} \to \mathcal{R}_\top$. The arrows indicate interactions between $\mathcal{P}$ and $\mathcal{V}$. The protocol can be turned to non-interactive via the Fiat-Shamir transform.

Theorem 4.1. *Let* 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 proof of knowledge, 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 proof of knowledge 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

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

Therefore by Thm. A.6, Π_{Circ} is a public coin, HVZK proof of knowledge for the relation $\mathcal{R}_{\mathsf{pECDSA}}$. □

Instantiation trade-offs: foreign-field vs. native. The construction is generic in the choice of $\widetilde{\mathsf{GCS}}$ and the circuit-based proving system $\Pi_{\mathcal{C},\mathsf{P}}$, and different instantiations lead to different trade-offs. If both GCS and $\widetilde{\mathsf{GCS}}$ operate over groups where standard assumptions already hold – such as BLS12-381 or P-256 – then the construction is assumption-optimal in the sense that it introduces no new cryptographic assumptions beyond those already required by the credential scheme and the proving system. The drawback is that $\Pi_{\mathcal{C},\mathsf{P}}$ must then emulate P-256 arithmetic over a foreign field, incurring the associated overhead. We call this the foreign-field arithmetic (FFA) approach and instantiate it concretely in Sec. 5 as PoP-PLONK, using a variant of the Plonk proving system [31] with the foreign-field techniques of [5].

Alternatively, one can choose $\widetilde{\mathsf{GCS}}$ such that its message space is the base field $\mathbb{F}_p$ of P-256, making the P-256 operations native to the circuit. This eliminates the foreign-field overhead entirely, yielding a more efficient circuit. The trade-off is that this may introduce a new assumption: in our instantiation, we use the Tom curve T-256 [23] – whose scalar field equals $\mathbb{F}_p$ – for the commitment scheme $\widetilde{\mathsf{GCS}}$, which requires the discrete logarithm assumption to hold over T-256 in addition to the standard assumptions. We instantiate this approach in Sec. 5 as PoP-BP, using a Bulletproofs as proving system [16] over T-256.

4.2 Π_Σ: PoP via Schnorr-Protocols

We next describe our second construction, which also relies on commitment linking to reduce the statement to a native one. Unlike the previous two, however, it requires no circuit-based proofs and is built exclusively from Schnorr-type protocols – specifically, proofs of knowledge for the relations $\mathcal{R}_{\mathsf{PA}}$ (point addition) and $\mathcal{R}_{\mathsf{SM}}$ (scalar multiplication) (Eq. 2). This construction was first implemented in [7, 34] and formalised in [39], but only with an informal security analysis. We provide an analysis in the reductions of knowledge framework. We also simplify the construction and improve its concrete efficiency by omitting the commitment to the signature component z and all related commitments communicated during the proof, which in the original construction amount to $\mathcal{O}(\lambda)$ additional commitments.

The starting point is a commitment transfer RoK (Sec. 2.5): given a statement/witness pair $(\mathsf{x}_{\mathsf{pECDSA}}, \mathsf{w}_{\mathsf{pECDSA}})$ under an arbitrary commitment scheme GCS, we transfer the commitments to a native scheme $\widetilde{\mathsf{GCS}}$ with commitment space $\mathbb{F}_p^\ell$ where $\mathbb{F}_p$ is the base field of P-256 and ℓ the number of field elements to represent a P-256 group element. We then derive the corresponding $\mathcal{R}_{\mathsf{C\text{-}DLOG}}$ statement/witness pair, reducing the problem to proving that $zK = H + Q$, where z is a witness and Q is a committed input. The statement can 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 the verifier,
(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 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 and then present our final protocol Π_Σ in Fig. 4 that uses $\Pi_{\mathbb{G}}$.

Theorem 4.2. *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 $Z = zK$,

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

Figure 3: $\Pi_{\mathbb{G}} : \mathcal{R}_{\text{C-DLOG}} \to \mathcal{R}_{\mathsf{SM}} \times \mathcal{R}_{\mathsf{PA}}$.

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 $\mathsf{x}_{\text{C-DLOG}}$.

Knowledge Soundness. Let $\mathcal{A}$ and $\mathcal{P}^*$ be PPT adversaries as described in Def. A.1. We construct a PPT extractor $\mathcal{E}$ that works as follows: on input $\mathsf{x}_{\text{C-DLOG}}$ and st, $\mathcal{E}$ runs $\mathsf{w}_{\mathsf{SM}}, \mathsf{w}_{\mathsf{PA}} \leftarrow \mathcal{P}^*(\mathsf{x}_{\text{C-DLOG}}, \mathsf{st})$[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 $\mathsf{w}_{\text{C-DLOG}} := \rho_{Q_2}, Q_2, z_1$. We claim that the probability that $(\mathsf{x}_{\text{C-DLOG}}, \mathsf{w}_{\text{C-DLOG}}) \in \mathcal{R}_{\text{C-DLOG}}$ 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 $(\mathsf{x}_{\text{C-DLOG}}, \mathsf{w}_{\text{C-DLOG}}) \in \mathcal{R}_{\text{C-DLOG}}$.

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. □

Theorem 4.3. *Let* GCS, $\widetilde{\text{GCS}}$ *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 proof of knowledge for* $\mathcal{R}_{\mathsf{pECDSA}}$.

Proof. We argue similarly to the proof of Thm. 4.1. Let $\Pi : \mathcal{R}_{\mathsf{ECDSA}} \to \mathcal{R}_{\text{C-DLOG}}$ be the trivial reduction as in the proof of Thm. 4.1. Then, Π_Σ is by construction the composition of reductions of knowledge $(\Pi_{\mathsf{SM}} \times \Pi_{\mathsf{PA}}) \circ \Pi_{\mathbb{G}} \circ \Pi_{\text{GCS}\to\widetilde{\text{GCS}}} : \mathcal{R}_{\mathsf{pECDSA}} \to \mathcal{R}_\top$ and the theorem follows by Thm. A.6. □

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

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

Figure 4: $\Pi_\Sigma : \mathcal{R}_{\mathsf{pECDSA}} \to \mathcal{R}_\top$. **The arrows indicate interactions between $\mathcal{P}$ and $\mathcal{V}$. The protocol can be turned to non-interactive via the Fiat-Shamir transform.**

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

Instantiation. We next describe how to concretely instantiate Π_Σ to derive our first construction PoP-Σ. The initial part of the protocol – transferring commitments from GCS to $\widetilde{\text{GCS}}$ – is identical to the native circuit case and is handled as in that case. For proving $\mathcal{R}_{\mathsf{SM}}$ and $\mathcal{R}_{\mathsf{PA}}$ we rely on Schnorr-type proofs using the techniques of [17, 23]. We present more details in Sec. 5.

5 Implementation and Benchmarks

In this section we describe how to concretely instantiate the instantiations of Sec. 4; in particular we propose two circuit-based instantiations, PoP-PLONK and PoP-BP and a slightly optimized variant of PoP-Σ. For the first construction PoP-PLONK, we provide a Rust implementation of the circuit part, corresponding to $\Pi_{\mathcal{C},\mathsf{p}}$ of Fig. 2, which is the dominant efficiency bottleneck; constructions PoP-BP and PoP-Σ are implemented in Rust in their entirety. We also implement a separate Rust crate that defines types and the composition operations for reductions of knowledge which the former utilize. We provide benchmarks of the constructions in Sec.5.4. The implementation is available as an open source library in [3].

We next describe how to instantiate our generic constructions of Sec 4 to derive our three constructions PoP-PLONK, PoP-BP and PoP-Σ. For all our constructions the starting point is the relation:

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

where GCS uses the Pedersen commitment over BLS12-381 to commit to P-256 points. This is motivated by the framework of [39];

in their framework, the user outputs BLS12-381 Pedersen commitments to attributes and proves that they are consistent with the credential. Then other proofs – as our PoP – are proven of the outputted commitments.

Encoding and Committing to P-256 *points.* We first start by describing the commitment GCS that allows to commit to Q. We use Pedersen commitment over the BLS12-381 curve. In particular, denote the BLS12-381 group with $\mathbb{G}_{\text{BLS12-381}}$ and its scalar with $\mathbb{F}_{\text{BLS12-381}}$. To commit to a P-256 point $Q = (x, y) \in \mathbb{F}_p^2$ we start by decomposing the coordinates to 128-bit limbs, that is, we compute $0 \leq x_l, x_h, y_l, y_h < 2^{128}$ s.t. $x = x_l + 2^{128}x_h$ and $y = y_l + 2^{128}y_h$. Let $G_b, H_b \in \mathbb{G}_{\text{BLS12-381}}$ be a Pedersen commitment key over BLS12-381. The commitment to Q corresponds to the four commitments to x_l, x_h, y_l and y_h.

In the context of AC, the commitment to the ECDSA public key Q is assumed to be verifiably disclosed during a credential presentation as described in [39] and is assumed to be honest: it indeed encodes a P-256 point and each of its comprising elements is a commitment to a 128-bit number. These guarantees are imposed during issuance of the credential, in which the ECDSA public key is revealed in clear to the issuer.

5.1 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. In particular we need to explain how to instantiate (1) the circuit par $\Pi_{\mathcal{C},\text{P}}$ with its corresponding commitment GCS and (2) the reduction $\Pi_{\text{GCS}\to\widetilde{\text{GCS}}}$.

Proving system $\Pi_{\mathcal{C},\text{P}}$ *and* GCS. We use the implementation of [41] which is a variant of Plonk [31] with richer arithmetisation, in particular custom gates [29] and lookup tables [30]. The proving system supports "committed inputs" using the techniques described in [6]; in particular, it supports the KZG polynomial commitment scheme [35]. The commitment key is the same as in standard (compact) Pedersen commitment scheme – i.e. the variant that uses many generators to commit to many elements in a single group element – 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 (depending on how the circuit is defined)[9]. Regarding the circuit implementation, we rely on the techniques of [5] to implement efficient foreign field arithmetic, in particular emulating $\mathbb{F}_p$ arithmetic over the scalar field $\mathbb{F}_{\text{BLS12-381}}$. As an optimization, we only commit to the x-coordinates of these points[10].

Linking Proof $\Pi_{\text{GCS}\to\widetilde{\text{GCS}}}$. Our task is to prove that four (plain) Pedersen commitments, corresponding to the public key Q and the commitment produced during the Committed Schnorr execution, and a KZG commitment have the same openings. Since the latter can be viewed as compact Pedersen, a standard Schnorr-type proof can be used. For completeness, we recall such proof in App. C.

Security and Trust Assumptions. Our task is to prove that two (plain) Pedersen commitments and a KZG commitment have the same openings. 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 [28]. The construction requires a KZG polynomial commitment key which is universal – i.e. requires no trusted setup per circuit. Also it is updateable [33] which means it can be sample 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 (see for example [21]).

5.2 Construction PoP-BP

We next explain how to instantiate the PoP-BP construction. We explain how to instantiate the two primitives of Π_{Circ} as in the previous case.

Proving system $\Pi_{\mathcal{C},\text{P}}$ *and* GCS. The proving system is based on techniques of Bulletproofs [16], in particular the constructions described in [15, 46]. It consists of a simple reduction of R1CS to an inner-product relation and is much simpler than other generic constructions, making it particularly relevant for our use-case. The proving system implementation is a modified version of [51]; in particular, we modify the library to allow parts of the public input to be committed. The corresponding commitment scheme $\widetilde{\text{GCS}}$ is *compact Pedersen commitment scheme over* T-256. As in the previous case, we only commit to the x-coordinates of the points.

Linking Proof $\Pi_{\text{GCS}\to\widetilde{\text{GCS}}}$. Our task is to show equality of openings of (1) four plain Pedersen commitments over BLS12-381 (GCS) with (2) a compact Pedersen commitment over T-256 ($\widetilde{\text{GCS}}$). We do this in two steps. First we use the techniques of [42] to show that four plain Pedersen commitments over BLS12-381 (GCS) have the same openings with four plain Pedersen commitments over P-256 (GCS). Then, we show that the latter have the same openings with a compact Pedersen commitment over T-256 using the same techniques as in the case of PoP-PLONK. The techniques of [42] have similar efficiency to standard Schnorr-type proofs. We recall both constructions in App. C.

Security and Trust Assumptions. 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*; in particular, the parameters consist of the Pedersen commitment keys.

An optimisation for reducing proof size. Note that we do the commitment linking twice: from plain Pedersen over BLS12-381 to plain Pedersen over T-256 and then to compact Pedersen over T-256. One can perform the former *before the execution of Committed Schnorr* to avoid also transferring the commitment produced in its execution. This corresponds to reducing $\mathcal{R}_{\text{pECDSA}}^{\text{BLS12-381}}$ to $\mathcal{R}_{\text{pECDSA}}^{\text{T-256}}$ where the superscript denotes where the commitments of the relation

[9] 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.

[10] This means that we lose roughly two bits of security since each x-coordinate corresponds to two elliptic curve points. Asserting the witnessed points Q, R lie on P-256 is done inside the circuit.

	Prover (ms)	Verifier (ms)	Proof (KB)
PoP-PLONK	2675	4.5	3.2
PoP-BP	344	54	1.47
PoP-Σ	356	680	172

Table 1: The average efficiency measures over 50 executions. For PoP-PLONK we only implement the circuit that realizes the component $\Pi_{\mathcal{C},\mathsf{P}}$ of Sec. 4. In both circuit-based constructions we instantiate the committed schnorr protocol with a security parameter of 128 bits. For PoP-PLONK and PoP-Σ, we instantiate the equality across group with a security parameter of 112 bits.

live. Then, we can use the construction Π_{Circ} to prove the latter unchanged but this time we only need to transfer the plain to the compact Pedersen, both over T-256. 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. A.6).

5.3 Construction PoP-Σ

Our last construction relies on the implementation of [7] formalized in [39]. We slightly improve the construction by omitting some unnecessary commitments to the P-256 scalars that were used. The commitment linking uses again the techniques of [42] as in the previous case. The proofs for Π_{PA}, Π_{SM} are based on [17, 23]. The former essentially proves the $\mathbb{F}_p$ equations used for point addition and proves them with simple Schnorr-type proofs while the latter uses bit-challenge Schnorr-type proofs with parallel repetitions. We recall both in App. D, E; the latter includes also our optimizations. As in the previous case, the constructions rely on security of DLOG in both BLS12-381 and T-256.

5.4 Implementation and Experimental Results

We provide rust implementations of PoP-PLONK, PoP-BP, and PoP-Σ in [3]. For PoP-PLONK we only implement the circuit which is the dominant part; the proof size when considering the rest parts – Committed Schnorr and commitment linking – roughly contributes to 700B with minimal impact on proving time. For the implementation we use [41] which also includes support for highly performant emulation of P-256 over BLS12-381 using the techniques of [5]. We don't consider a circuit depending on a hardcoded Committed Schnorr challenge in this case since the setup is costly and essentially counters all the efficiency prover gains while significantly harming the verifier. Therefore, a single circuit is used for all statements. The resulting circuit complexity is captured by a computational trace of 12457 rows, 9 advice columns and 39 fixed columns.

For the PoP-BP construction, we extend the implementation of the Bulletproof based proving system of [51] to allow support for committed inputs. After our optimisation from the committed Schnorr protocol, the resulting circuit's complexity is captured by 760 R1CS constraints (on average). In this case, the circuit depends on the statement.

The constructions PoP-BP and PoP-Σ are implemented in the reduction of knowledge framework of [38]; we provide a separate, independent rust library that implements the framework. For PoP-Σ we use the implementation of [34] which we slightly optimize by reducing some unnecessary commitments and the corresponding proof elements in the scalar multiplication proof.

We benchmark our constructions on a mid-range commodity laptop with an Intel Core i5-1345U processor, 16GB or RAM and no dedicated graphics card. The benchmarks also include the time to serialize and de-serialize the proof, which in the case of PoP-Σ is significant due to the large proof size and the compression of group elements. We present the benchmark results, which correspond to averages of 50 executions, in Table 1.

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 175 KB with proving times consistently under 500 ms. 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 [39], 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 F. The first, building on Desmoulins et al. [18], 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 sizes. However, the approach requires raw ECDSA signing support from the secure element – a capability currently being deprecated on major platforms [8] – and its security model warrants further analysis.

The second extension addresses the threat of subverted secure elements, recently studied in [26]. 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. This research was partially funded by SPRIND, the Federal Agency for Breakthrough Innovation. The authors are solely responsible for the content of this publication; the positions presented here do not reflect the views of SPRIND.

References

[1] 2014. 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. Accessed: 2026-04-27.

[2] 2025. *ISO/IEC WD 24843: Information security - Attribute-Based Credentials*. Working Draft ISO/IEC WD 24843. International Organization for Standardization. Under development. A working group has prepared a draft..

[3] 2026. ecdsa_pops. https://github.com/hpicrypto/ecdsa_pops.

[4] 0xPARC. 2024. ZK Bug Tracker. https://github.com/0xPARC/zk-bug-tracker. A community-maintained collection of bugs, vulnerabilities, and exploits in applications using zero-knowledge cryptography. Accessed 27 April 2026.

[5] Miguel Ambrona, Denis Firsov, and Iñigo Querejeta-Azurmendi. 2025. Efficient Foreign-Field Arithmetic in PLONK. Cryptology ePrint Archive, Report 2025/695. https://eprint.iacr.org/2025/695

[6] Miguel Ambrona, Anne-Laure Schmitt, Raphael R. Toledo, and Danny Willems. 2022. New optimization techniques for PlonK's arithmetization. Cryptology ePrint Archive, Report 2022/462. https://eprint.iacr.org/2022/462

[7] Patrick Amrein. 2025. https://github.com/UbiqueInnovation/zkattest-rs.

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

[9] Thomas Attema, Serge Fehr, and Michael Klooß. 2022. Fiat-Shamir Transformation of Multi-round Interactive Proofs. In *TCC 2022, Part I (LNCS, Vol. 13747)*, Eike Kiltz and Vinod Vaikuntanathan (Eds.). Springer, Cham, 113–142. doi:10.1007/978-3-031-22318-1_5

[10] Mihir Bellare and Tadayoshi Kohno. 2003. A Theoretical Treatment of Related-Key Attacks: RKA-PRPs, RKA-PRFs, and Applications. In *EUROCRYPT 2003 (LNCS, Vol. 2656)*, Eli Biham (Ed.). Springer, Berlin, Heidelberg, 491–506. doi:10.1007/3-540-39200-9_31

[11] Dan Boneh, Xavier Boyen, and Hovav Shacham. 2004. Short Group Signatures. In *CRYPTO 2004 (LNCS, Vol. 3152)*, Matthew Franklin (Ed.). Springer, Berlin, Heidelberg, 41–55. doi:10.1007/978-3-540-28628-8_3

[12] Jonathan Bootle, Alessandro Chiesa, and Katerina Sotiraki. 2021. Sumcheck Arguments and Their Applications. In *CRYPTO 2021, Part I (LNCS, Vol. 12825)*, Tal Malkin and Chris Peikert (Eds.). Springer, Cham, Virtual Event, 742–773. doi:10.1007/978-3-030-84242-0_26

[13] Ernie Brickell, Liqun Chen, and Jiangtao Li. 2008. A New Direct Anonymous Attestation Scheme from Bilinear Maps. In *Trusted Computing - Challenges and Applications (TRUST 2008) (Lecture Notes in Computer Science)*. Springer.

[14] Ernest F. Brickell, Jan Camenisch, and Liqun Chen. 2004. Direct Anonymous Attestation. In *ACM CCS 2004*, Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel (Eds.). ACM Press, 132–145. doi:10.1145/1030083.1030103

[15] Benedikt Bünz. 2023. *Improving the Privacy, Scalability, and Ecological Impact of Blockchains*. Doctor of Philosophy dissertation. Stanford University. https://cs.nyu.edu/~bb/papers/thesis.pdf

[16] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. 2018. Bulletproofs: Short Proofs for Confidential Transactions and More. In *2018 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 315–334. doi:10.1109/SP.2018.00020

[17] Sofía Celi, Shai Levin, and Joe Rowell. 2024. CDLS: Proving Knowledge of Committed Discrete Logarithms with Soundness. In *AFRICACRYPT 24 (LNCS, Vol. 14861)*, Serge Vaudenay and Christophe Petit (Eds.). Springer, Cham, 69–93. doi:10.1007/978-3-031-64381-1_4

[18] Nicolas Desmoulins, Antoine Dumanois, Seyni Kane, and Jacques Traoré. 2025. Making BBS Anonymous Credentials eIDAS 2.0 Compliant. Cryptology ePrint Archive, Report 2025/619. https://eprint.iacr.org/2025/619

[19] Liam Eagen, Hy Ngo, Vikas Rushi, Ying Tong, Moven Tsai, and Janabel Xia. 2026. OpenAC: Open Design for Transparent and Lightweight Anonymous Credentials. *IACR Cryptol. ePrint Arch.* 2026 (2026), 251. https://eprint.iacr.org/2026/251

[20] Electronic Signatures and Trust Infrastructures (ESI). 2024. ETSI TR 119 476 V1.2.1: Analysis of Selective Disclosure and Zero-Knowledge Proofs Applied to Electronic Attestation of Attributes.

[21] Ethereum Foundation. 2023. *KZG Summoning Ceremony*. https://ceremony.ethereum.org/ Accessed: 2026-04-21.

[22] European Commission. 2026. EU Digital Identity Wallet – Architecture and Reference Framework. https://eudi.dev/2.4.0/. Accessed: 2026-04-27.

[23] Armando Faz-Hernández, Watson Ladd, and Deepak Maram. 2021. ZkAttest: Ring and Group Signatures on top of existing ECDSA keys. Cryptology ePrint Archive, Report 2021/1183. https://eprint.iacr.org/2021/1183

[24] Amos Fiat and Adi Shamir. 1987. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *CRYPTO'86 (LNCS, Vol. 263)*, Andrew M. Odlyzko (Ed.). Springer, Berlin, Heidelberg, 186–194. doi:10.1007/3-540-47721-7_12

[25] Centre for Research on Cryptography and Security. [n. d.]. P-256. https://std.neuromancer.sk/nist/P-256/. Accessed 2026-04-16.

[26] Karla Friedrichs, Franklin Harding, Anja Lehmann, and Anna Lysyanskaya. 2025. Device-Bound Anonymous Credentials With(out) Trusted Hardware. Cryptology ePrint Archive, Report 2025/1995. https://eprint.iacr.org/2025/1995

[27] Matteo Frigo and abhi shelat. 2024. Anonymous credentials from ECDSA. Cryptology ePrint Archive, Report 2024/2010. https://eprint.iacr.org/2024/2010

[28] Georg Fuchsbauer, Eike Kiltz, and Julian Loss. 2018. The Algebraic Group Model and its Applications. In *CRYPTO 2018, Part II (LNCS, Vol. 10992)*, Hovav Shacham and Alexandra Boldyreva (Eds.). Springer, Cham, 33–62. doi:10.1007/978-3-319-96881-0_2

[29] Ariel Gabizon and Zachary J. Williamson. 2019. The Turbo-PLONK Program Syntax for Specifying SNARK Programs. https://docs.zkproof.org/pages/standards/accepted-workshop3/proposal-turbo_plonk.pdf Preprint.

[30] Ariel Gabizon and Zachary J. Williamson. 2020. plookup: A simplified polynomial protocol for lookup tables. Cryptology ePrint Archive, Report 2020/315. https://eprint.iacr.org/2020/315

[31] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. 2019. PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge. Cryptology ePrint Archive, Report 2019/953. https://eprint.iacr.org/2019/953

[32] Jens Groth. 2016. On the Size of Pairing-Based Non-interactive Arguments. In *EUROCRYPT 2016, Part II (LNCS, Vol. 9666)*, Marc Fischlin and Jean-Sébastien Coron (Eds.). Springer, Berlin, Heidelberg, 305–326. doi:10.1007/978-3-662-49896-5_11

[33] Jens Groth, Markulf Kohlweiss, Mary Maller, Sarah Meiklejohn, and Ian Miers. 2018. Updatable and Universal Common Reference Strings with Applications to zk-SNARKs. In *CRYPTO 2018, Part III (LNCS, Vol. 10993)*, Hovav Shacham and Alexandra Boldyreva (Eds.). Springer, Cham, 698–728. doi:10.1007/978-3-319-96878-0_24

[34] Lovesh Harchandani. 2025. hhttps://github.com/docknetwork/crypto/tree/main/equality_across_groups.

[35] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. 2010. Constant-Size Commitments to Polynomials and Their Applications. In *ASIACRYPT 2010 (LNCS, Vol. 6477)*, Masayuki Abe (Ed.). Springer, Berlin, Heidelberg, 177–194. doi:10.1007/978-3-642-17373-8_11

[36] Darya Kaviani and Srinath Setty. 2025. Vega: Low-Latency Zero-Knowledge Proofs over Existing Credentials. Cryptology ePrint Archive, Report 2025/2094. https://eprint.iacr.org/2025/2094

[37] Abhiram Kothapalli. 2024. *A Theory of Composition for Proofs of Knowledge*. Ph. D. Dissertation. Carnegie Mellon University, USA. doi:10.1184/R1/25900222.V1

[38] Abhiram Kothapalli and Bryan Parno. 2023. Algebraic Reductions of Knowledge. In *CRYPTO 2023, Part IV (LNCS, Vol. 14084)*, Helena Handschuh and Anna Lysyanskaya (Eds.). Springer, Cham, 669–701. doi:10.1007/978-3-031-38551-3_21

[39] Anja Lehmann, Andrey Sidorenko, and Alexandros Zacharakis. 2025. Vision: A Modular Framework for Anonymous Credential Systems. Cryptology ePrint Archive, Report 2025/1981. https://eprint.iacr.org/2025/1981

[40] Tobias Looker, Vasilis Kalos, Andrew Whitehead, and Mike Lodder. 2025. *The BBS Signature Scheme*. Internet-Draft draft-irtf-cfrg-bbs-signatures-09. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-irtf-cfrg-bbs-signatures/09/ Work in Progress.

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

[42] Michele Orrù, George Kadianakis, Mary Maller, and Greg Zaverucha. 2025. Beyond the Circuit: How to minimize foreign arithmetic in ZKP circuits. *CiC* 2, 1 (2025), 23. doi:10.62056/an-4c3c2h

[43] Christian Paquin, Guru-Vamsi Policharla, and Greg Zaverucha. 2024. Crescent: Stronger Privacy for Existing Credentials. Cryptology ePrint Archive, Report 2024/2013. https://eprint.iacr.org/2024/2013

[44] Paolo De Rosa. 2024. Discussion comment on Cryptographers' Feedback on the EU Digital Identity's ARF #211. github.com. https://github.com/eu-digital-identity-wallet/eudi-doc-architecture-and-reference-framework/discussions/211/#discussioncomment-9882388
[45] Michael Rosenberg, Jacob D. White, Christina Garman, and Ian Miers. 2023. zkcreds: Flexible Anonymous Credentials from zkSNARKs and Existing Identity Infrastructure. In *2023 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 790–808. doi:10.1109/SP46215.2023.10179430
[46] Gil Segev. 2025. Bulletproofs for R1CS: Bridging the Completeness-Soundness Gap and a ZK Extension. Cryptology ePrint Archive, Report 2025/327. https://eprint.iacr.org/2025/327
[47] Srinath Setty. 2020. Spartan: Efficient and General-Purpose zkSNARKs Without Trusted Setup. In *CRYPTO 2020, Part III (LNCS, Vol. 12172)*, Daniele Micciancio and Thomas Ristenpart (Eds.). Springer, Cham, 704–737. doi:10.1007/978-3-030-56877-1_25
[48] Trusted Computing Group. 2013. TPM 2.0 Library Specification. https://trustedcomputinggroup.org/resource/tpm-library-specification/. Accessed: 2026-04-27.
[49] Riad S. Wahby, Ioanna Tzialla, abhi shelat, Justin Thaler, and Michael Walfish. 2018. Doubly-Efficient zkSNARKs Without Trusted Setup. In *2018 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 926–943. doi:10.1109/SP.2018.00060
[50] Anna P. Y. Woo, Alex Ozdemir, Chad Sharp, Thomas Pornin, and Paul Grubbs. 2025. Efficient Proofs of Possession for Legacy Signatures. In *2025 IEEE Symposium on Security and Privacy*, Marina Blanton, William Enck, and Cristina Nita-Rotaru (Eds.). IEEE Computer Society Press, 3291–3308. doi:10.1109/SP61157.2025.00080
[51] zaverucha. 2026. signature-proof. https://github.com/zaverucha/signature-proof. accessed 2026-04-22.
[52] ZKProof.org. 2018. ZKProof Charter. https://docs.zkproof.org/general.

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 [37, 38][11].

Definition A.1 (Reductions of Knowledge). Let $\mathcal{R}, \mathcal{R}'$ be ternary relations taking as input public parameters, a statement and a witness. A reduction of knowledge $\Pi : \mathcal{R} \to \mathcal{R}'$ 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 $\mathcal{P}, \mathcal{V}$, denoted as the prover and verifier respectively, 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 $\mathcal{A}$

$$\Pr\left[\begin{array}{c|c} (pp, \mathsf{x}, \mathsf{w}) \in \mathcal{R} & pp \leftarrow \mathsf{Gen}(1^\lambda) \\ \wedge & (\mathsf{x}, \mathsf{w}) \leftarrow \mathcal{A}(pp) \\ (pp, \mathsf{x}', \mathsf{w}') \notin \mathcal{R}' & (\mathsf{w}', \mathsf{x}') \leftarrow \langle \mathcal{P}(pp, \mathsf{x}, \mathsf{w}), \mathcal{V}(pp, \mathsf{x}) \rangle \end{array}\right] = 0$$

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 $\mathcal{P}^*, \mathcal{A}$

$$\Pr\left[\begin{array}{c|c} \phi(pp, \mathsf{x}, \mathsf{tr}) \neq \mathsf{x}' & pp \leftarrow \mathsf{Gen}(1^\lambda) \\ & (\mathsf{x}, \mathsf{st}) \leftarrow \mathcal{A}(pp) \\ & (\mathsf{w}', \mathsf{x}') \leftarrow \langle \mathcal{P}^*(pp, \mathsf{x}, \mathsf{st}), \mathcal{V}(pp, \mathsf{x}) \rangle \\ & \mathsf{tr} \leftarrow \mathsf{tr}\left(\langle \mathcal{P}^*(pp, \mathsf{x}, \mathsf{st}), \mathcal{V}(pp, \mathsf{x}) \rangle\right) \end{array}\right] = 0$$

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 reduction of knowledge 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 reduction of knowledge is *public coin*. Such protocols can be transformed to non-interactive via the Fiat-Shamir transform [9, 24].

Note that reductions of knowledge are more expressive than proofs of knowledge in the sense that the former can capture the latter. Indeed, we can view a proof of knowledge as a reduction of knowledge $\Pi : \mathcal{R} \to \mathcal{R}_\top$ where $\mathcal{R}_\top$ is the trivial relation that only accepts $\mathsf{x} = \mathsf{true}$ (capturing that the proof of knowledge verifier accepts) and reject all other values (capturing that the proof of knowledge verifier rejects).

We also recall the results for abstracting knowledge soundness for a large family of algebraic public coin protocols. These results are taken verbatim from [38] which in turn is an adaptation of the results of [12].

Definition A.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 Def. A.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 a prover message for round i; (2) each outgoing edge from layer $i \in [m]$ is labeled with a different choice of verifier randomness for round i; (3) each leaf is labeled with an accepting statement-witness pair output by the prover and verifier corresponding to the interaction along the path.

[11] We slightly modify the HVZK definition of [37] to capture the property against unbounded adversaries.

Lemma A.3. *Consider an m-round public-coin interactive protocol* $(\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ *that satisfies the interface described in Def. A.1 and completeness. Then* $(\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ *is a reduction of knowledge 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 verifier's randomness is sampled from space* Q *such that* $|Q| = O(2^\lambda)$*, and* $\prod_i n_i = \mathsf{poly}(\lambda)$.

Composing reductions of knowledge. We next recall the results of [37, 38] about composition of reductions of knowledge. We first present the definition for a relation pair.

Definition A.4. 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 next define sequential composition for reductions of knowledge.

Definition A.5. 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 reductions of knowledge. 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 $(\mathsf{x}_3, \mathsf{x}_4)$.

(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 reductions of knowledge. 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 A.6. *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 reductions of knowledge. Then, the sequential composition* $\Pi = \Pi_2 \circ \Pi_1$ *is a reduction of knowledge 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 reductions of knowledge. Then, the parallel composition* $\Pi = \Pi_2 \times \Pi_1$ *is a reduction of knowledge 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 [38, 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
$$\{ \ \mathsf{tr}_\mathsf{i} \mid \mathsf{tr}_\mathsf{i} \leftarrow \mathcal{S}_i(pp, \mathsf{x}_i) \ \}$$
$$\{ \ \widetilde{\mathsf{tr}_\mathsf{i}} \mid \widetilde{\mathsf{tr}_\mathsf{i}} \leftarrow \mathsf{tr}\left(\langle \mathcal{P}_i(pp, \mathsf{x}_i), \mathcal{V}_i(pp, \mathsf{x}_i) \rangle\right) \ \}$$
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
$$\{ \ \mathsf{tr}_1 \| \mathsf{tr}_2 \mid \mathsf{tr}_1 \| \mathsf{tr}_2 \leftarrow \mathcal{S}_i(pp, \mathsf{x}_1) \ \}$$
$$\{ \ \widetilde{\mathsf{tr}_1} \| \widetilde{\mathsf{tr}_2} \mid \widetilde{\mathsf{tr}_1} \| \widetilde{\mathsf{tr}_2} \leftarrow \mathsf{tr}\left(\langle \mathcal{P}(pp, \mathsf{x}_1), \mathcal{V}(pp, \mathsf{x}_1) \rangle\right) \ \}$$
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.

□

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 proof of knowledge Π_{EQ} to prove knowledge of two openings of a committed value under different commitment schemes. We present the construction in Fig. 5.

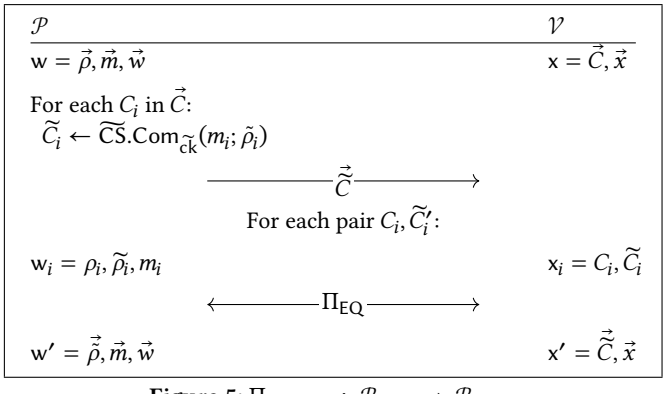

Figure 5: $\Pi_{\mathsf{CS} \to \widetilde{\mathsf{CS}}} : \mathcal{R}_{\mathsf{CS},\mathsf{P}} \to \mathcal{R}_{\widetilde{\mathsf{CS}},\mathsf{P}}$.

Theorem B.1. *Let* $\mathsf{CS}, \widetilde{\mathsf{CS}}$ *be a perfectly hiding commitment schemes with the same message spaceand* $\mathcal{M}$ *andlet* $\Pi_{\mathsf{EQ}} : \mathcal{R}_{\mathsf{EQ}} \to \mathcal{R}_\top$ *be a public coin HVZK proof of knowledge. Then, the protocol* $\Pi_{\mathsf{CS} \to \widetilde{\mathsf{CS}}}$

described in Fig. 5 is a public-coin HVZK reduction of knowledge $\Pi_{\mathsf{CS}\to\widetilde{\mathsf{CS}}} : \mathcal{R}_{\mathsf{CS,P}} \to \mathcal{R}_{\widetilde{\mathsf{CS}},\mathsf{P}}$.

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 Def. A.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 $\mathcal{R}_{\mathsf{CS,P}}$. 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 Thm. A.6, 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 $\mathsf{w}_{\mathsf{EQ}} := \vec{m}_{\mathsf{EQ}}, \vec{\rho}_{\mathsf{EQ}}, \vec{\tilde{\rho}}_{\mathsf{EQ}}$
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 $(m_{\mathsf{EQ},i}, \tilde{\rho}_{\mathsf{EQ},i}) = (m'_i, \rho'_i)$ 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.*

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 [42] 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:

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

We present a simple public coin HVZK proof of knowledge for the above relation in Fig. 6. The protocol is based on standard techniques and its security properties are well understood.

C.2 Equality of Pedersen Commitments Across Groups

We recall the protocol of [42] 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 provethe relation

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

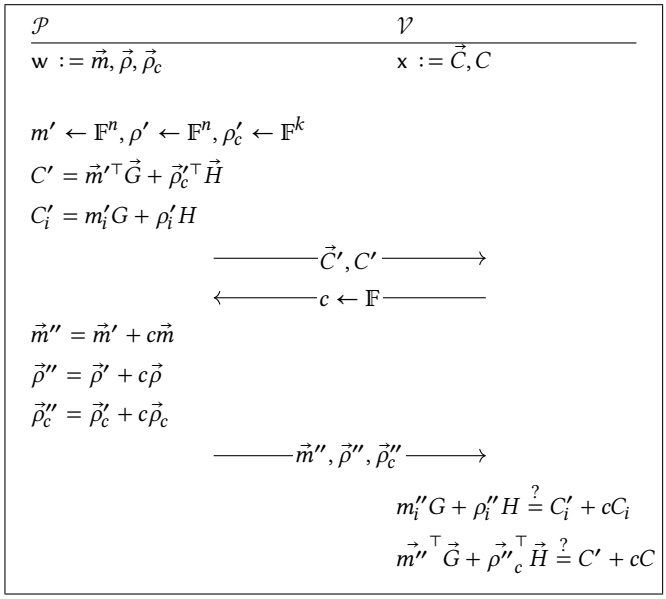

Figure 6: $\Pi_{\mathsf{PD}} : \mathcal{R}_{\mathsf{EQ}} \to \mathcal{R}_{\top}$.

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 Fig 7 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

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

and refer the reader to [42] for the security analysis.

D RoK for EC Point Addition

In this section we recall the construction of [17, 23] that we use to implement the RoK $\Pi_{\mathsf{SM}} : \mathcal{R}_{\mathsf{SM}} \to \mathcal{R}_{\top}$. We first recall how

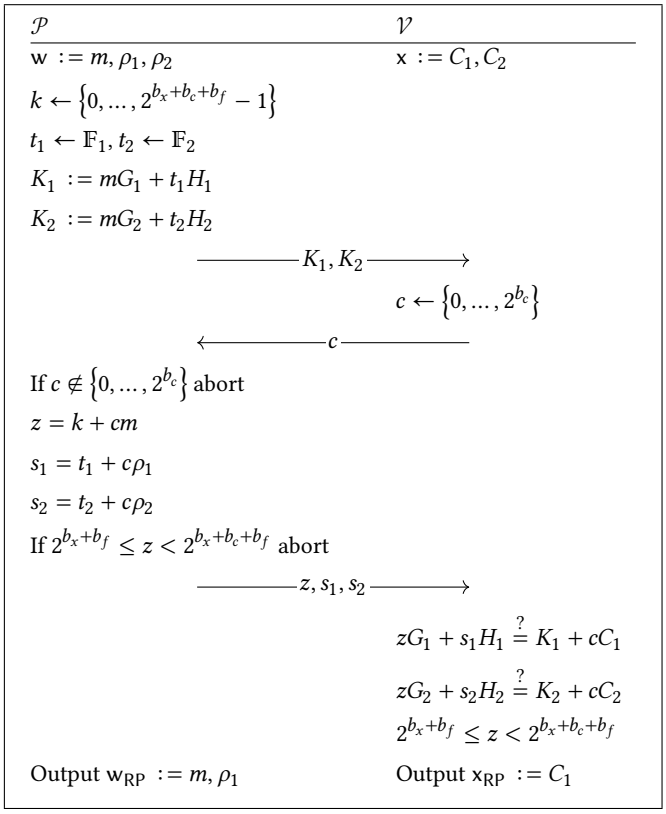

Figure 7: $\Pi_{\mathsf{PD},\mathbb{G}_1,\mathbb{G}_2} : \mathcal{R}_{\mathsf{EQ},\mathbb{G}_1,\mathbb{G}_2} \to \mathcal{R}_{\mathsf{RP}}$.

point addition over P-256 is defined. Given non-identity points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ s.t. $P_1 \neq P_2$ and $P_1 \neq -P_2$, the point $P_3 = P_1 + P_2$ is computed as

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2$$
$$y_3 = \frac{y_2 - y_1}{x_2 - x_1}(x_1 - x_3) - y_1$$

Setting $\tau = (y_2 - y_1)(x_2 - x_1)^{-1}$ we can express the above equations as a simple R1CS system over the base field of T-256 defined by the equations:

$$\begin{aligned} \tau(x_2 - x_1) &= y_2 - y_1 \\ \tau^2 &= x_1 + x_2 + x_3 \\ \tau(x_1 - x_3) &= y_1 + y_3 \end{aligned} \tag{3}$$

If these equations are satisfied and additionally $x_2 \neq x_1$[12] then indeed $P_3 = (x_3, y_3) = P_1 + P_2$. This can be turned to a proof of knowledge 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 reduction of knowledge. 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:

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

[12] This can also be expressed as the constraint $(x_2 - x_1)r = 1$

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

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

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[13].

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

We can define a reduction of knowledge

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

where we denote $\mathcal{R}_{\mathsf{MUL}}^3 = \mathcal{R}_{\mathsf{MUL}} \times \mathcal{R}_{\mathsf{MUL}} \times \mathcal{R}_{\mathsf{MUL}}$. We present the construction in Fig. 8.

$\mathcal{P}$		$\mathcal{V}$
$\mathsf{w} := \begin{pmatrix} x_1, y_1, x_2, y_2, x_3, y_3 \\ \rho_{x,1}, \rho_{y,1}, \rho_{x,2}, \\ \rho_{y,2}, \rho_{x,3}, \rho_{y,3} \end{pmatrix}$		$\mathsf{x} := \begin{pmatrix} C_{x,1}, C_{y,1} \\ C_{x,2}, C_{y,2} \\ C_{x,3}, C_{y,3} \end{pmatrix}$
$\tau := \frac{y_2 - y_1}{x_2 - x_1}$		
$\rho_\tau \leftarrow \mathbb{F}_p$ and $R := rK$		
$C_\tau = \tau G_t + \rho_\tau H_t$		
	$\xrightarrow{\qquad C_\tau \qquad}$	
$\mathsf{w}_1 := \begin{pmatrix} y_2 - y_1, \rho_{y,2} - \rho_{y,1} \\ x_2 - x_1, \rho_{x,2} - \rho_{x,1} \\ \tau, \rho_\tau \end{pmatrix}$		$\mathsf{x}_1 := \begin{pmatrix} C_{y,2} - C_{y,1}, \\ C_{x,2} - C_{x,1}, \\ C_\tau \end{pmatrix}$
$\mathsf{w}_2 := \begin{pmatrix} \tau, \rho_\tau \\ \tau, \rho_\tau \\ x_1 + x_2 + x_3, \\ \rho_1 + \rho_2 + \rho_3 \end{pmatrix}$		$\mathsf{x}_2 := \begin{pmatrix} C_\tau, C_\tau \\ C_{x,1} + C_{x,2} + C_{x,3} \end{pmatrix}$
$\mathsf{w}_3 := \begin{pmatrix} \tau, \rho_\tau \\ x_1 - x_3, \rho_{x,1} - \rho_{x,1} \\ y_1 - y_3, \rho_{y,1} - \rho_{y,1} \end{pmatrix}$		$\mathsf{x}_3 := \begin{pmatrix} C_\tau \\ C_{x,1} - C_{x,3} \\ C_{y,1} - C_{y,3} \end{pmatrix}$
$\mathsf{w}_{\mathsf{PD}} := \big(y_1, \rho_{y,1}\big)$		$\mathsf{x}_{\mathsf{PD}} := \big(C_{y,1}\big)$
$\mathsf{w}_{\mathsf{NZ}} := \big(x_2 - x_1, \rho_{x,2} - \rho_{x,1}\big)$		$\mathsf{x}_{\mathsf{NZ}} := \big(C_{x,2} - C_{x,1}\big)$
$\mathsf{w} = (\mathsf{w}_1, \mathsf{w}_2, \mathsf{w}_3, \mathsf{w}_{\mathsf{PD}}, \mathsf{w}_{\mathsf{NZ}})$		$\mathsf{x} = (\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3, \mathsf{x}_{\mathsf{PD}}, \mathsf{x}_{\mathsf{NZ}})$

Figure 8: $\Pi_{\mathsf{PA}^*} \ : \ \mathcal{R}_{\mathsf{PA}^*} \to \mathcal{R}_{\mathsf{MUL}}^3 \times \mathcal{R}_{\mathsf{PD}} \times \mathcal{R}_{\mathsf{NZ}}$.

Theorem D.1. *The protocol Π_{PA^*} described in Fig. 8 is a public-coin HVZK reduction of knowledge $\Pi_{\mathsf{PA}^*} \ : \ \mathcal{R}_{\mathsf{PA}^*} \to \mathcal{R}_{\mathsf{MUL}}^3 \times \mathcal{R}_{\mathsf{PD}} \times \mathcal{R}_{\mathsf{NZ}}$.*

Proof. (sketch) Completeness and public reducibility follow by construction. HVZK holds since the only message communicated is the commitment C_τ which is perfectly simulatable. For knowledge soundness, assume we have a valid witness for the composed reduced statement. We consider two cases: (1) there exists a commitment for which the witness contains two different openings for it and (2) all the openings are "consistent", i.e. there exists one unique opening for each commitment. In the latter case, if the openings P_1, P_2 correspond to elements of $\mathbb{G}_p$, a straightforward calculation shows that the equations in Eq. 3 are satisfied by the witness, which implies that indeed $C_{3,x}, C_{3,y}$ opens to $(x_3, y_3) = P_3 = P_1 + P_2$. Furthermore, the proof for $\mathcal{R}_{\mathsf{NZ}}$ guarantees that the x-coordinate of the points P_1 and P_2 are different, therefore $P_1 \neq P_2$ and $P_1 \neq -P_2$. The first case can only happen with negligible probability due to the binding property of the commitment scheme. □

The target relation can then be proven using standard techniques, in particular, simple Schnorr type proofs of knowledge (see for example [49]). We note that the statement can be proven by executing the protocols in parallel[14], i.e. use the same challenge for all which results in a Schnorr-type proof. We denote the resulting protocol $\Sigma_{\mathsf{PA}^*, \mathcal{C}}$ where $\mathcal{C}$ denotes the challenge space.

E RoK for EC Scalar Multiplication

In this section we recall the construction of [17, 23] 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 imporve its efficiency. The relation we want to prove is

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

and we consider the case where we use the Pedersen commitment over T-256 as the commitment scheme $\widetilde{\mathsf{GCS}}$. 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 $\Sigma_{\mathsf{SM}^*, \{0,1\}}$ in parallel. We describe the protocol in Fig. 9.

Theorem E.1. *The protocol Π_{SM^*} described in Fig. 9 is a public-coin HVZK reduction of knowledge $\Pi_{\mathsf{SM}} \ : \ \mathcal{R}_{\mathsf{SM}} \to \mathcal{R}_{\top}$. The notation $\mathbb{F}_q$ refers to the scalar field of* T-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[15]. 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 $\sigma_{\mathsf{PA}^*, \{0,1\}}$ to create a simulated transcript for the

[13] In Π_Σ the points correspond to the public key Q and a public point $\alpha \mathrm{G}_p$ committed by the verifier as well.

[14] Note that is different than parallel composition of RoKs as defined in Def.A.1

[15] As noted in [17] the protocol achieves statistical (and not perfect) HVZK since the message ω is sampled from $\mathbb{F}_q \setminus \{0, z, 2z\}$ instead of $\mathbb{F}_q$.

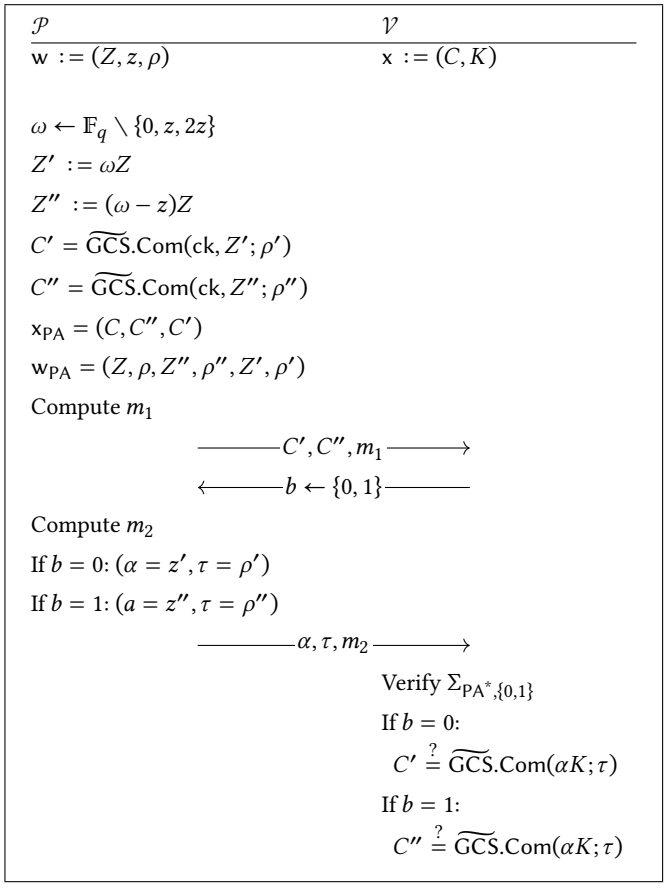

Figure 9: $\Pi_{\mathsf{SM}} : \mathcal{R}_{\mathsf{SM}} \to \mathcal{R}_{\top}$. **In the above,** m_1, m_2 **denote the first and second prover message in an execution of** $\Sigma_{\mathsf{PA}^*,\{0,1\}}$.

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 $\sigma_{\mathsf{PA}^*,\{0,1\}}$ to create a simulated transcript for the case of $b = 0$. 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 $\mathbb{F} \setminus \{z, 2z\}$[16] 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 A.3 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

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

Since the $\Sigma_{\mathsf{PA}^*,\{0,1\}}$ verifier accepts in both cases, we can extract openings for $\mathsf{x}_{\mathsf{PA}^*} = (C, C'', C')$ by using the transcripts

$$\mathsf{tr}_{\mathsf{PA}^*,0} = (m_1, 0, m_{2,0})$$
$$\mathsf{tr}_{\mathsf{PA}^*,1} = (m_1, 1, m_{2,1})$$

Let Z, Z', Z'' be the openings of the commitments and note that they satisfy $Z + Z'' = Z'$. By the second verification test, we have that

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

We claim that the discrete logarithm of Z w.r.t. K is $\alpha_0 - \alpha_1$ 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.

□

F 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 [26].

Blinding the public key and precomputation of PoPs. Desmoulins et al. [18] 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* public key by only accessing the secure element in a black-box way. Intuitively, consider the ECDSA verification equation[17,18]

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

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

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

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 (pre-hashed) signature on the message $\mathsf{h} = r^{-1}\mathsf{H}(\mathsf{n}, Q_B)$. The SE will

[16] 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.

[17] We present their approach from the lens of the alternative ECDSA verification algorithm presented in Sec. 2.1.

[18] As noted in [18], to achieve proof-of-possession with this technique, the public key must be included in the digest to prevent related key attacks [10]. 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.

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:

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

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:

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

Noting that the proven predicate is equivalent to $r^{-1}Q_B = Q$, the above relation is a subset of $\mathcal{R}_{\text{C-DLOG}}$ 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 $\mathcal{R}_{\text{re-rand}}$, 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[19]. 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 [26]. We defer such analysis to future work.

[19]While android supports it, it is marked as deprecated and it will be removed [8].

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

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

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.