Anchored Merkle Range Proof for Pedersen Commitments

Leona Hioki paper@intmax.io

19.09.2025

Abstract

We present a simple range-proof mechanism for Pedersen commitments that avoids per-transaction heavy ZK verification and pairings. The idea is to commit once to a *Merkleized range table* of points $\{(U,\, aX{\cdot}G)\}_{X\in\{1,\ldots,2^n\}}$ for a secret $a \in \mathbb{Z}_q$ and a public anchor $U = a{\cdot}B$. At transaction time, a prover shows set membership of the leaf $(U,\, ax{\cdot}G)$, proves via a Chaum–Pedersen DLEQ that $\log_B U = \log_C C'$ where $C' = a\cdot C$ and C is the Pedersen commitment, and finally proves (Schnorr) that $C' - (ax{\cdot}G)$ lies in the H-direction. These three checks enforce x to be the in-range value indexed by the Merkle leaf while preserving privacy. Verification costs a single Merkle proof plus a DLEQ and a Schnorr discrete-log proof over an elliptic curve group.

Keywords. Pedersen commitment, range proof, set membership, Merkle tree, DLEQ, Chaum–Pedersen, Schnorr, EVM gas.

1 Model and Preliminaries

Let $(\mathbb{G}, +)$ be a prime-order EC group of order q with independent generators $G, H, B \in \mathbb{G}$ (the discrete-log relations among them are unknown). We use additive notation: a Pedersen commitment to $x \in \mathbb{Z}_q$ with blinding $r \in \mathbb{Z}_q$ is

$$C = x{\cdot}G + r{\cdot}H \in \mathbb{G}.$$

Fix a target range $\mathcal{R} = \{1, 2, \ldots, 2^n\}$. Hash denotes a collision-resistant hash (for Merkle), and $\mathcal{H}$ a random oracle for Fiat–Shamir (FS). Pedersen commitments are perfectly hiding; binding follows from the independence of G, H and DL hardness [6].

Statements of knowledge. We use two standard Σ-protocols (FS-NIZKs): (i) Chaum–Pedersen *DLEQ* for equality of discrete logs: prove $\log_B U = \log_C C'$ [1, 9], and (ii) Schnorr proof for DL in base H: prove $R = t\cdot H$ for public R [2]. Both are honest-verifier ZK and knowledge-sound; under FS in the ROM they become NIZKs.

2 Construction

We formalize three algorithms ($\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify}$).

2.1 Setup (one-time by a prover)

1. Sample $a \leftarrow \mathbb{Z}_q$ and set the public *anchor* $U := a{\cdot}B$.
2. For each $X \in \{1, \ldots, 2^n\}$, define the leaf payload

$$\mathsf{Leaf}_X := \big(U,\, aX{\cdot}G\big) \in \mathbb{G} \times \mathbb{G},$$

and build a Merkle tree over the serialization of all Leaf_X in canonical order, obtaining root R.

3. Produce a one-time NIZK Π_{setup} that there exists a *single* $a \in \mathbb{Z}_q$ such that *all* leaves equal $(U, aX \cdot G)$ for $X = 1, \ldots, 2^n$, where $U = a \cdot B$ (binds the entire table and anchor to the same a).

4. Publish $\mathsf{pp} = (R, n, B, U, \Pi_{\text{setup}})$; keep a secret for future proofs.

2.2 Prove

On input pp and witness (x, r) with $x \in \mathbb{Z}_q$:

1. Form $C = x \cdot G + r \cdot H$ and $C' := a \cdot C$.

2. Compute the leaf $\mathsf{Leaf}_x = (U,\ ax \cdot G)$ and Merkle path π proving $\mathsf{Leaf}_x \in R$.

3. Set $R_H := C' - (ax \cdot G)$.

4. Produce two FS-NIZKs:
   - *DLEQ* Π_{dleq}: prove $\log_B U = \log_C C'$.
   - *Schnorr* Π_H: prove knowledge of t s.t. $R_H = t \cdot H$.

5. Output $\Pi = (C, C', \mathsf{Leaf}_x, \pi, \Pi_{\mathsf{dleq}}, \Pi_H)$.

2.3 Verify

1. (One-time) Check Π_{setup} for (R, B, U) and store R.

2. Recompute Leaf_x from the proof and verify $\mathsf{MerkleVerify}(R, \mathsf{Leaf}_x, \pi) = 1$.

3. Verify DLEQ Π_{dleq} that $\log_B U = \log_C C'$.

4. Compute $R_H := C' -$ (second component of Leaf_x) and verify Schnorr Π_H that $R_H \in \langle H \rangle$.

5. Accept iff all checks pass.

Correctness. Honest executions satisfy $C' = a \cdot C = (ax) \cdot G + (ar) \cdot H$, $\mathsf{Leaf}_x = (U, ax \cdot G)$ and $R_H = (ar) \cdot H$, so both subproofs and the Merkle check pass.

3 Security

We prove that the protocol in Section 2 is a NIZK argument that a Pedersen commitment C opens to some $X \in \mathcal{R} = \{1, \ldots, 2^n\}$, and that the proof leaks nothing beyond this membership (modulo whether the set-membership itself is hiding). Our reductions make the use of Chaum–Pedersen DLEQ explicit: it ties the scaling factor used in the anchor U and in C' so that the Schnorr proof forces C to open to the same in-range value as the Merkle leaf.

Assumptions. We work in a prime-order group $(\mathbb{G}, +)$ of order q. Let $G, H, B \in \mathbb{G}$ be fixed generators with unknown discrete-log relations (standard assumption for Pedersen binding). We assume: (A1) *DL hardness* in $\mathbb{G}$; (A2) *FS-in-ROM knowledge soundness* of Chaum–Pedersen DLEQ and Schnorr Σ-protocols (obtained by special soundness + forking lemma [1, 2, 9, 10]); (A3) collision resistance of the Merkle hash Hash; (A4) soundness of the one-time setup proof Π_{setup} that binds *all* leaves and the anchor to a single $a \in \mathbb{Z}_q$. We do *not* need to assume a known linear independence between G and H (which is false in a cyclic group); instead we rely only on (A1)–(A4). We also assume $2^n < q$ (no wrap-around ambiguity).

3.1 Relations proved and extracted witnesses

Let the public input be (R, n, B, U, C, C') and the proof contain $(\mathsf{Leaf}_X, \pi, \Pi_{\mathsf{dleq}}, \Pi_H)$ where $\mathsf{Leaf}_X = (U, aX\cdot G)$ for some leaf index X and Merkle path π. Define $R_H := C' - (aX\cdot G)$ (the verifier recomputes this from the proof).

Lemma 1 (Black-box extractor for an accepting proof)**.** *Assume (A2) and that* Verify *accepts. Then there exists a PPT extractor $\mathcal{E}$ which, by rewinding the FS challenges, outputs numbers $a, t \in \mathbb{Z}_q$ such that*

$$U = a\cdot B, \quad C' = a\cdot C, \quad R_H = t\cdot H.$$

Moreover, by (A4) and Merkle verification, the leaf equals $\mathsf{Leaf}_X = (U, aX\cdot G)$ *for some* $X \in \{1, \ldots, 2^n\}$.

Proof. Knowledge soundness in ROM for Chaum–Pedersen DLEQ yields a with $U = a\cdot B$ and $C' = a\cdot C$ from Π_{dleq}. Knowledge soundness in ROM for Schnorr on base H yields t with $R_H = t\cdot H$ from Π_H. Soundness of Π_{setup} together with the accepted Merkle path implies that the leaf has the stated form for some $X \in \mathcal{R}$. □

3.2 Range soundness (membership)

Theorem 1 (Range soundness reduced to (A2)+(A4))**.** *Under (A2) and (A4), for any PPT adversary $\mathcal{A}$ that outputs an accepting proof, the extractor of Lemma 1 produces $X \in \{1, \ldots, 2^n\}$ and $r' \in \mathbb{Z}_q$ such that $C = X\cdot G + r'\cdot H$. Equivalently, the verified statement is exactly that the committed value lies in $\mathcal{R}$.*

Proof. From Lemma 1, $C' = a\cdot C$, $R_H = t\cdot H$, and the accepted leaf is $(U, aX\cdot G)$ with the *same* a as in $U = a\cdot B$. By verifier recomputation,

$$R_H \;=\; C' - (aX\cdot G) \;=\; a\cdot(C - X\cdot G).$$

Since $a \in \mathbb{Z}_q$, multiplying both sides by a^{-1} gives $C - X\cdot G \;=\; (a^{-1}t)\cdot H$. Setting $r' := a^{-1}t$ yields $C = X\cdot G + r'\cdot H$ as claimed. Thus any accepting proof certifies that C opens to an in-range value X. □

Tightness and the role of DLEQ. Without DLEQ, the prover could take *different* scalars a_U and a_C, use $(U, a_U\cdot B)$ in the leaf but set $C' = a_C\cdot C$ so that $C' - (a_U X\cdot G)$ accidentally lands on the H-line. A Schnorr proof would then be trivial to produce by choosing C accordingly, while C need not open to X. The DLEQ prevents this by enforcing $a_U = a_C = a$ in ROM, making the above algebraic manipulation impossible unless C truly opens to X.

3.3 Unforgeability against out-of-range openings

Theorem 2 (No out-of-range opening unless (A2) or (A4) fails)**.** *Let* $_{\text{rng}}$ *be the maximum success probability of any PPT adversary in producing an accepting proof for some C that* does not *open to any $X \in \mathcal{R}$. Then*

$$_{\text{rng}} \;\leq\; {}^{\mathsf{sound}}_{\Pi_{setup}} + {}^{\mathit{know}}_{\mathit{DLEQ}} + {}^{\mathit{know}}_{\mathit{Schnorr}} + (\lambda),$$

where the three advantages are the standard (ROM) soundness/knowledge advantages of the cited sub-proofs and λ is the security parameter.

Proof sketch. If Π_{setup} fails to be sound, the adversary can plant inconsistent leaves; this is captured by ${}^{\mathsf{sound}}_{\Pi_{\text{setup}}}$. Else, by Lemma 1 we extract (a, t) and an in-range X with $C = X\cdot G + a^{-1}t\cdot H$, contradicting the premise that C has no such opening. Therefore a successful forgery implies breaking DLEQ-knowledge or Schnorr-knowledge soundness in ROM. Union bound yields the inequality. □

3.4 Privacy / Zero-knowledge

We separate (i) leakage from the *set-membership channel* (the leaf and its opening) and (ii) leakage from the algebraic ties (DLEQ & Schnorr).

Theorem 3 (Zero-knowledge of algebraic ties in ROM)**.** *Under (A2), there exists a PPT simulator $\mathcal{S}$ which, given public (R, B, U) and a* chosen *group element C' (consistent with the statement format), outputs simulated transcripts $(\Pi_{\mathsf{dleq}}, \Pi_H)$ that are computationally indistinguishable from honestly generated ones, without knowing a or t. In particular, the DLEQ and Schnorr subproofs leak no information about a, t, r, or X beyond the fact that the verifier already sees U, C', and $R_H = C' -$ (leaf second component).*

Proof. Chaum–Pedersen and Schnorr are honest-verifier ZK Σ-protocols; their Fiat–Shamir transforms are simulatable in the ROM by programming the oracle [1, 2, 9, 10]. □

Theorem 4 (Range-privacy up to membership-channel leakage)**.** *Assume (A1)–(A3). Fix public (R, B, U) and a commitment C. Consider two values $X_0, X_1 \in \mathcal{R}$ and the corresponding honest proofs. If the set-membership mechanism (Merkle or its replacement) is* hiding *(i.e., it does not reveal the index/leaf beyond membership), then the resulting full transcripts are computationally indistinguishable. If a plain Merkle path is used (which can reveal the leaf position), then the only information about X_b leaked by the transcript is the explicit leaf payload $(U, aX_b \cdot G)$ and any position bits in π. Under (A1), recovering X_b from $(U, aX_b \cdot G)$ is as hard as DL w.r.t. the unknown base $a \cdot G$.*

Proof sketch. By Theorem 3, the algebraic subproofs are simulatable and thus reveal nothing beyond their public statements. The commitment C is perfectly hiding, and C' is a scalar multiple of C by an unknown a proven only via ZK DLEQ; hence (C, C') leak nothing about X beyond membership. The residual leakage is precisely the membership channel. □

Remark on binding/uniqueness. Pedersen commitments are computationally binding under (A1) when the DL between G and H is unknown; thus once Theorem 1 concludes that $C = X \cdot G + r' \cdot H$ for some $X \in \mathcal{R}$, an adversary cannot (except with negligible probability) also open C to $X' \neq X$.

3.5 Putting it together

Theorem 5 (Main security theorem)**.** *Under (A1)–(A4) in the ROM, the protocol* $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ *is a non-interactive zero-knowledge* argument of set-membership for Pedersen commitments*: for every accepting proof, the committed value lies in $\mathcal{R}$ (Theorem 1), and the transcript leaks no information beyond the membership claim and any information explicitly revealed by the chosen set-membership primitive (Theorem 4). Any successful out-of-range forgery breaks either the setup soundness, the DLEQ knowledge-soundness, or the Schnorr knowledge-soundness (Theorem 2).*

4 Efficiency and Gas Estimates

Asymptotics. Verification uses one Merkle proof (depth n), one DLEQ (two bases), and one Schnorr (base H). No pairings and no large multi-scalar multiplications are needed on-chain.

Concrete gas on EVM (indicative).

- **Merkle:** Keccak-256 costs $30 + 6 \cdot \text{words}$ gas per call. Each internal node typically hashes 64 bytes (two 32-byte children), i.e., ≈ 42 gas per level, thus $\approx 42n$ gas for depth n (ignoring memory expansion); for $n = 32$, about 1,344 gas [27].

- **EC ops (alt_bn128 precompiles):** With EIP-1108, $\mathsf{ECADD} = 150$ gas, $\mathsf{ECMUL} = 6{,}000$ gas [26]. A straightforward verifier uses

$$\text{DLEQ}: \; 4\,\mathsf{ECMUL} + 2\,\mathsf{ECADD}, \qquad \text{Schnorr}: \; 2\,\mathsf{ECMUL} + 1\,\mathsf{ECADD},$$

  totaling $6\,\mathsf{ECMUL} + 3\,\mathsf{ECADD} \approx 36{,}450$ gas for curve math.

- **Call/overheads:** Each precompile call pays the CALL base and warm-access/memory overhead (*post-Berlin* precompiles are warm);gas [28].

Overall, a single range verification is typically in the ballpark of 40k $\sim$ 50k gas on Ethereum mainnet, plus calldata/memory effects.

5 NIZK Setup Optimizations

The one-time NIZK for generating the full 2^n-range Merkle tree may be expensive if n is large. We outline two practical techniques to shrink n while preserving utility:

(1) Checkpoint embedding. Construct a smaller tree for $\{1, \ldots, 2^{n_0}\}$, with $n_0 \ll n$, and embed additional *checkpoint* leaves that are random-looking points corresponding to amounts outside this base range. The verifier allows the prover to present multiple membership proofs whose leaves add linearly on-chain. This realizes larger ranges by combining several verified leaves.

(2) On-chain scalar mixing. Permit the verifier to multiply verified leaves by small public scalars (e.g., $< 2^{10}$) and add them before comparing to C'. Because the leaves are opaque group elements and the discrete log is hard, predicting external amounts from such linear combinations is computationally difficult, while enabling a compact base tree to span wide effective ranges. Both techniques rely on the linearity of $\mathbb{G}$ (closure under addition and scalar multiplication) and preserve soundness provided the contract enforces that the final combined G-component cancels exactly against C' (i.e., remains in the H-direction) and all constituent membership proofs are valid.

6 Related Work

Classical interval proofs include Boudot's exact and efficient range proof [7], and DL-setting range/membership proofs by Camenisch–Chaabouni–shelat [8]. Modern short proofs leverage inner-product arguments [11] and Bulletproofs [12], widely used in CT/RingCT systems [13, 14, 15]. Set-membership via accumulators and their dynamic/bilinear variants are well-studied [16, 17, 18], with comprehensive recent surveys [24, 25]. Vector commitments (VC) [19] and KZG polynomial commitments [20] underpin Verkle trees [21, 22], offering alternative membership structures with succinct openings (though pairing-based).

7 Discussion and Practical Notes

- **One-time setup.** Π_{setup} binds (R, B, U) and all leaves to the same a; it can be verified once at deployment.

- **Privacy.** Avoid revealing $ax \cdot G / ar \cdot H$ beyond what the NIZKs require; transcripts are simulatable in ROM.

- **Linkability.** Using a fixed root R (thus fixed U) across epochs can create linkage; rotate a/R per epoch.

- **Domain separation.** Include contract address, chain id, (R, B, U), (C, C'), R_H in FS challenges.

References

[1] D. Chaum and T. P. Pedersen. Wallet Databases with Observers. *CRYPTO 1992*. URL: https://www.iacr.org/cryptodb/.

[2] C.-P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 1991.

[3] S. Micali, M. Rabin, J. Kilian. Zero-Knowledge Sets. *FOCS 2003*.

[4] D. Catalano, M. Di Raimondo, D. Fiore, M. Messina. Mercurial Commitments with Applications to Zero-Knowledge Sets. *SCN 2006*.

[5] D. Benarroch et al. Zero-knowledge proofs for set membership: efficient, succinct, modular. *Designs, Codes and Cryptography*, 2023.

[6] T. P. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. *CRYPTO 1991*. https://link.springer.com/chapter/10.1007/3-540-46766-1_9.

[7] F. Boudot. Efficient Proofs that a Committed Number Lies in an Interval. *EUROCRYPT 2000*. https://www.iacr.org/archive/eurocrypt2000/1807/18070437-new.pdf.

[8] J. Camenisch, R. Chaabouni, A. shelat. Efficient Protocols for Set Membership and Range Proofs. *ASIACRYPT 2008*. https://www.iacr.org/archive/asiacrypt2008/53500238/53500238.pdf.

[9] J. Camenisch, M. Stadler. Proof Systems for General Statements about Discrete Logarithms. 1997. https://crypto.ethz.ch/publications/files/CamSta97b.pdf.

[10] E. Bangerter, J. Camenisch, M. Lysyanskaya. Efficient Proofs of Knowledge of Discrete Logarithms and Representations. *PKC 2005*. https://www.iacr.org/archive/pkc2005/33860155/33860155.pdf.

[11] J. Bootle, A. Cerulli, P. Chaidos, J. Groth, C. Petit. Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting. *EUROCRYPT 2016*. https://ora.ox.ac.uk/objects/uuid:2f919864-a097-48ce-9a28-2b9dc3e6382d.

[12] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, G. Maxwell. Bulletproofs: Short Proofs for Confidential Transactions and More. *IEEE S&P 2018*. https://cryptopapers.info/assets/pdf/bulletproofs.pdf.

[13] G. Maxwell. Confidential Transactions. 2015 (plain text note). https://people.xiph.org/~greg/confidential_values.txt.

[14] S. Noether et al. Ring Confidential Transactions. *Ledger*, 2016. https://www.ledgerjournal.org/ojs/ledger/article/view/34.

[15] A. Poelstra et al. Confidential Assets. *FC 2017*. https://blockstream.com/bitcoin17-final41.pdf.

[16] J. C. Benaloh, M. de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Signatures. *EUROCRYPT 1993*. https://dl.acm.org/doi/10.1007/3-540-48285-7_24.

[17] J. Camenisch, A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. *CRYPTO 2002*. https://cs.brown.edu/people/alysyans/papers/camlys02.pdf.

[18] J. Camenisch, M. Kohlweiss, A. Shoup. An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. *PKC 2009*. https://www.iacr.org/archive/pkc2009/54430486/54430486.pdf.

[19] D. Catalano, D. Fiore. Vector Commitments and Their Applications. *PKC 2013*. https://www.iacr.org/archive/pkc2013/77780054/77780054.pdf.

[20] A. Kate, G. Zaverucha, I. Goldberg. Constant-Size Commitments to Polynomials and Their Applications. *ASIACRYPT 2010*. https://www.iacr.org/archive/asiacrypt2010/6477178/6477178.pdf.

[21] J. Kuszmaul. Verkle Trees. 2018 preprint. https://math.mit.edu/research/highschool/primes/materials/2018/Kuszmaul.pdf.

[22] V. Buterin. Verkle trees (explainer). 2021 blog post. https://vitalik.eth.limo/general/2021/06/18/verkle.html.

[23] D. Boneh, B. Bünz, B. Fisch. Batching Techniques for Accumulators with Applications to IOPs and Stateless Blockchains. *CRYPTO 2019*. https://dl.acm.org/doi/10.1007/978-3-030-26948-7_20.

[24] I. Ozcelik, S. S. Huang, A. Joulin. An Overview of Cryptographic Accumulators. *ICS 2021*. https://arxiv.org/abs/2103.04330.

[25] M. Loporchio et al. A survey of set accumulators for blockchain systems. *Journal of Computer Security*, 2023. https://www.sciencedirect.com/science/article/pii/S1574013723000370.

[26] EIP-1108: Reduce alt_bn128 precompile gas costs. https://eips.ethereum.org/EIPS/eip-1108.

[27] Ethereum StackExchange: What is the gas cost of keccak256? https://ethereum.stackexchange.com/questions/64026/what-is-the-gas-cost-of-keccak256abc.

[28] EVM Codes: Precompiled Contracts (notes on warm addresses post-Berlin). https://www.evm.codes/precompiled.