BRaccoon: Concurrently Secure Blind Lattice Signatures from Raccoon

Lucjan Hanzlik[1], Mark Manulis[2], Marzio Mula[2], Alan Pulval-Dady[2], Tjerand Silde[3], and Daniel Slamanig[2]

[1] CISPA Helmholtz Center for Information Security, Germany, hanzlik@cispa.de
[2] Research Institute CODE, Universität der Bundeswehr München, Germany, firstname.lastname@unibw.de
[3] Norwegian University of Science and Technology, Norway, tjerand.silde@ntnu.no

Abstract. Blind signatures are a central primitive for privacy-preserving applications such as e-cash, anonymous credentials, and e-voting.

Classical constructions such as blind Schnorr adapt an ordinary signature scheme while producing signatures that follow the same format and distribution as those generated by the ordinary signing algorithm. This property is particularly relevant in settings where blind and non-blind signatures coexist within the same infrastructure. Achieving an analogous result for lattice-based Fiat–Shamir signatures, with security based on the underlying ordinary signature scheme, has remained open.

We present BRaccoon, the first concurrently secure lattice-based realization of the "blind signatures from a signature assumption" paradigm of Fuchsbauer and Wolf (EUROCRYPT 2024). Our construction builds on the rejection-free lattice signature scheme Raccoon, yielding signatures that retain the algebraic format and verification relation of Raccoon.

At a high level, we introduce blinding at the commitment stage and enforce correct challenge and response generation via linearly homomorphic encryption combined with NIZK proofs. Since these proofs already bind the hidden message to the resulting signature, their relation can naturally incorporate predicates on that message, providing a direct route toward predicate blind signatures.

A central technical challenge stems from discrete Gaussian sampling, where blinding induces a non-trivial distributional shift that precludes direct security reductions. To overcome this, we introduce a modified scheme Raccoon* that explicitly captures this shift. We analyze one-more unforgeability through non-rounded versions of Raccoon* and Raccoon. The deployed blind-signing protocol retains rounding.

For a concrete instantiation, we encode the lattice relations and the non-linear challenge computation in a common mixed Boolean and arithmetic relation. Our instantiation retains SHA3-256 and SHAKE256 and proves both protocol statements using the ZK-LaBRADOR toolkit. For up to 2^{32} signing queries, its total communication is estimated at approximately 862 KB.

Keywords: Blind Signature · Lattices · Concurrent Security

1 Introduction

Blind signatures, introduced by Chaum in the 1980s [Cha82], are an interactive variant of digital signatures with the key property that the signer learns nothing about the message being signed, while the transcript of the signing protocol remains unlinkable to the resulting signature. These features make blind signatures a fundamental building block for privacy-preserving applications such as e-cash [Cha82], e-voting [IKSA03], and anonymous credential systems [BL13]. In recent years, they have also seen increasing practical deployment, for example in systems such as TumbleBit [HAB+17] and Privacy Pass [DGS+18].

Blind signatures can be constructed under a variety of classical hardness assumptions, including RSA [Cha82], discrete logarithms [PS00, BL13, HKL19, CAHL+22], and bilinear pairings [Bol03,

FHKS16, HLW23]. However, Shor's algorithm and ongoing advances in quantum computing increasingly threaten such assumptions, creating strong demand for efficient constructions based on quantum-resistant primitives. Despite this urgency, blind signatures in the post-quantum setting remain significantly less mature and less well understood than their classical counterparts. Addressing this gap is the focus of this work.

Desirable properties. Before proceeding, we highlight properties that are particularly relevant for the practical deployment of blind signature schemes. In typical applications, multiple signing sessions may occur concurrently and interleave with other protocol executions. Consequently, it is essential for such schemes to provide *concurrent security*. Notably, this property is achieved automatically by protocols with only two moves, but is non-trivial otherwise.

While simple and efficient two-move constructions exist under classical assumptions, thanks to their favorable algebraic structure, such as those based on RSA [Cha82] or bilinear pairings [Bol03, FHKS16, HLW23], no analogous solutions are known in the post-quantum setting. The only practical approach to obtaining two-move blind signatures in this regime is to follow Fischlin's generic transformation [Fis06]. Given an ordinary (non-blind) digital signature scheme, which we call the *underlying scheme*, Fischlin's transformation produces a blind signature protocol whose final output is a non-interactive zero-knowledge (NIZK) proof of a valid signature obtained from the underlying scheme. This paradigm has proven particularly successful for lattice-based constructions [AKSY22, dPK22, BLNS23a], and has recently been extended to other families following the hash-and-sign paradigm, e.g., multivariate and code-based signatures [BFM+25]. Moreover, recent work by Baum *et al.* [BBB+26] demonstrates practical instantiations using VOLE-in-the-head techniques and MAYO as an underlying multivariate signature scheme.

We say that a blind-signature scheme is *syntactically compatible* with its underlying signature scheme if its final outputs have the algebraic format of signatures of the underlying scheme and are accepted by its verification algorithm instantiated with the corresponding (possibly enlarged) parameters.

Although the above-mentioned constructions can be practically efficient, particularly in terms of interaction, none of them are syntactically compatible with the underlying signature scheme: they are NIZK proofs verified by the proof system rather than by the verifier of the underlying scheme. As a result, the computational cost and structural complexity of the proof system are inherited by every verifier and propagate to higher-level protocols, leading to increased verification overhead and more complex interfaces. In the post-quantum setting, where both signatures and zero-knowledge proofs are comparatively expensive, this overhead becomes particularly pronounced.

This observation motivates a key design goal: while auxiliary zero-knowledge mechanisms may be unavoidable during the interactive issuance phase, they should be confined to this phase, ensuring that the *final output remains syntactically compatible with the underlying scheme*. This keeps the proof-system-specific machinery out of final verification: the same verification procedure can be used for both blind and ordinary signatures. Importantly, this goal aligns with historical practice: classical blind signature schemes – ranging from Chaum's blind RSA to blind Schnorr and blind ECDSA – were obtained by adapting existing signatures while preserving their format and verification procedures, reflecting a strong preference for compatibility that is likely to persist in the post-quantum setting.

Constructing such schemes. Taking Schnorr signatures as the canonical example, blind signatures have traditionally been obtained by applying blinding techniques to identification schemes, which are then transformed into signature schemes via the Fiat–Shamir (FS) transform. A key drawback of this approach is that the resulting protocols are typically not round-optimal, and therefore concurrent security does not follow automatically. Indeed, it has been shown in the work of Do, Hanzlik and Paracucchi [DHP24] that standard parallel repetitions of Σ-protocols with (polynomially) small challenge spaces cannot yield blind signatures secure against polynomially many concurrent executions. While this limitation does not apply in general to exponentially large challenge spaces, it is well known that Schnorr signatures in particular are vulnerable to ROS attacks [BLL+21], and thus

only achieve weaker (and often impractical) sequential security. As a consequence, existing approaches often resort to designing specialized "blindable" identification or signature schemes that avoid such attacks. Notable examples include recent constructions based on group actions [HLM+25, HLPP26] and lattice-based schemes achieving concurrent security at the cost of additional rounds and heavier machinery [KLR26].

Avenues around these problems. For Schnorr signatures, several alternative approaches have been proposed to restore concurrent security. One direction is the use of clause blind Schnorr signatures [FPS20]. Another line of work, introduced by Fuchsbauer and Wolf [FW24] and later followed by Maire and Pulval-Dady [MPD25], constructs blind signatures directly from the signature assumption. In the work of Fuchsbauer and Wolf [FW24], which constructs a blind signature scheme for Schnorr signatures, the user first commits to its randomness and message before receiving the signer's first message, and later proves in zero knowledge that the challenge was computed correctly. This, however, requires proving knowledge of pre-images under the hash function, while the security of Schnorr signatures themselves relies on modeling the hash as a random oracle, leading to a conceptual tension. To resolve this, they instead assume that Schnorr instantiated with a concrete hash function (e.g., SHA-256) is unforgeable. Although this constitutes a non-standard assumption, it can be viewed as a minimal and practically motivated strengthening, and is implicitly adopted in several recent works on blind signatures [BBB+26], SNARK-friendly signatures [ZSE+24, ADK24, FR25] and anonymous credentials, both in the classical [Fas26, PPZ24, WOS+25] and post-quantum settings [PWFW23, FR25].

Towards a solution. A key advantage of the "blind signatures from a signature assumption" paradigm of Fuchsbauer and Wolf [FW24] is that it simultaneously addresses the challenges identified above: if the security of the blind signature can be reduced to the EUF-CMA security of the underlying signature scheme, then concurrent security is obtained while syntactic compatibility with the underlying scheme is preserved. Unfortunately, adapting this paradigm to the post-quantum setting, and in particular to lattice-based signatures, is far from simple. Existing concurrently secure post-quantum blind signature schemes either produce signatures that are NIZK proofs of valid signatures, include NIZK components in the final signature (e.g., Kloss *et al.* [KLR26]), or require designing entirely new identification and signature schemes.

This leads us to the following research question:

Can we design a blinding technique for lattice-based signature schemes that yields a blind-signature scheme which simultaneously (i) achieves concurrent security, (ii) is syntactically compatible with an underlying signature scheme, and (iii) remains practically efficient?

1.1 Our Contributions

We affirmatively answer the above question. At a high level, we show that the "blind signatures from a signature assumption" paradigm of Fuchsbauer and Wolf [FW24] can be adapted to the lattice setting, despite several fundamental obstacles.

We recall that in the classical setting, Schnorr's algebraic structure enables relatively simple blinding techniques. In contrast, lattice-based signatures introduce significant additional challenges. A first major obstacle is the use of rejection sampling [Lyu09], which ensures that the output distribution of signatures does not leak information about the secret key. In the blind signature setting, however, the final signature is only seen by the user, so the signer cannot verify whether rejection is required. Designing oblivious rejection sampling would in turn risk violating blindness, making this approach problematic.

To circumvent this issue, we build on Raccoon [dPKPR24], a rejection-free lattice-based signature scheme. While its rejection-free design eliminates one major barrier, several obstacles remain. In the original Raccoon proposal, the signing noise is sampled as a sum of uniform distributions. Similarly to Threshold Raccoon [DKM+24], we instead use a discrete-Gaussian variant. The Raccoon

challenge generation is not a simple hash evaluation, but involves an extendable-output function (XOF) and an internal sampling procedure that is not zero-knowledge-friendly. To address this, we adapt the techniques of Policharla *et al.* [PWFW23] to modify the SampleInBall procedure so that rejections become sufficiently rare, enabling efficient zero-knowledge proofs. Our instantiation retains SHA3-256 and SHAKE256 and proves the resulting mixed Boolean and arithmetic relation using ZK-LaBRADOR [BBL+26].

A second key challenge arises from the need to ensure blindness while maintaining syntactic compatibility with Raccoon. In our protocol, the user introduces a blinding shift to randomize the final signature, analogously to blind Schnorr signatures. However, unlike in the Schnorr setting, this shift leads to a non-uniform distribution in the lattice setting, potentially leaking information.

For the security analysis, we follow a strategy analogous to that used in Raccoon [dPKPR24]: we first analyze the Gaussian signature scheme without modulus rounding, and then relate the rounded construction used in the protocol to this non-rounded variant. We denote the latter by $\mathsf{Raccoon}_0$. The deployed blind-signing protocol retains rounding and produces signatures in the rounded Raccoon format, whereas its security analysis proceeds through $\mathsf{Raccoon}_0$. To capture the distributional effect of blinding, we additionally introduce a shifted non-rounded scheme $\mathsf{Raccoon}_0^\star$, whose signing algorithm takes the shift as an explicit input. The blinding shift is sampled from a distribution wide enough to statistically drown the distribution of the signer's initial response. This is what ensures blindness of the final signature, but it also increases the size of the resulting signature.

Concretely, for our 128-bit parameter set supporting up to 2^{32} blind-signing queries, the signature size grows from about 11 KB for the rounded Gaussian Raccoon instantiation to 32 KB for BRaccoon, while the public key has size 5.39 KB. We first reduce the one-more unforgeability of BRaccoon to the unforgeability of $\mathsf{Raccoon}_0^\star$, and then reduce the latter to the unforgeability of $\mathsf{Raccoon}_0$. Appendix D connects ordinary rounded Raccoon to the same non-rounded core. Thus, $\mathsf{Raccoon}_0$ serves as the common security foundation for both Raccoon and BRaccoon.

Finally, we encode the lattice, encryption, SHA3/SHAKE256, and challenge-sampling constraints in mixed R1CS statements proved with ZK-LaBRADOR. The binary and arithmetic components share one witness encoding, so the user and signer each send a single proof. For $Q_s = 2^{32}$, the resulting communication is approximately 862 KB.

Overall, our construction BRaccoon is a concurrently secure blind signature scheme that is syntactically compatible with the corresponding ordinary Raccoon instantiation, demonstrating that this goal can also be achieved for lattice-based Fiat–Shamir signatures.

1.2 Related Work

Table 1 summarizes the state of the art in lattice-based blind signatures, which is the primary focus of this section. For a broader overview of blind signatures from other quantum-safe assumptions, which currently remains limited, we refer to the work of Hanzlik *et al.* [HLM+25, Appendix A].[4]

Fischlin-based constructions. Several lattice-based blind signatures follow Fischlin's framework [Fis06], which yields two-move protocols where the final signature consists of a NIZK proof of knowledge of a valid pre-signature on a commitment to the message. Early lattice-based instantiations include the work of Agrawal *et al.* [AKSY22], whose security relies on the tailored *one-more-ISIS* assumption, and the work of del Pino and Katsumata [dPK22], which instead relies on more standard lattice assumptions but at the cost of significantly larger signatures and transcripts.

[4] The currently known concurrently secure schemes from non-lattice assumptions include [HLM+25], based on group actions, and [BBB+26], based on multivariate quadratic systems.

[5] We refer to [NS26, ALP26] for recent analyses of the hardness of ISIS-f and related SIS-with-hints assumptions.

[6] The figures are those reported in [BMW26, Tables 2 and 3] for the $n = 1296$ parameter set. The scheme additionally relies on NTRU, and its security is not reduced to that of an existing signature scheme, but to two instantiation assumptions on intermediate protocols.

Table 1. Comparison of concurrently secure lattice-based blind signature schemes. The column Q_s denotes the maximum number of signing queries per key supported by the security proof. We defer to Appendix F a brief discussion on how the data in this table was obtained.

Scheme	Non-standard assumptions	vk (KB)	Comm. (KB)	sig (KB)	Sec. (bits)	#Moves	Q_s
Proof of knowledge of signature (Fischlin approach)							
[AKSY22]	OM-ISIS	0.8	1.4	45.2	109	2	2^{64}
[dPK22, HSS24]	–	≈ 56	≈ 1000	≈ 100	128	2	2^{64}
[LNP22a]	–	1331	16384	150	128	2	2^{18}
[BLNS23b, LSS24]	ISIS-f[5]	1	22	27	128	2	2^{64}
[JRS23, AGJ+24]	–	48	45	79.6	128	2	2^{32}
[BLNS23a, BBL+26]	–	1	110	22	128	2	2^{20}
[JS25a, JS25b]	–	36.0	53.2	36.3	125	3	2^{32}
Blinded identification scheme (ad-hoc construction)							
[HKLN20]	–	443.7	34037.3	7915.5	128	3	7
[KLR26]	–	≈ 122750	≈ 9000	≈ 350	128	4	2^{64}
GPV-compatible signature							
[BMW26][6]	$\mathsf{BS}^{\mathsf{UNF}}$, $\mathsf{BS}^{\mathsf{BLIND}}$	4.85	≈ 110	4.70	116	≈ 2.2	2^{64}
Raccoon-compatible signature (our scheme)							
$\mathsf{BRaccoon}_{\mathsf{SHA3}}$	Raccoon	5.39	≈ 794 ≈ 862 ≈ 1066	28.00 31.78 43.10	128	4	2^{20} 2^{32} 2^{64}

Another round-optimal construction in the same "NIZK-in-the-signature" paradigm is the work by Lyubashevsky *et al.* [LNP22a], which requires fixing in advance an upper bound Q_s on the number of signatures per key, resulting in public key size and running time that grow linearly with Q_s. Bootle *et al.* [BLNS23b] also instantiate a round-optimal blind signature via Fischlin's footprint, although the construction is developed within a broader framework for anonymous credentials and related privacy-preserving primitives. Its unforgeability relies on the ISIS-f assumption family, and the corresponding proof system was subsequently implemented in the LaZer library [LSS24]. More recently, Jeudy and Sanders [JS25a] (with refinements in [JS25b]) reduce communication and signature size by reusing commitment entropy, at the cost of tagged public keys and maintaining state to avoid tag reuse. The most efficient scheme in this line is that of Beullens, Lyubashevsky, Nguyen and Seiler [BLNS23a], in which the signer signs a commitment to the message and the user proves knowledge of the resulting signature with the LNP proof system [LNP22b]: signatures are about 22 KB and the transcript is dominated by the user's proof. The size of that proof was originally an analytical estimate; the recent ZK-LaBRADOR work of Biasioli *et al.* [BBL+26], which treats this statement as a case study, reports about 110 KB, roughly twice the original figure, and Table 1 uses the latter value. In contrast to our construction, the final signature is a NIZK proof and its unforgeability is proven for a bounded number $Q_s = 2^{20}$ of signatures per key.

Connections to anonymous credentials and SEPs. In addition to the work of Bootle *et al.* [BLNS23b] discussed above, the STARK-friendly variant of Dilithium by Policharla *et al.* [PWFW23], known as

zkDilithium, provides another connection between lattice-based anonymous credentials and the "NIZK-in-the-signature" paradigm. It replaces the SHAKE-based XOF with the Poseidon hash function and adapts the SampleInBall procedure accordingly. However, existing implementations of SampleInBall [PWFW23] focus on the proof system and do not realize the full unlinkable signing protocol, which is why we omit them from Table 1.

Post-quantum analogues of signatures with efficient protocols (SEPs) [CL03] can also be used as building blocks for blind signatures [JRS23, AGJ+24]. In both anonymous credential systems and SEPs, unlinkable presentation of signatures typically requires either interactive verification or the generation of fresh NIZK proofs. Notably, these constructions, like those of the Fischlin line above, are round-optimal, and thus concurrent security follows automatically.

Schnorr-style constructions. An alternative line of work attempts to follow the template of classical Schnorr blind signatures. The first lattice-based construction in this direction is due to Rückert [Rüc10], with subsequent optimizations in [AEB20, AHJ21]. However, Hauck *et al.* [HKLN20] showed that the original security analyses of these constructions are flawed. They also provide a provably secure variant achieving sequential security, but this comes at the expense of extremely large signatures and communication, as well as a very limited number of queries per public key.

Finally, a recent four-move Fiat–Shamir-based construction aiming at concurrent security is given in [KLR26]. That work is generic and admits instantiations from several assumption families; Table 1 reports its lattice instantiation, which is the one relevant here. It incurs substantially larger communication costs and is based on a tailored blinded identification scheme.

Blind signatures with a GPV output. A recent unpublished work of Blazy, Mallordy and Wen [BMW26] pursues the same goal as ours in the hash-and-sign setting. Their blind signature is a plain GPV signature, namely a short $\boldsymbol{\sigma}$ with $\mathbf{A}\boldsymbol{\sigma} = \mathsf{H}(\mathsf{msg}, \rho)$ for a salt ρ, verified exactly as in Falcon [PFH+22]. The signer holds an NTRU trapdoor, the user embeds the commitment phase of a Σ-protocol into its signing request, and the pre-image returned by the signer is unblinded by bimodal rejection sampling. This yields very short signatures, 4.70 KB, and a transcript of about 110 KB. The price is that the protocol is no longer round-optimal: a session restarts whenever the rejection step fails, and in order to bound the number of restarts a malicious user must additionally prove in zero knowledge that its previous attempt was indeed rejected. The reported expectation is 1.1 rounds for honest users and at most 2.6 for malicious ones.

Their construction is syntactically compatible with the corresponding GPV-type ordinary signature scheme, and it yields substantially shorter signatures than ours. The main difference lies in the endpoint of the security reduction. Their unforgeability is reduced to the one-more unforgeability of an intermediate protocol $\mathsf{BS}^{\mathsf{UNF}}$, which they explicitly note is neither a signature scheme nor a blind signature. As in our setting, this intermediate object is assumed to remain secure once the relevant random oracles are instantiated with concrete hash functions. Our reduction, by contrast, passes through $\mathsf{Raccoon}_0^\star$ and ultimately reduces to the EUF-CMA security of $\mathsf{Raccoon}_0$, which act as a security foundation for both BRaccoon and Raccoon. Their construction also relies on NTRU in addition to MSIS, whereas ours stays with MLWE/MSIS.

Universal blinders from homomorphic encryption. A different route to syntactic compatibility with an existing signature scheme is to evaluate its signing algorithm homomorphically. Boneh and Kim [BK26] give compilers that turn any secure signature scheme into a syntactically compatible one-round blind signature, using two-key fully homomorphic encryption together with NIZK proofs, and introduce a notion of committed verifiable FHE for this purpose. This attains generically, and in a single round, the syntactic compatibility that we pursue for a specific lattice signature. The trade-off is computational: their construction homomorphically evaluates the entire signing circuit, whereas our protocol evaluates only a linear function of the challenge and the signing key, at the cost of four moves and a larger transcript. They do not provide concrete parameters for their construction.

Discussion and comparison. In summary, most existing concrete post-quantum blind-signature constructions are not syntactically compatible with an independently specified ordinary signature scheme: their final output either consists of a NIZK proof of a valid signature or follows the format of a tailored blindable identification scheme. This stands in contrast to classical blind signature schemes, where the output typically retains the format and verification procedure of an underlying ordinary signature scheme.

In this sense, BRaccoon is, to the best of our knowledge, the first concurrently secure lattice-based blind-signature scheme in the Fiat–Shamir family that is syntactically compatible with its underlying signature scheme and whose one-more unforgeability reduces to the unforgeability of that scheme, so that both rely on the same lattice assumptions. Two alternative routes to syntactically compatible outputs were discussed above: the hash-and-sign construction of [BMW26], whose output is a GPV signature but whose security rests on non-standard assumptions on intermediate protocols, and the generic compilers from verifiable fully homomorphic encryption of [BK26], which are round-optimal but not concretely instantiated. The price we pay is a non-optimal number of rounds (4) and a transcript that is larger than in the most compact Fischlin-type scheme: for $Q_s = 2^{20}$, [BLNS23a] needs two moves, about 110 KB of communication and 22 KB signatures, against four moves, about 794 KB and 28 KB here, with a 5.39 KB public key. What we obtain in exchange is a common verification procedure: under the blinding-compatible Raccoon parameter instantiation, both ordinary and blind signatures are accepted by the Raccoon verification algorithm. In contrast, Fischlin-type constructions output a NIZK proof that must be checked by a scheme-specific proof-system verifier rather than by the verifier of the underlying signature scheme. This compatibility is non-vacuous: Raccoon was submitted to NIST's call for additional post-quantum signatures and has an implementation and analysis that are independent of this work.

For clarity, Table 1 omits auxiliary assumptions on hash functions and NIZK proof systems. For instance, [BLNS23a] relies on SHA-2/SHA-3. Our instantiation likewise relies on SHA-3, while the security of the associated NIZK proofs and homomorphic encryption components can be reduced to MLWE assumptions or statistical arguments.

1.3 Technical Overview

Our construction. The main design goal of our blind signature scheme, BRaccoon, is to achieve syntactic compatibility with Raccoon, which is a rejection-free Fiat–Shamir lattice signature scheme producing Dilithium-style signatures without relying on aborts.

At a high level, our protocol preserves the structure of a Raccoon signature while introducing a blinding transformation at the commitment stage; see Figure 1 for an overview. Recall that a Raccoon signature consists of three steps: sampling randomness to form a commitment value $\mathbf{w}$, deriving a challenge c via hashing, and computing a response $\mathbf{z}$ that binds the challenge to the secret key.

In our blind-signing protocol, the user first commits to the message msg together with a blinding factor $\mathbf{r}$ using a commitment scheme (which we can make extractable in our one-more unforgeability proof). The signer then samples fresh randomness $\mathbf{r}'$ and computes the corresponding commitment $\mathbf{w}'$ as in the underlying signature scheme. The user re-randomizes this value using $\mathbf{r}$ to obtain $\mathbf{w}$ and derives the challenge c from it.

To enforce correct challenge computation, the user encrypts c under a linearly homomorphic encryption scheme, obtaining ctx_c, and proves via a NIZK proof π_1 that this encryption is consistent with the committed values. The signer then homomorphically computes an encryption of the response $\mathbf{z}'$ using the signing key $\mathbf{s}$, randomness $\mathbf{r}'$ and ctx_c, yielding $\mathsf{ctx}_{\mathbf{z}'}$, and provides a NIZK proof π_2 that this computation was performed correctly.

Finally, the user decrypts $\mathsf{ctx}_{\mathbf{z}'}$ to obtain $\mathbf{z}'$ and adds the blinding factor $\mathbf{r}$ to derive $\mathbf{z}$. The resulting signature $\mathsf{sig} = (c, \mathbf{z})$ satisfies the Raccoon verification equation (with the enlarged acceptance bound required for blinding) and is therefore syntactically compatible with the corresponding ordinary Raccoon instantiation.

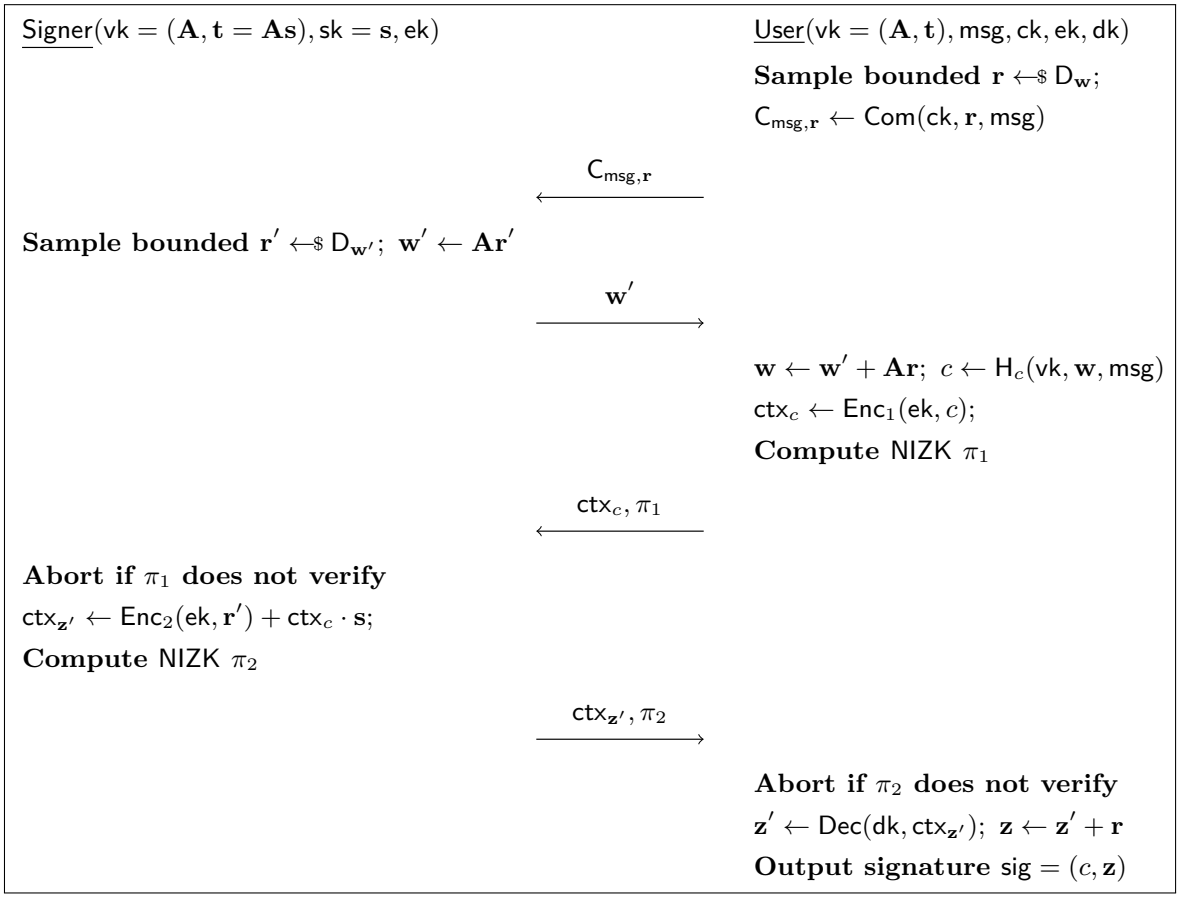


Fig. 1. High-level message flow of the blind signature protocol. Local computations, as well as the explicit computation of the error, rounding, and hint generation, are omitted. Each NIZK proves correctness of the corresponding move.

Security reduction. Proving one-more unforgeability of our construction assuming only the unforgeability of Raccoon naturally suggests simulating the BRaccoon signing oracle using access to a Raccoon signing oracle. However, such a direct simulation fails to perfectly reproduce the distribution of signatures generated by BRaccoon. In particular, the signatures obtained in the reduction are distinguishable from those produced by a genuine BRaccoon execution.

This issue does not arise in the Schnorr-setting of [FW24], where all randomness is sampled uniformly. In that case, adding a blinding shift preserves uniformity and therefore leaves the signature distribution unchanged. In contrast, many lattice-based schemes rely on discrete Gaussian sampling. Introducing an additive blinding shift results in a convolution of distributions, thereby altering the output distribution and preventing a straightforward simulation.

To overcome this obstacle, we introduce a modified signature scheme, denoted $\mathsf{Raccoon}^\star$, that captures the precise distributional effect of the blinding transformation. We reduce one-more unforgeability of the rounded blind protocol to the non-rounded shifted variant $\mathsf{Raccoon}^\star_0$, and then to the non-rounded scheme $\mathsf{Raccoon}_0$. The full OMUF proof is given in Section 4, and Corollary 1 collects the losses of the reductions. Blindness, on the other hand, follows from a standard composition of the following properties: statistical drowning ($\mathbf{r}$ has to be much larger than $\mathbf{r}'$), the hiding of the commitment scheme, the IND-CPA security of the homomorphic encryption scheme, and the soundness and zero-knowledge guarantees of the underlying NIZK proofs.

Bridging lattice and hash constraints. The relation needed to be proven by the user via a NIZK mixes modular lattice equations with SHA3-256, SHAKE256, and SampleInBall. We lift the modular equations to bounded integer equalities and combine them with the Boolean constraints through the mixed-R1CS reduction of LaBRADOR [BS23]. Both parts share one binary witness encoding and are proved using the toolkit of Biasioli *et al.* [BBL+26]. The signer relation is handled by the same toolkit. Thus, π_1 and π_2 are each a single succinct proof.

Instantiation. Finally, we instantiate all building blocks and provide detailed parameter choices, communication complexity estimates, and signature sizes.

2 Background

2.1 Notation

We recall standard notation and definitions, following [MR04] and [LS15, §2].

Rings. Throughout this paper, let $\mathcal{R}$ denote the cyclotomic ring $\mathbb{Z}[X]/(X^n+1)$, and let $\mathcal{R}_q$ denote its reduction modulo q, i.e., $\mathcal{R}_q = \mathcal{R}/q\mathcal{R} = \mathbb{Z}[X]/(q, X^n+1)$.

Norms. Let $z \in \mathcal{R}_q$ be written as $z = \sum_{i=0}^{n-1} a_i X^i$ with $a_i \in (-q/2, q/2]$. We define its ∞-norm and 2-norm as

$$\|z\|_\infty = \max_i |a_i| \qquad \text{and} \qquad \|z\|_2 = \sqrt{\sum_i a_i^2}.$$

For a vector $\mathbf{z} = (z_1, \ldots, z_\ell) \in \mathcal{R}_q^\ell$, we extend these norms component-wise:

$$\|\mathbf{z}\|_\infty = \max_i \|z_i\|_\infty \qquad \text{and} \qquad \|\mathbf{z}\|_2 = \sqrt{\sum_i \|z_i\|_2^2}.$$

Modulus Rounding. Let $\nu \geq 0$, $q > 2^\nu$, and $q_\nu = \lfloor q/2^\nu \rfloor$. For $x \in \mathbb{Z}_q$, let $\overline{x} \in \{0, \ldots, q-1\}$ be its canonical representative. Following [DKM+24, Section 3.1], define modulus rounding by

$$\lfloor x \rceil_\nu := \left\lfloor \frac{\overline{x}}{2^\nu} + \frac{1}{2} \right\rfloor \bmod q_\nu \in \mathbb{Z}_{q_\nu}.$$

The integer rounding is half-up and precedes reduction modulo q_ν. In particular, $\lfloor x \rceil_0 = x$. The operation extends coefficient-wise to ring elements and vectors. The reconstruction error modulo q is bounded as in Lemma 11; after reduction modulo q_ν, it need not lie in the half-width integer rounding interval.

For a rounded hint $\mathbf{h} \in R_{q_\nu}^k$, the scaled hint in a verification norm always means

$$\|(\mathbf{z}, 2^\nu \mathbf{h})\|_2 := \|(\mathbf{z}, 2^\nu \overline{\mathbf{h}} \bmod q)\|_2.$$

Thus the lift of $\mathbf{h}$ is canonical, while the norm is taken using centered representatives modulo q after scaling. This is not the same convention as multiplying the centered lift modulo q_ν by 2^ν over the integers.

Module Lattices. Let R be a ring and let $\mathbf{A} \in R^{k \times \ell}$. The *module lattice generated by* $\mathbf{A}$ is defined as

$$\mathcal{L}(\mathbf{A}) = \{\mathbf{A}\mathbf{z} \mid \mathbf{z} \in R^\ell\} \subseteq R^k.$$

The *kernel module lattice of* $\mathbf{A}$ is defined as

$$\mathcal{L}^\perp(\mathbf{A}) = \{\mathbf{z} \in R^\ell \mid \mathbf{A}\mathbf{z} = \mathbf{0}\} \subseteq R^\ell.$$

In our setting, the relevant rings are $R = \mathcal{R}$ and $R = \mathcal{R}_q$. The ring $\mathcal{R}$ naturally embeds into $\mathbb{Z}^n$ by mapping a polynomial to its vector of coefficients. Consequently, when $R = \mathcal{R}$, $\mathcal{L}(\mathbf{A})$ and $\mathcal{L}^\perp(\mathbf{A})$ can be viewed as integral lattices in $\mathbb{R}^{nk}$ and $\mathbb{R}^{n\ell}$, respectively. Reducing modulo q yields the corresponding lattices over $\mathcal{R}_q$.

Additional Notation. We let λ denote the security parameter. The notation $a \leftarrow\!\$\ \mathcal{X}$ indicates that a is sampled uniformly from a set $\mathcal{X}$. We use PPT to denote probabilistic polynomial-time algorithms. For a cryptographic scheme Scheme consisting of several algorithms, we write $\mathsf{Scheme.A}$ to denote a specific algorithm A within the scheme. Finally, $\overset{s}{\approx}$ denotes statistical indistinguishability.

2.2 Discrete Gaussian Distribution

We recall some standard definitions and properties regarding distributions over lattices [MR04, GPV08, LS15, Pei16]. A *distribution* D over $\mathcal{R}_q$ is a map $\mathsf{D}\colon \mathcal{R}_q \to [0,1]$ such that $\sum_{z\in\mathcal{R}_q} \mathsf{D}(z) = 1$. Distributions over $\mathcal{R}_q$ are typically obtained by reducing modulo q a distribution – specifically, in our case, a Gaussian distribution – over $\mathcal{R}$, viewed as $\mathbb{Z}^n \subseteq \mathbb{R}^n$.

For any $s > 0$ and $\mathbf{c} \in \mathbb{R}^n$ we define the *Gaussian function* on $\mathbb{R}^n$ centered at $\mathbf{c}$ with scaling parameter s as

$$\varrho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x}-\mathbf{c}\|^2/s^2).$$

Given a lattice $\Lambda \subseteq \mathbb{R}^n$, the *discrete Gaussian distribution* over Λ is

$$\mathsf{D}_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{\varrho_{s,\mathbf{c}}(\mathbf{x})}{\varrho_{s,\mathbf{c}}(\Lambda)},$$

where $\varrho_{s,\mathbf{c}}(\Lambda)$ is the *Gaussian mass* of Λ,

$$\varrho_{s,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x}\in\Lambda} \varrho_{s,\mathbf{c}}(\mathbf{x}).$$

When omitted, the subscripts s and $\mathbf{c}$ are taken to be 1 and $\mathbf{0}$ respectively, while Λ is taken to be $\mathcal{R} \cong \mathbb{Z}^n$.

We conclude the section with some classic results that are useful to generate secure parameters for lattice-based schemes. To see how the size of the parameter s affects the tails of the distribution $\mathsf{D}_{\mathbb{Z},s,c}$, following [MR04, §3] we define, for any real $\epsilon > 0$, the *smoothing parameter* of Λ, denoted by $\eta_\epsilon(\Lambda)$, as the smallest s such that $\varrho_{1/s}(\Lambda^* \setminus \{\mathbf{0}\}) \le \epsilon$. Here Λ^* denotes the *dual lattice*

$$\Lambda^* = \{\mathbf{x} \mid \langle \mathbf{x}, \mathbf{y}\rangle \in \mathbb{Z} \ \ \forall \mathbf{y} \in \Lambda\}.$$

The smoothing parameter, intuitively, describes the smoothness of the distribution obtained by sampling random points on Λ and adding a noise with Gaussian distribution of parameter $\eta_\epsilon(\Lambda)$. The smaller ϵ is, the closer this distribution is to the uniform distribution on $\mathbb{R}^n$.

The following result gives an upper bound for $\eta_\epsilon(\Lambda)$, relating it to the successive minima $\lambda_1, \ldots, \lambda_n$ of Λ and Λ^*.

Lemma 1 ([MR04, Lemm. 3.2 and 3.3]).

$$\eta_{1/2^n}(\Lambda) \le \min\left\{ \frac{\sqrt{n}}{\lambda_1(\Lambda^*)}, \sqrt{\frac{\ln(2n(1+2^n))}{\pi}} \cdot \lambda_n(\Lambda) \right\}.$$

For s larger than the smoothing parameter, one has the following tail inequality on $\mathsf{D}_{\mathbb{Z},s}$.

Lemma 2 ([GPV08, Lem. 4.2]). *Let $x \leftarrow \mathsf{D}_{\mathbb{Z},s}$. For any $\epsilon > 0$, any $s \ge \eta_\epsilon(\mathbb{Z})$, and any $t > 0$,*

$$\Pr[|x| \ge t \cdot s] \le 2e^{-\pi t^2} \cdot \frac{1+\epsilon}{1-\epsilon}$$

We also have, for any s, the following tail inequality for 2-norms on $\mathsf{D}_{\mathcal{R},s}$.

Lemma 3 ([PKN+25, Lem. 2.1]). *Let $s > 0$. For $\mathbf{x} \leftarrow\!\$\ \mathsf{D}^k_{\mathcal{R},s}$, we have*

$$\Pr\left[\|\mathbf{x}\|_2 \ge e^{1/4} s\sqrt{nk}\right] \le 2^{-nk/10}.$$

2.3 Lattice Problems

In this subsection we recall some standard lattice problems such as the Module-SIS and Module-LWE, which are formally described in [LS15], and Hint-MLWE, a variant of MLWE described in [KLSS23]. We also introduce the SelfTargetMSIS problem from [DKL+18].

Definition 1 (Module-SIS (MSIS) [LS15]). *Let $\mathbf{A} \leftarrow\!\$\ \mathcal{R}_q^{k\times\ell}$. The Module-SIS problem with parameters $k, \ell > 0$ and $0 < B < q$ consists of finding $\mathbf{z} \in \mathcal{R}_q^{\ell}$ such that $\mathbf{Az} = 0$ over $\mathcal{R}_q$ and $0 < \|\mathbf{z}\|_2 < B$. An adversary $\mathcal{A}$ solves the $\mathsf{MSIS}_{k,\ell,B}$ problem with advantage $\mathsf{Adv}^{\mathsf{MSIS}}_{\mathcal{A}}(\lambda)$ if,*

$$\Pr\left[0 < \|\mathbf{z}\|_2 < B \wedge \mathbf{Az} = \mathbf{0} \;\middle|\; \mathbf{A} \leftarrow \mathcal{R}_q^{k\times\ell} \text{ s.t. } \mathbf{z} \leftarrow \mathcal{A}(\mathbf{A})\right] \geq \mathsf{Adv}^{\mathsf{MSIS}}_{\mathcal{A}}(\lambda).$$

Definition 2 (Module-LWE (MLWE) [LS15]). *The Module-LWE problem with parameters $k, \ell > 0$ and an error distribution D over $\mathcal{R}_q$ asks the adversary $\mathcal{A}$ to distinguish between the following two tuples:*

1. $(\mathbf{A}, \mathbf{As} + \mathbf{e})$
2. $(\mathbf{A}, \mathbf{b})$,

where $\mathbf{A} \in \mathcal{R}_q^{k\times\ell}$, $\mathbf{s} \leftarrow\!\$\ \mathsf{D}^{\ell}$ is a secret vector, $\mathbf{e} \leftarrow\!\$\ \mathsf{D}^{k}$ is an error vector and $\mathbf{b} \leftarrow\!\$\ \mathcal{R}_q^{k}$. $\mathcal{A}$ is said to have advantage $\mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{A}}(\lambda)$ in solving $\mathsf{MLWE}_{k,\ell,\mathsf{D}}$ if,

$$\begin{aligned}\Big|\Pr\left[\mathsf{res} = 1 \;\middle|\; \mathbf{A} \leftarrow \mathcal{R}_q^{k\times\ell}; \mathbf{s} \leftarrow \mathsf{D}^{\ell}; \mathbf{e} \leftarrow \mathsf{D}^{k}; \mathsf{res} \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{As}+\mathbf{e})\right]\\ - \Pr\left[\mathsf{res} = 1 \;\middle|\; \mathbf{A} \leftarrow \mathcal{R}_q^{k\times\ell}; \mathbf{b} \leftarrow \mathcal{R}_q^{k}; \mathsf{res} \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{b})\right]\Big| \geq \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{A}}(\lambda)\end{aligned}$$

Definition 3 (Hint-MLWE (H-MLWE) [KLSS23]). *Let k, ℓ, Q be positive integers, let D_1 and D_2 be probability distributions over $\mathcal{R}_q$, and let $\mathcal{C}$ be a subset of $\mathcal{R}_q$. The Hint-MLWE problem $\mathsf{H\text{-}MLWE}_{q,\ell,k,Q,\mathsf{D}_1,\mathsf{D}_2,\mathcal{C}}$ asks an adversary $\mathcal{A}$ to distinguish between the following two distributions:*

1. $$\left(\mathbf{A},\ \mathbf{As}+\mathbf{e},\ (c_i, \mathbf{z}_i, \mathbf{z}'_i)_{i\in[Q]}\right),$$
2. $$\left(\mathbf{A},\ \mathbf{b},\ (c_i, \mathbf{z}_i, \mathbf{z}'_i)_{i\in[Q]}\right),$$

where $\mathbf{A} \leftarrow \mathcal{R}_q^{k\times\ell}, \mathbf{b} \leftarrow \mathcal{R}_q^{k}, \mathbf{s} \leftarrow \mathsf{D}_1^{\ell}, \mathbf{e} \leftarrow \mathsf{D}_1^{k}$, and, for each $i \in [Q]$, $c_i \leftarrow \mathcal{C}, \mathbf{r}_i \leftarrow \mathsf{D}_2^{\ell}, \mathbf{e}'_i \leftarrow \mathsf{D}_2^{k}$, with

$$\mathbf{z}_i := c_i\mathbf{s} + \mathbf{r}_i, \qquad \mathbf{z}'_i := c_i\mathbf{e} + \mathbf{e}'_i.$$

Definition 4 (SelfTargetMSIS [DKL+18, DKM+24]). *Let k, ℓ be positive integers, let $0 < \eta \ll q$, and let $\mathcal{C} \subseteq \mathcal{R}_q$. Let $\mathsf{H}\colon \mathcal{R}_q^{k} \times \{0,1\}^* \to \mathcal{C}$ be a cryptographically secure hash function modeled as a random oracle. The $\mathsf{SelfTargetMSIS}_{q,\ell,k,\mathcal{C},\eta}$ problem asks an adversary $\mathcal{A}$, given $\mathbf{A} \leftarrow\!\$\ \mathcal{R}_q^{k\times\ell}$, to find $(\mathbf{z}, \mu)$, where*

$$\mathbf{z} = \begin{pmatrix} c \\ \mathbf{z}' \end{pmatrix} \in \mathcal{R}_q^{\ell+k}, \qquad c \in \mathcal{C},$$

such that $0 < \|\mathbf{z}\|_2 \leq \eta$ and $\mathsf{H}([\mathbf{A} \mid I_k]\mathbf{z}, \mu) = c$. The adversary $\mathcal{A}$ is said to have advantage $\mathsf{Adv}^{\mathsf{SelfTargetMSIS}}_{\mathcal{A}}(\lambda)$ in solving $\mathsf{SelfTargetMSIS}_{q,\ell,k,\mathcal{C},\eta}$ if

$$\Pr\left[\begin{array}{c} \mathsf{H}([\mathbf{A} \mid I_k]\mathbf{z}, \mu) = c \ \wedge\ 0 < \|\mathbf{z}\|_2 \leq \eta \\ \mathbf{A} \leftarrow\!\$\ \mathcal{R}_q^{k\times\ell}; \quad (\mathbf{z}, \mu) \leftarrow \mathcal{A}^{\mathsf{H}}(\mathbf{A}); \quad \mathbf{z} = \begin{pmatrix} c \\ \mathbf{z}' \end{pmatrix},\ c \in \mathcal{C} \end{array}\right] \geq \mathsf{Adv}^{\mathsf{SelfTargetMSIS}}_{\mathcal{A}}(\lambda).$$

2.4 Digital Signatures

A digital signature scheme DS is a triple of PPT algorithms $(\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ defined as follows.

- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KGen}(\mathsf{pp})$: Takes the public parameters pp as input and outputs a signing key sk and a verification key vk.
- $\mathsf{sig} \leftarrow \mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, \mathsf{msg})$: Takes the signing key sk and a message $\mathsf{msg} \in \{0,1\}^*$ as input and outputs a signature sig.
- $b \leftarrow \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \mathsf{msg}, \mathsf{sig})$: Takes the verification key vk, a message msg and a signature sig as input and outputs a bit indicating whether sig is deemed valid (1) or invalid (0).

The scheme is *correct* if there exists a negligible function $\mathsf{negl}(\lambda)$ such that, for every message msg,

$$\Pr\left[\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \mathsf{msg}, \mathsf{sig}) = 1 \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda), \\ (\mathsf{sk}, \mathsf{vk}) \leftarrow\!\!\$\ \mathsf{KGen}(\mathsf{pp}), \\ \mathsf{sig} \leftarrow\!\!\$\ \mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, \mathsf{msg}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

Definition 5 (Existential Unforgeability). *A signature scheme* DS *is existentially unforgeable under chosen-message attack, or* EUF-CMA *secure, if for every* PPT *adversary* $\mathcal{A}$ *making at most* $Q_s(\lambda)$ *queries to the signing oracle* $\mathcal{O}^{\mathsf{Sign}}$ *there exists a negligible function* $\mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{DS},\mathcal{A}}(\lambda)$ *such that*

$$\Pr\left[\begin{array}{l} \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \mathsf{msg}^*, \mathsf{sig}^*) = 1 \\ \mathsf{msg}^* \notin \mathcal{Q} \end{array} \;\middle\wedge\; \begin{array}{l} \mathsf{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda), \\ (\mathsf{sk}, \mathsf{vk}) \leftarrow\!\!\$\ \mathsf{KGen}(\mathsf{pp}), \\ (\mathsf{msg}^*, \mathsf{sig}^*) \leftarrow\!\!\$\ \mathcal{A}^{\mathcal{O}^{\mathsf{Sign}}}(\mathsf{pp}, \mathsf{vk}) \end{array}\right] \leq \mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{DS},\mathcal{A}}(\lambda),$$

where the signing oracle, on input a message msg *chosen by the adversary, returns* $\mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, \mathsf{msg})$ *and appends* msg *to the initially empty list* $\mathcal{Q}$ *of queried messages. The adversary may query the oracle adaptively, and* $\mathcal{Q}$ *records only the messages of* completed *queries.*

2.5 Blind Signatures

A blind signature (BS) scheme, introduced in [Cha82] and later formalized in [PS96], can be defined as a 4-tuple of PPT algorithms $(\mathsf{Setup}, \mathsf{KGen}, \Pi, \mathsf{Verify})$, where $\Pi := (\mathsf{User}, \mathsf{Signer})$ is an interactive protocol between a user, who holds a message msg, and a signer, who holds a signing key sk, as defined below:

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$: Takes the security parameter λ as input and outputs public parameters pp.
- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KGen}(\mathsf{pp})$: Takes parameters pp as input and outputs a secret key sk and a verification key vk.
- $\Pi := (\mathsf{User}, \mathsf{Signer})$: This is the interactive protocol between a user and a signer. Each entity has a set of subroutines that we describe below:
  - $(\rho_1, \mathsf{state}_{\mathcal{U}}) \leftarrow \mathsf{User}_1(\mathsf{pp}, \mathsf{vk}, \mathsf{msg})$: The user uses a verification key vk and a message $\mathsf{msg} \in \{0,1\}^*$ to compute a state $\mathsf{state}_{\mathcal{U}}$ and a first message ρ_1. The message ρ_1 is sent to Signer.
  - $(\rho_2, \mathsf{state}_{\mathcal{S}}) \leftarrow \mathsf{Signer}_1(\mathsf{pp}, \mathsf{sk}, \rho_1)$: The signer uses a secret key sk and the first message ρ_1 to compute a state $\mathsf{state}_{\mathcal{S}}$ and a message ρ_2. The message ρ_2 is sent to User.
  - $(\rho_3, \mathsf{state}_{\mathcal{U}}) \leftarrow \mathsf{User}_2(\mathsf{pp}, \mathsf{vk}, \mathsf{state}_{\mathcal{U}}, \rho_2)$: The user uses a verification key vk and the message ρ_2 to compute a state $\mathsf{state}_{\mathcal{U}}$ and a message ρ_3. The message ρ_3 is sent to Signer.
  - $\rho_4 \leftarrow \mathsf{Signer}_2(\mathsf{pp}, \mathsf{sk}, \mathsf{state}_{\mathcal{S}}, \rho_3)$: The signer uses a secret key sk and the message ρ_3 to compute a message ρ_4. The message ρ_4 is sent to User.
  - $\sigma \leftarrow \mathsf{User}_3(\mathsf{pp}, \mathsf{vk}, \mathsf{state}_{\mathcal{U}}, \rho_4)$: The last part of the interaction, where the user uses ρ_4 to derive a signature σ or $\perp$ if the signing protocol fails.

– $\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \mathsf{msg}, \mathsf{sig})$: A verification algorithm that takes as input a verification key vk, a message msg and a signature sig and outputs a bit to indicate whether the signature is deemed valid (1) or invalid (0).

At the end of the protocol, the user obtains a valid signature sig on msg without learning anything about the signing key, while the signer learns nothing about the message. More precisely, we require the following three properties to hold (as in [Cha82, PS00, SU12]),

Definition 6 (Correctness). *A blind signature* $\mathsf{BS} = (\mathsf{Setup}, \mathsf{KGen}, (\mathsf{User}, \mathsf{Signer}), \mathsf{Verify})$ *is correct if there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that for any message* msg*, we have that*

$$\Pr\left[\mathsf{Verify}(\mathsf{pp},\mathsf{vk},\mathsf{msg},\mathsf{sig}) = 1 \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda), \\ (\mathsf{sk},\mathsf{vk}) \leftarrow\!\$\ \mathsf{KGen}(\mathsf{pp}), \\ (\rho_1, \mathsf{state}_{\mathcal{U}}) \leftarrow\!\$\ \mathsf{User}_1(\mathsf{pp},\mathsf{vk},\mathsf{msg}), \\ (\rho_2, \mathsf{state}_{\mathcal{S}}) \leftarrow\!\$\ \mathsf{Signer}_1(\mathsf{pp},\mathsf{sk},\rho_1), \\ (\rho_3, \mathsf{state}_{\mathcal{U}}) \leftarrow\!\$\ \mathsf{User}_2(\mathsf{pp},\mathsf{vk},\mathsf{state}_{\mathcal{U}},\rho_2), \\ \rho_4 \leftarrow\!\$\ \mathsf{Signer}_2(\mathsf{pp},\mathsf{sk},\mathsf{state}_{\mathcal{S}},\rho_3), \\ \mathsf{sig} \leftarrow\!\$\ \mathsf{User}_3(\mathsf{pp},\mathsf{vk},\mathsf{state}_{\mathcal{U}},\rho_4) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

Definition 7 (Blindness). *A blind signature scheme* BS *is blind if for all polynomial-time adversaries* $\mathcal{A}$*, there exists a negligible function* $\mathsf{Adv}^{\mathsf{BLD}}_{\mathsf{BS}}(\lambda)$ *such that the following experiment satisfies:*

$$\Pr\left[b^* = \hat{b} \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda), \\ (\mathsf{vk}, \mathsf{msg}_0, \mathsf{msg}_1, \mathsf{state}_{\mathcal{S}}) \leftarrow\!\$\ \mathcal{A}(\mathsf{pp}), \\ \hat{b} \leftarrow\!\$\ \{0,1\}, \\ \forall j \in \{0,1\}, (\rho_{1,j}, \mathsf{state}_{\mathcal{U},j}) \leftarrow\!\$\ \mathsf{User}_1(\mathsf{pp},\mathsf{vk},\mathsf{msg}_{j\oplus\hat{b}}), \\ (\rho_{2,0}, \rho_{2,1}, \mathsf{state}_{\mathcal{S}}) \leftarrow\!\$\ \mathcal{A}(\mathsf{pp}, \mathsf{state}_{\mathcal{S}}, \rho_{1,0}, \rho_{1,1}), \\ \forall j \in \{0,1\}, (\rho_{3,j}, \mathsf{state}_{\mathcal{U},j}) \leftarrow\!\$\ \mathsf{User}_2(\mathsf{pp},\mathsf{vk},\mathsf{state}_{\mathcal{U},j},\rho_{2,j}), \\ (\rho_{4,0}, \rho_{4,1}, \mathsf{state}_{\mathcal{S}}) \leftarrow\!\$\ \mathcal{A}(\mathsf{pp}, \mathsf{state}_{\mathcal{S}}, \rho_{3,0}, \rho_{3,1}), \\ \forall j \in \{0,1\}, \mathsf{sig}_j \leftarrow\!\$\ \mathsf{User}_3(\mathsf{pp},\mathsf{vk},\mathsf{state}_{\mathcal{U},j},\rho_{4,j}), \\ \textit{if } \perp \in \{\mathsf{sig}_0, \mathsf{sig}_1\} \textit{ then } (\mathsf{sig}_0, \mathsf{sig}_1) \leftarrow (\perp, \perp), \\ b^* \leftarrow\!\$\ \mathcal{A}(\mathsf{pp}, \mathsf{sig}_{\hat{b}}, \mathsf{sig}_{1-\hat{b}}, \mathsf{state}_{\mathcal{S}}) \end{array}\right] \leq \tfrac{1}{2} + \mathsf{Adv}^{\mathsf{BLD}}_{\mathsf{BS}}(\lambda).$$

Here j indexes sessions and $\oplus$ denotes XOR. Session j uses $\mathsf{msg}_{j\oplus\hat{b}}$, while the final pair is returned in message order $(\mathsf{msg}_0, \mathsf{msg}_1)$, not session order. If either session fails, the final pair is $(\perp, \perp)$.

Definition 8 (One-More Unforgeability). *Let $Q_s : \mathbb{N} \to \mathbb{N}$. A blind signature* $(\mathsf{Setup}, \mathsf{KGen}, \Pi := (\mathsf{User}, \mathsf{Signer}), \mathsf{Verify})$ *is one-more unforgeable if for every polynomial-time adversary $\mathcal{A}$ that makes at most $Q_s(\lambda)$* ***concurrent queries*** *there exists a negligible function* $\mathsf{Adv}^{\mathsf{OMUF}}_{\mathsf{BS}}(\lambda)$ *such that,*

$$\Pr\left[\begin{array}{l} \forall\, i \in [\ell], \mathsf{Verify}(\mathsf{pp},\mathsf{vk},m_i,\mathsf{sig}_i) = 1 \;\;\wedge \\ \forall\, 1 \leq i < j \leq \ell, m_i \neq m_j \end{array} \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda), \\ (\mathsf{sk},\mathsf{vk}) \leftarrow \mathsf{KGen}(\mathsf{pp}), \\ \{(m_i, \mathsf{sig}_i)\}_{i\in[\ell]} \leftarrow\!\$\ \mathcal{A}^{\mathsf{Signer}(\mathsf{pp},\mathsf{sk},\cdot)}(\mathsf{pp},\mathsf{vk}) \end{array}\right] \leq \mathsf{Adv}^{\mathsf{OMUF}}_{\mathsf{BS}}(\lambda),$$

where $\ell = Q_s(\lambda)+1$. The adversary plays the role of the user and reaches the honest signer through the oracle $\mathsf{Signer}(\mathsf{pp},\mathsf{sk},\cdot)$*, which consists of two interfaces sharing a table of sessions indexed by identifiers* sid*:*

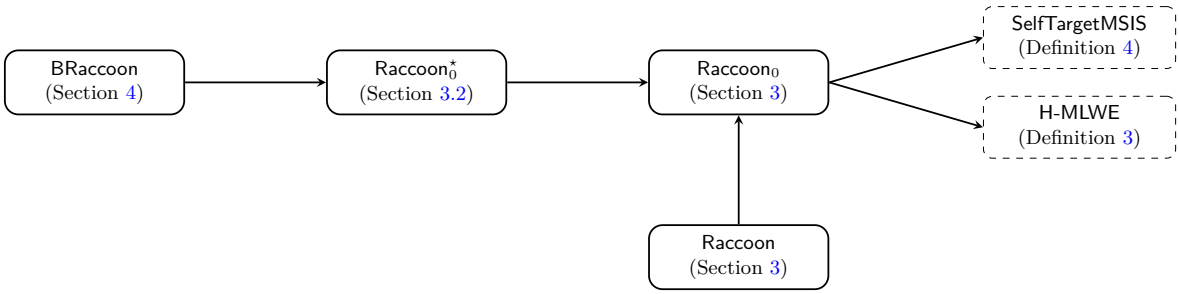


Fig. 2. Reduction structure.

- *On a query* (sid, ρ_1) *with a fresh identifier* sid*, the oracle runs* $(\rho_2, \mathsf{state}_{\mathcal{S}}) \leftarrow\!\$\ \mathsf{Signer}_1(\mathsf{pp}, \mathsf{sk}, \rho_1)$*, stores* $\mathsf{state}_{\mathcal{S}}$ *under* sid*, marks the session* open*, and returns* ρ_2*. It answers at most* $Q_s(\lambda)$ *such queries.*
- *On a query* (sid, ρ_3) *for an open session, the oracle retrieves the stored state* $\mathsf{state}_{\mathcal{S}}$*, runs* $\mathsf{Signer}_2(\mathsf{pp}, \mathsf{sk}, \mathsf{state}_{\mathcal{S}}, \rho_3)$ *to obtain* ρ_4*, marks the session* closed *(whether or not* $\rho_4 = \perp$*), and returns* ρ_4*. Queries for unknown or closed identifiers are answered with* $\perp$*.*

The adversary may open sessions and continue them in any order, in particular interleaving the moves of different sessions, which is what makes the notion concurrent*. The budget* Q_s *counts opened sessions, including sessions the adversary never completes, so the adversary wins only by producing one more valid signature on distinct messages than it opened sessions.*

In BRaccoon, the first message ρ_1 carries an encryption key chosen by the user; the definition therefore lets the adversary choose this key in every session and retain the corresponding decryption key and key-generation randomness.

3 Underlying Signature Schemes

Our blind signature protocol uses modulus rounding to reduce key and signature sizes. For the security analysis, however, we work with non-rounded variants of Raccoon. Following the approach used in the security analysis of Raccoon [dPKPR24], we relate the rounded schemes to their non-rounded counterparts. In particular, both BRaccoon and the standard rounded Raccoon scheme can be reduced to the same non-rounded scheme $\mathsf{Raccoon}_0$, as illustrated in Figure 2. The upper path is the main security reduction for BRaccoon: we first reduce its one-more unforgeability to the EUF-CMA security (Definition 5) of the shifted non-rounded scheme $\mathsf{Raccoon}_0^\star$, which retains the verification equation of $\mathsf{Raccoon}_0$ but modifies its signing procedure to capture the distributional shift induced by the blinding mechanism of our protocol, and then reduce the EUF-CMA security of $\mathsf{Raccoon}_0^\star$ to that of $\mathsf{Raccoon}_0$. The lower path, the reduction from standard rounded Raccoon to $\mathsf{Raccoon}_0$, is deferred to Appendix D. Since $\mathsf{Raccoon}_0$ is the common security foundation of both constructions, we first recall its security and then introduce and analyze its shifted variant $\mathsf{Raccoon}_0^\star$.

3.1 Gaussian Raccoon and its Non-Rounded Variant

Raccoon [dPPRS23, dPKPR24, dEK+23, DKM+24] is a rejection-free lattice-based digital signature scheme whose security is based on the H-MLWE [KLSS23] and MSIS [LS15] assumptions. We consider the variant of del Pino *et al.* [DKM+24], in which the signing noise follows a discrete Gaussian distribution rather than a sum of uniform distributions. We first recall the rounded scheme, which corresponds to the concrete Raccoon construction and fixes the notation used throughout the paper. We then remove modulus rounding to obtain the auxiliary scheme $\mathsf{Raccoon}_0$ on which our security reductions are based.

Table 2. Main parameters and bounds used for Raccoon, $\mathsf{Raccoon}_0$, $\mathsf{Raccoon}_0^\star$, and BRaccoon.

Parameter	Description
$\mathcal{R}_q$	Polynomial ring $\mathbb{Z}_q[X]/(X^n+1)$
ℓ	Secret dimension
k	Public dimension
ω	Challenge weight, with $\|c\|_1 = \omega$
$\nu_{\mathbf{t}}$	Public-key rounding parameter
$\nu_{\mathbf{w}}$	Commitment rounding parameter
$\mathsf{D}_{\mathbf{t}}$	Key-generation distribution for $(\mathbf{s}, \mathbf{e}_{\mathsf{kg}})$
$\mathsf{D}_{\mathbf{w}'}$	Signer randomness distribution for $(\mathbf{r}', \mathbf{e}')$
$\mathsf{D}_{\mathbf{w}}$	User blinding distribution for $(\mathbf{r}, \mathbf{e})$
$\sigma_{\mathbf{t}}$	Gaussian width of $\mathsf{D}_{\mathbf{t}}$
$\sigma_{\mathbf{w}'}$	Gaussian width of $\mathsf{D}_{\mathbf{w}'}$
$\sigma_{\mathbf{w}}$	Gaussian width of $\mathsf{D}_{\mathbf{w}}$
$B_{\mathbf{t}}$	Norm bound on $(\mathbf{s}, \mathbf{e}_{\mathsf{kg}})$
$B_{\mathbf{w}'}$	Norm bound on $(\mathbf{r}', \mathbf{e}')$
$B_{\mathbf{w}}$	Norm bound on $(\mathbf{r}, \mathbf{e})$
$B_2^\star$	Final BRaccoon verification bound on $(\mathbf{z}, 2^{\nu_{\mathbf{w}}}\mathbf{h})$

The norm bounds of Table 2 are the Gaussian tail bounds of Lemma 3 applied to the corresponding distributions. For a pair of vectors of ℓ and k ring elements sampled with width σ, we set $B_{\mathbf{t}} = e^{1/4}\sigma_{\mathbf{t}}\sqrt{n(k+\ell)}$, $B_{\mathbf{w}'} = e^{1/4}\sigma_{\mathbf{w}'}\sqrt{n(k+\ell)}$ and $B_{\mathbf{w}} = e^{1/4}\sigma_{\mathbf{w}}\sqrt{n(k+\ell)}$, so that $\|(\mathbf{s}, \mathbf{e}_{\mathsf{kg}})\|_2 \le B_{\mathbf{t}}$, $\|(\mathbf{r}', \mathbf{e}')\|_2 \le B_{\mathbf{w}'}$ and $\|(\mathbf{r}, \mathbf{e})\|_2 \le B_{\mathbf{w}}$ each hold except with probability at most $2^{-n(k+\ell)/10}$.

Construction Overview. Let $q_{\mathbf{t}} = \lfloor q \rceil_{\nu_{\mathbf{t}}}$ and $q_{\mathbf{w}} = \lfloor q \rceil_{\nu_{\mathbf{w}}}$. The public parameters pp include the security parameter λ, the ring $\mathcal{R}_q$, and two hash functions H and H_c. The first hash compresses the verification key and the message into a fixed-length digest $\mu = H(\mathsf{vk}, \mathsf{msg}) \in \mathcal{S}$, where in our concrete instantiation $\mathcal{S} = \{0,1\}^{2\lambda}$. The challenge hash is modeled as

$$H_c : \mathcal{S} \times \mathcal{R}_{q_{\mathbf{w}}}^k \to \mathcal{C},$$

and maps $(\mu, \mathbf{w})$ to the challenge space $\mathcal{C} = \{c \in \mathcal{R}_q \mid \|c\|_\infty = 1,\ \|c\|_1 = \omega\}$. The scheme, given implicitly generated public parameters pp, is described in Figure 3; it is presented in the same format as its shifted variant in Figure 4 so that the two can be compared line by line.

Removing modulus rounding. For the security analysis, we consider the non-rounded variant of the scheme above, denoted by $\mathsf{Raccoon}_0$, obtained by setting $\nu_{\mathbf{t}} = \nu_{\mathbf{w}} = 0$. In this case, the public key and signing commitment are no longer rounded: $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}_{\mathsf{kg}}$ and $\mathbf{w}' = \mathbf{A}\mathbf{r}' + \mathbf{e}'$. Consequently, the hint simplifies to

$$\mathbf{h} = c\mathbf{e}_{\mathsf{kg}} + \mathbf{e}',$$

and verification reconstructs

$$\mathbf{A}\mathbf{z}' - c\mathbf{t} + \mathbf{h} = \mathbf{A}\mathbf{r}' + \mathbf{e}' = \mathbf{w}'.$$

Thus, $\mathsf{Raccoon}_0$ preserves the algebraic structure of the rounded scheme while removing the rounding terms that are not needed for the core security analysis. We use this non-rounded variant as the basis for the reductions developed below.

Theorem 1 (Correctness of $\mathsf{Raccoon}_0$). *Let* $\mathsf{Raccoon}_0$ *denote the* Raccoon *signature scheme with* $\nu_{\mathbf{t}} = \nu_{\mathbf{w}} = 0$. *Define*

$$B_{2,\mathsf{corr}}^{(0)} := e^{1/4}\big(\omega\sigma_{\mathbf{t}} + \sigma_{\mathbf{w}'}\big)\sqrt{n(k+\ell)}.$$

Then $\mathsf{Raccoon}_0$ *is correct for every verification bound* $B_2^{(0)} \ge B_{2,\mathsf{corr}}^{(0)}$.

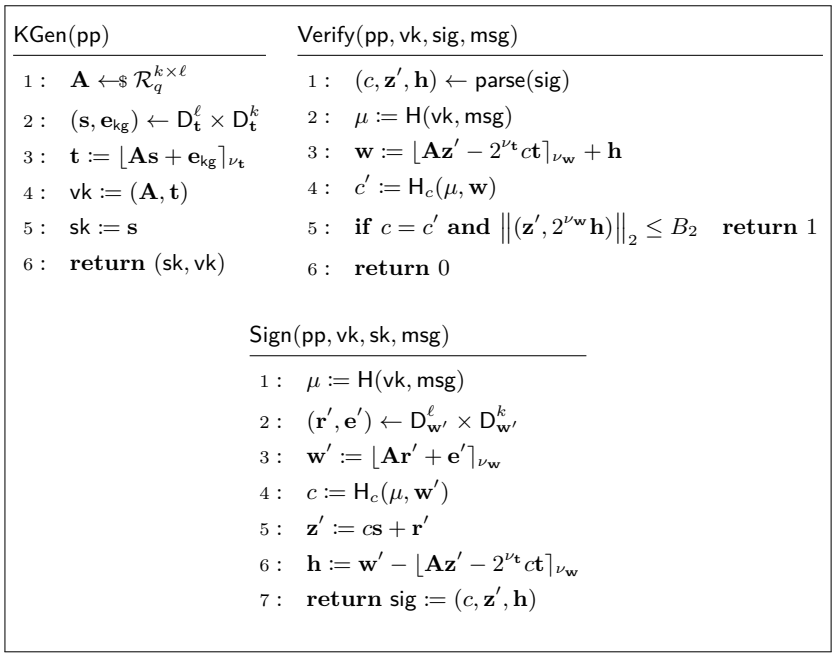


Fig. 3. Gaussian Raccoon with rounding parameters $(\nu_{\mathbf{t}}, \nu_{\mathbf{w}})$. The signing algorithm has no rejection step; the verification bound B_2 holds except with the Gaussian tail probability. Setting $\nu_{\mathbf{t}} = \nu_{\mathbf{w}} = 0$ gives $\mathsf{Raccoon}_0$, and adding the shift $(\mathbf{r}, \mathbf{e})$ to $(\mathbf{r}', \mathbf{e}')$ gives $\mathsf{Raccoon}_0^{\star}$ (Figure 4).

Theorem 2 (Unforgeability of $\mathsf{Raccoon}_0$). *Let $\mathcal{A}$ make at most Q_s signing queries and Q_H queries to the random oracle H_c, let $B_2^{(0)} \ge B_{2,\mathsf{corr}}^{(0)}$, define $B_{\mathsf{MSIS}}^{(0)} := B_2^{(0)} + \sqrt{\omega}$, and assume $\sigma_{\mathbf{w}'} > 2n\, q^{\frac{1}{k+\ell}+\frac{2}{n\ell}}$. Then there exist adversaries $\mathcal{B}$ and $\mathcal{C}$, running in about the time of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{Raccoon}_0,\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathsf{H\text{-}MLWE}_{q,\ell,k,Q_s,\sigma_{\mathbf{t}},\sigma_{\mathbf{w}'},\mathcal{C}}}_{\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{SelfTargetMSIS}_{q,\ell+1,k,\mathcal{C},B^{(0)}_{\mathsf{MSIS}}}}_{\mathcal{C}}(\lambda) + \left(Q_s Q_H + \frac{Q_s(Q_s-1)}{2}\right) 2^{-(n-1)} + 2^{-n+1} + \varepsilon_{\mathsf{hash}},$$

where $\varepsilon_{\mathsf{hash}}$ bounds the probability of a collision of $\mathsf{msg} \mapsto \mathsf{H}(\mathsf{vk}, \mathsf{msg})$ between the forged message and a signed one. The two Gaussian-entropy terms account for programming H_c at the signing commitments, and the condition on $\sigma_{\mathbf{w}'}$ is the hypothesis of the min-entropy lemma of Appendix B. Both theorems are proven in Appendix C.1.

Theorem 2 holds for any verification bound $B_2^{(0)} \ge B_{2,\mathsf{corr}}^{(0)}$, so a later reduction may enlarge $B_2^{(0)}$ to absorb a distributional shift or the removal of modulus rounding; this changes the norm parameter of the final SelfTargetMSIS instance through $B_{\mathsf{MSIS}}^{(0)} = B_2^{(0)} + \sqrt{\omega}$.

3.2 $\mathsf{Raccoon}_0^{\star}$: Non-Rounded Shifted Raccoon

We now introduce $\mathsf{Raccoon}_0^{\star}$, a shifted variant of $\mathsf{Raccoon}_0$ that captures the distribution of signatures arising in our blind-signature protocol. Recall that a signature of $\mathsf{Raccoon}_0$ is generated using $(\mathbf{r}', \mathbf{e}')$. In the blind-signature protocol, the user additionally contributes an independent blinding shift $(\mathbf{r}, \mathbf{e})$, so that the resulting signature effectively depends on $(\mathbf{r}' + \mathbf{r}, \mathbf{e}' + \mathbf{e})$ rather than on $(\mathbf{r}', \mathbf{e}')$ alone. The scheme $\mathsf{Raccoon}_0^{\star}$ makes this shift explicit. Before signing, an auxiliary pair $(\mathbf{r}, \mathbf{e})$ is sampled from the blinding distribution $\mathsf{D}_{\mathbf{w}}^{\ell} \times \mathsf{D}_{\mathbf{w}}^{k}$ and provided to the signing algorithm. The signer then generates an otherwise standard $\mathsf{Raccoon}_0$ signature with this shift added to its signing randomness. In particular,

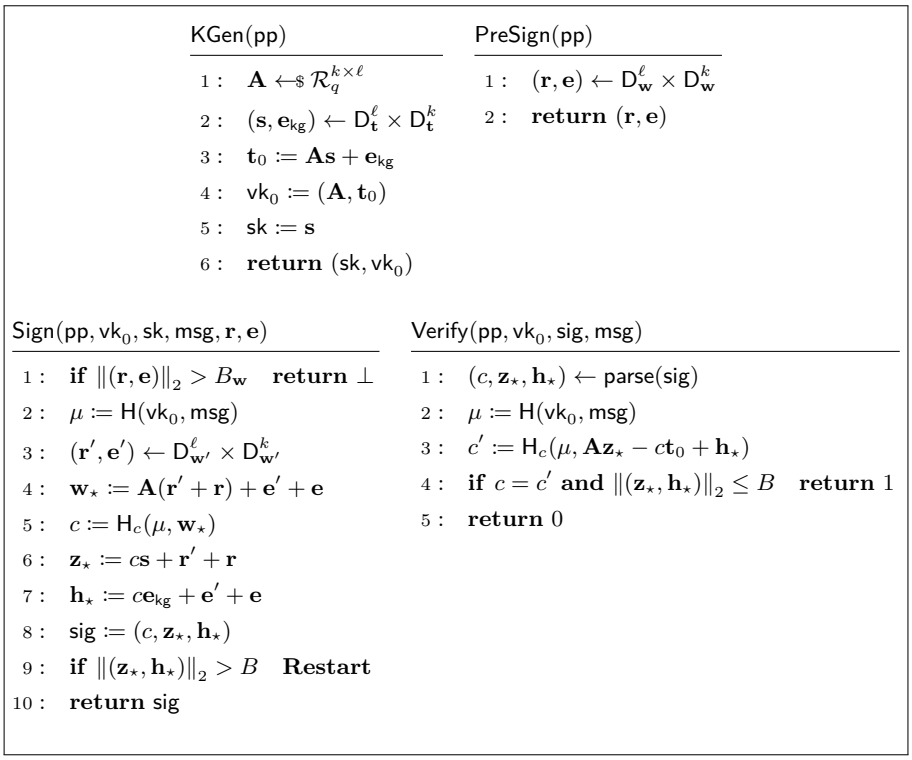


Fig. 4. Protocol description of $\mathsf{Raccoon}_0^\star$. The scheme is defined relative to implicitly generated public parameters pp. The auxiliary shift $(\mathbf{r}, \mathbf{e}) \leftarrow \mathsf{D}_{\mathbf{w}}^{\ell} \times \mathsf{D}_{\mathbf{w}}^{k}$ models the blinding introduced by our blind-signature protocol. Correctness follows from $\mathbf{A}\mathbf{z}_\star - c\mathbf{t}_0 + \mathbf{h}_\star = \mathbf{A}(\mathbf{r}' + \mathbf{r}) + \mathbf{e}' + \mathbf{e} = \mathbf{w}_\star \pmod q$. The distribution $\mathsf{D}_{\mathbf{w}}$ is chosen sufficiently wide to statistically drown the contribution of $(\mathbf{r}', \mathbf{e}')$, while the verification bound satisfies $B \ge B_{2,\mathsf{corr}}^{(0)} + B_{\mathbf{w}}$.

the verification equation remains unchanged, while the distribution of the resulting signature matches the shifted distribution needed to model the blind-signing protocol.

Introducing this intermediate scheme allows us to separate two issues in the security proof: first, relating the blind-signature protocol to the shifted signature distribution captured by $\mathsf{Raccoon}_0^\star$, and second, relating the unforgeability of $\mathsf{Raccoon}_0^\star$ back to that of the ordinary non-rounded scheme $\mathsf{Raccoon}_0$; it acts as a bridge between the two schemes. Figure 4 gives the formal description. An ordinary signature scheme of a similar flavour, namely a Lyubashevsky-style signature that dispenses with rejection sampling and relies on Gaussian noise flooding, was given earlier by Agrawal, Stehlé and Yadav [ASY22] and could serve as an alternative underlying scheme here. We build on Raccoon because our reduction relies on the H-MLWE analysis of the underlying scheme; note that, while Raccoon and [ASY22] use uniform noise and a Rényi-divergence argument respectively, our variant uses Gaussian noise and the blinding shift is drowned statistically.

Theorem 3 (Correctness of $\mathsf{Raccoon}_0^\star$). *The scheme $\mathsf{Raccoon}_0^\star$ described in Figure 4 is correct for every verification bound satisfying*

$$B \ge B_{2,\mathsf{corr}}^{(0)} + B_{\mathbf{w}}. \tag{1}$$

Proof. As shown in Figure 4, $(\mathbf{z}_\star, \mathbf{h}_\star) = (\mathbf{z}_0, \mathbf{h}_0) + (\mathbf{r}, \mathbf{e})$. Hence Theorem 1, the shift bound $\|(\mathbf{r}, \mathbf{e})\|_2 \le B_{\mathbf{w}}$, and the triangle inequality give

$$\|(\mathbf{z}_\star, \mathbf{h}_\star)\|_2 \le B_{2,\mathsf{corr}}^{(0)} + B_{\mathbf{w}} \le B.$$

The verification equation follows directly from the equality displayed in the caption of Figure 4.

Theorem 4 (Unforgeability of $\mathsf{Raccoon}_0^\star$). *Let $\mathcal{A}$ make at most Q_s signing queries and Q_H challenge-oracle queries against $\mathsf{Raccoon}_0^\star$ with verification bound $B \geq B_{2,\mathsf{corr}}^{(0)} + B_{\mathbf{w}}$, and let $\mathsf{Raccoon}_0$ use the same bound $B_2^{(0)} := B$. Assume that, before each fresh signing sample and conditioned on the preceding history and the chosen shift, the source commitment $\mathbf{w}_0 = \mathbf{A}\mathbf{r}' + \mathbf{e}'$ has min-entropy at least κ_w, except for an exceptional history event of probability at most $\varepsilon_{\mathsf{ent}}$ per query. Let $\varepsilon_{\mathsf{corr}}$ bound the probability that, over the at most Q_s signing queries, some first signing sample violates $\|(\mathbf{z}_\star, \mathbf{h}_\star)\|_2 \leq B$, so that the signing algorithm of Figure 4 would restart. Then there exists a reduction $\mathcal{B}$ using at most Q_s ordinary signing queries such that*

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{Raccoon}_0^\star,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{Raccoon}_0,\mathcal{B}}(\lambda) + \varepsilon_{\mathsf{sim}},$$
$$\varepsilon_{\mathsf{sim}} := 2Q_s(Q_H + 2Q_s)2^{-\kappa_w} + Q_s\varepsilon_{\mathsf{ent}} + \varepsilon_{\mathsf{corr}}.$$

The proof is given in Appendix C.2.

4 Concurrently Secure Blind Signatures from Raccoon

We now present $\mathsf{BRaccoon}$, our blind-signature protocol. The protocol consists of four moves and outputs signatures with the same syntax and verification procedure as rounded $\mathsf{Raccoon}$, with parameters enlarged to account for the user blinding. Its one-more unforgeability analysis proceeds through the non-rounded auxiliary schemes $\mathsf{Raccoon}_0^\star$ and $\mathsf{Raccoon}_0$ introduced in Section 3.

4.1 Protocol Overview

Our construction follows the "blind signatures from a signature assumption" paradigm of Fuchsbauer–Wolf [FW24], together with techniques appearing in subsequent threshold lattice-signature constructions [GKS24, GHK+25]. The protocol combines commitments (CS), linearly homomorphic encryption (HE), and NIZK proofs in order to preserve the structure of a Raccoon signature while hiding the message and the resulting signature from the signer.

At a high level, the user first commits to the message msg together with a blinding shift $(\mathbf{r}, \mathbf{e})$. The signer then samples fresh Raccoon signing randomness $(\mathbf{r}', \mathbf{e}')$ and sends the corresponding unrounded value

$$\mathbf{w}' = \mathbf{A}\mathbf{r}' + \mathbf{e}'.$$

The user adds its blinding shift and only then applies modulus rounding,

$$\mathbf{w} = \lfloor \mathbf{w}' + \mathbf{A}\mathbf{r} + \mathbf{e} \rceil_{\nu_{\mathbf{w}}},$$

from which it derives the challenge c. The challenge is encrypted under HE, yielding ctx_c, and the user proves in zero knowledge, through the language L_1, that the commitment, the blinded value $\mathbf{w}$, the challenge c, and its encryption are mutually consistent.

The signer then evaluates the response computation in the encrypted domain. Using its signing key $\mathbf{s}$ and fresh randomness $\mathbf{r}'$, it obtains an encryption of

$$\mathbf{z}' = c\mathbf{s} + \mathbf{r}'$$

and proves through L_2 that this computation was performed correctly. The user decrypts $\mathbf{z}'$, adds its blinding term $\mathbf{r}$, and computes the Raccoon hint $\mathbf{h}$. The final output

$$\mathsf{sig} = (c, \mathbf{z}, \mathbf{h})$$

therefore has exactly the syntax expected by the rounded Raccoon verification algorithm. The two proof languages play complementary roles. The user proof π_1 prevents a malicious user from choosing a challenge inconsistently with its previously committed message and blinding shift, which is required

for the one-more unforgeability reduction. Conversely, the signer proof π_2 ensures that the encrypted response is generated according to the prescribed signing computation, which is required for blindness. Figures 5 and 6 give the formal algorithms, following the blind-signature syntax recalled in Section 2.5.

Here HE is a *linearly* homomorphic public-key encryption scheme, in the sense of Section A.2: its admissible function family consists of the affine maps $f_a(b, c) = a \cdot b + c$ whose multiplier a has bounded ℓ_1 norm, which is all our protocol ever evaluates. We instantiate it with the MLWE-based BGV variant, and we never require full homomorphism.

We write pp for the public parameters of the scheme. They collect everything the auxiliary primitives need in order to run: their respective rings and the security parameter, the hash functions H and H_c, the commitment key ck, the public matrices of the two NIZK proof systems, and the public data of HE. These parameters are generated transparently from public seeds; the corresponding seed expansion is modeled as a random oracle in the security proof, and the concrete generation is deferred to the instantiation section. Following the convention of Section 2.1, every algorithm below takes pp implicitly, and we write it explicitly only where it clarifies the statement.

Language descriptions. We now formalize the consistency conditions enforced by the two NIZK proofs. The language L_1 certifies the user's challenge-generation step, whereas L_2 certifies the signer's encrypted response computation. For readability, we suppress explicit domain conditions when they are clear from context; the complete bounded relations, including commitment-opening and encryption-randomness bounds, are given in Appendix G.

$$\mathsf{L}_1 := \left\{ \begin{array}{c} (C_{\mathsf{msg},\mathbf{r},\mathbf{e}}, \mathsf{ctx}_c, \mathbf{w}', \\ \mathsf{vk} = (\mathbf{A}, \mathbf{t}), \mathsf{ek}, \mathsf{pp}) \end{array} \middle| \begin{array}{l} \exists\, (\mathbf{r}, \mathbf{e}, \mathbf{w}, \mathsf{msg}, \mu, c) \text{ s.t.} \\ \mathsf{CS.Verify}(\mathsf{ck}, (\mathsf{msg}, \mathbf{r}, \mathbf{e}), C_{\mathsf{msg},\mathbf{r},\mathbf{e}}) = 1, \\ \mathbf{w} = \lfloor \mathbf{w}' + \mathbf{A}\mathbf{r} + \mathbf{e} \rceil_{\nu_{\mathbf{w}}}, \\ \mu = \mathsf{H}(\mathsf{vk}, \mathsf{msg}), \\ c = \mathsf{H}_c(\mu, \mathbf{w}), \\ \mathsf{ctx}_c = \mathsf{HE.Enc}(\mathsf{pp}, \mathsf{ek}, c), \\ \|(\mathbf{r}, \mathbf{e})\|_2 \le B_{\mathbf{w}} \end{array} \right\}.$$

The language L_1 binds the encrypted challenge to the message and blinding shift fixed in the user's first-round commitment. In particular, once $(\mathsf{msg}, \mathbf{r}, \mathbf{e})$ is extracted from this commitment in the OMUF reduction, soundness of π_1 guarantees that the accepted ciphertext ctx_c encrypts the unique challenge obtained from the corresponding rounded commitment $\mathbf{w}$.

$$\mathsf{L}_2 := \left\{ (\mathsf{ctx}_{\mathbf{z}'}, \mathsf{ctx}_c, \mathbf{w}',\ \mathsf{vk} = (\mathbf{A}, \mathbf{t}), \mathsf{ek}, B_{\mathbf{w}'}, \mathsf{pp}) \middle| \begin{array}{l} \exists\, (\mathbf{r}', \mathbf{e}', \mathbf{s}, \mathbf{e}_{\mathsf{kg}}, \mathsf{ctx}_{\mathbf{r}'}) \text{ s.t.} \\ \mathbf{w}' = \mathbf{A}\mathbf{r}' + \mathbf{e}', \\ \mathbf{t} = \lfloor \mathbf{A}\mathbf{s} + \mathbf{e}_{\mathsf{kg}} \rceil_{\nu_{\mathbf{t}}}, \\ \mathsf{ctx}_{\mathbf{r}'} = \mathsf{HE.Enc}(\mathsf{pp}, \mathsf{ek}, \mathbf{r}'; \mathsf{D}_{\mathsf{fl}}), \\ \mathsf{ctx}_{\mathbf{z}'} = \mathsf{HE.Eval}(\mathsf{pp}, \mathsf{ek}, f_{\mathbf{s}}, (\mathsf{ctx}_c, \mathsf{ctx}_{\mathbf{r}'})), \\ \|(\mathbf{r}', \mathbf{e}')\|_2 \le B_{\mathbf{w}'} \end{array} \right\}.$$

The language L_2 certifies that the signer formed its response using a valid signing key and fresh bounded signing randomness, and that the returned ciphertext is the prescribed private affine evaluation on ctx_c. In the blindness proof, this ensures that even a malicious signer cannot encode additional information in its response beyond what is allowed by the prescribed signing computation.

4.2 Security of the Protocol

Concurrent security refers to one-more unforgeability: the adversary of Definition 8 may open up to Q_s sessions and interleave their moves arbitrarily, and our reduction does not rewind it. Blindness (Definition 7) is the standard two-session notion.

$\mathsf{KGen}(\mathsf{pp})$

1 : $\mathbf{A} \leftarrow\$ \mathcal{R}_q^{k\times\ell}$
2 : $(\mathbf{s}, \mathbf{e}_{\mathsf{kg}}) \leftarrow \mathsf{D}_{\mathbf{t}}^{\ell} \times \mathsf{D}_{\mathbf{t}}^{k}$
3 : $\mathbf{t} \coloneqq \lfloor \mathbf{A}\mathbf{s} + \mathbf{e}_{\mathsf{kg}} \rceil_{\nu_{\mathbf{t}}}$
4 : $\mathsf{vk} \coloneqq (\mathbf{A}, \mathbf{t})$
5 : $\mathsf{sk} \coloneqq \mathbf{s}$
6 : **return** $(\mathsf{sk}, \mathsf{vk})$

$\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \mathsf{sig}, \mathsf{msg})$

1 : $(c, \mathbf{z}, \mathbf{h}) \coloneqq \mathsf{parse}(\mathsf{sig})$
2 : $\mu \coloneqq \mathsf{H}(\mathsf{vk}, \mathsf{msg})$
3 : $c' \coloneqq \mathsf{H}_c(\mu, \lfloor \mathbf{A}\mathbf{z} - 2^{\nu_{\mathbf{t}}} c\mathbf{t} \rceil_{\nu_{\mathbf{w}}} + \mathbf{h})$
4 : **if** $c = c'$ **and** $\|(\mathbf{z}, 2^{\nu_{\mathbf{w}}}\mathbf{h})\|_2 \le B_2^{\star}$ **then**
5 : **return** 1
6 : **return** 0

Fig. 5. KeyGen and Verify algorithms of BRaccoon.

$\mathsf{User}_1(\mathsf{pp}, \mathsf{ck}, \mathsf{vk}, \mathsf{msg}, \mathsf{ek}, \mathsf{dk})$

1 : $(\mathbf{r}, \mathbf{e}) \leftarrow \mathsf{D}_{\mathbf{w}}^{\ell} \times \mathsf{D}_{\mathbf{w}}^{k}$
2 : $C_{\mathsf{msg},\mathbf{r},\mathbf{e}} \coloneqq \mathsf{CS.Com}(\mathsf{ck}, (\mathbf{r}, \mathbf{e}, \mathsf{msg}))$
3 : $\mathsf{state}_{\mathsf{U}} \coloneqq (\mathbf{r}, \mathbf{e}, \mathsf{msg}, \mathsf{ck}, \mathsf{vk}, \mathsf{ek}, \mathsf{dk})$
4 : **return** $(C_{\mathsf{msg},\mathbf{r},\mathbf{e}}, \mathsf{state}_{\mathsf{U}})$

$\mathsf{Signer}_1(\mathsf{pp}, \mathsf{ek}, \mathsf{sk}, \mathsf{vk}, C)$

1 : $(\mathbf{r}', \mathbf{e}') \leftarrow \mathsf{D}_{\mathbf{w}'}^{\ell} \times \mathsf{D}_{\mathbf{w}'}^{k}$
2 : $\mathbf{w}' \coloneqq \mathbf{A}\mathbf{r}' + \mathbf{e}'$
3 : $\mathsf{state}_{\mathsf{S}} \coloneqq (C_{\mathsf{msg},\mathbf{r},\mathbf{e}}, \mathsf{ek}, \mathsf{sk}, \mathsf{vk}, \mathbf{r}', \mathbf{e}')$
4 : **return** $(\mathbf{w}', \mathsf{state}_{\mathsf{S}})$

$\mathsf{User}_2(\mathsf{pp}, \mathsf{state}_{\mathsf{U}}, \mathbf{w}')$

1 : $\mathbf{w} \coloneqq \lfloor \mathbf{w}' + \mathbf{A}\mathbf{r} + \mathbf{e} \rceil_{\nu_{\mathbf{w}}}$
2 : $\mu \coloneqq \mathsf{H}(\mathsf{vk}, \mathsf{msg})$
3 : $c \coloneqq \mathsf{H}_c(\mu, \mathbf{w})$
4 : $\mathsf{ctx}_c \coloneqq \mathsf{HE.Enc}(\mathsf{pp}, \mathsf{ek}, c)$
5 : $\pi_1 \coloneqq \mathsf{NIZK}_{\mathsf{L}_1}.\mathsf{Prove}(\mathsf{pp}, \mathsf{msg}, \mathbf{r}, \mathbf{e}, \mathbf{w}, \mu)$
6 : $\mathsf{state}_{\mathsf{U}} \coloneqq \mathsf{state}_{\mathsf{U}} \| (\mathsf{ctx}_c, c, \mathbf{w}', \mathbf{w})$
7 : **return** $(\mathsf{state}_{\mathsf{U}}, \pi_1, \mathsf{ctx}_c)$

$\mathsf{Signer}_2(\mathsf{pp}, \mathsf{state}_{\mathsf{S}}, \pi_1, \mathsf{ctx}_c)$

1 : **if** $\mathsf{NIZK}_{\mathsf{L}_1}.\mathsf{Verify}(\mathsf{pp}, \pi_1) = 0$ **then**
2 : **abort**
3 : $\mathsf{ctx}_{\mathbf{r}'} \coloneqq \mathsf{HE.Enc}(\mathsf{pp}, \mathsf{ek}, \mathbf{r}'; \mathsf{D}_{\mathsf{fl}})$
4 : $\mathsf{ctx}_{\mathbf{z}'} \coloneqq \mathsf{HE.Eval}(\mathsf{pp}, \mathsf{ek}, f_{\mathbf{s}}, (\mathsf{ctx}_c, \mathsf{ctx}_{\mathbf{r}'}))$
5 : $\pi_2 \coloneqq \mathsf{NIZK}_{\mathsf{L}_2}.\mathsf{Prove}(\mathsf{pp}, \mathbf{r}', \mathbf{e}', \mathbf{s}, \mathbf{e}_{\mathsf{kg}})$
6 : **return** $(\pi_2, \mathsf{ctx}_{\mathbf{z}'})$

$\mathsf{User}_3(\mathsf{pp}, \mathsf{state}_{\mathsf{U}}, \pi_2, \mathsf{ctx}_{\mathbf{z}'})$

1 : **if** $\mathsf{NIZK}_{\mathsf{L}_2}.\mathsf{Verify}(\mathsf{pp}, \pi_2) = 0$ **then**
2 : **abort**
3 : $\mathbf{z}' \coloneqq \mathsf{HE.Dec}(\mathsf{pp}, \mathsf{dk}, \mathsf{ctx}_{\mathbf{z}'})$
4 : $\mathbf{z} \coloneqq \mathbf{z}' + \mathbf{r}$
5 : $\mathbf{y} \coloneqq \lfloor \mathbf{A}\mathbf{z} - 2^{\nu_{\mathbf{t}}} c\mathbf{t} \rceil_{\nu_{\mathbf{w}}}$
6 : $\mathbf{h} \coloneqq \mathbf{w} - \mathbf{y}$
7 : **return** $\mathsf{sig} = (c, \mathbf{z}, \mathbf{h})$

Fig. 6. User and Signer subroutines for BRaccoon. The subroutine Signer_1 receives the first-round commitment C but does not act on it: the commitment is bound to the session in the next move, when Signer_2 verifies π_1, whose statement contains C and whose language L_1 ties the encrypted challenge to the message and blinding shift committed in it.

Theorem 5 (Correctness). *For any $\nu_{\mathbf{t}}, \nu_{\mathbf{w}} \ge 0$, let*

$$B_2^{\star} = e^{\frac{1}{4}}(\omega\sigma_{\mathbf{t}} + \sigma_{\mathbf{w}'} + \sigma_{\mathbf{w}})\sqrt{n(k+\ell)} + (\omega(2^{\nu_{\mathbf{t}}} - 1) + 2^{\nu_{\mathbf{w}}} - 1 + f(\nu_{\mathbf{w}}))\sqrt{nk},$$

where ω is the 1-norm of c and f is defined in Lemma 11. Let $\mathsf{sig} = (c, \mathbf{z}, \mathbf{h})$ be a blind signature generated by honest participants using our protocol BRaccoon. Then sig is valid with overwhelming probability. The proof is given in Appendix E.1.

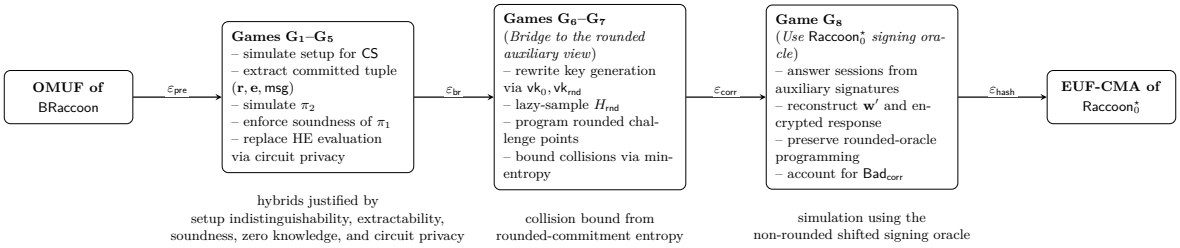


Fig. 7. Proof structure of the reduction from one-more unforgeability of BRaccoon to EUF-CMA security of $\mathsf{Raccoon}_0^\star$. Each arrow denotes a game transition or reduction step, and the quantity displayed above it bounds the corresponding loss in the adversary's success probability.

One-More Unforgeability. We reduce one-more unforgeability of the rounded protocol to the EUF-CMA security of the non-rounded shifted scheme $\mathsf{Raccoon}_0^\star$ from Section 3. The reduction extracts the message and blinding shift chosen by the adversarial user and then simulates each signing session using a $\mathsf{Raccoon}_0^\star$ signing query. The main technical issue is to reconcile the rounded challenge used by BRaccoon with the non-rounded challenge oracle of $\mathsf{Raccoon}_0^\star$. Figure 7 summarizes the game sequence and the corresponding reduction losses.

The proof relies on two additional ingredients. First, the shifted Raccoon commitment must retain sufficient entropy so that rounded challenge points can be programmed with only negligible collision probability. Second, an accepted rounded BRaccoon signature must be convertible into a valid non-rounded $\mathsf{Raccoon}_0^\star$ signature with a controlled increase in the verification bound. We establish these facts before giving the game sequence. For a fresh signing sample $(\mathbf{r}', \mathbf{e}')$ and an admissible shift $(\mathbf{r}, \mathbf{e})$ fixed beforehand, write

$$\mathbf{w}_{\mathsf{raw}} := \mathbf{A}(\mathbf{r}' + \mathbf{r}) + \mathbf{e}' + \mathbf{e}, \qquad \mathbf{w} := \lfloor \mathbf{w}_{\mathsf{raw}} \rceil_{\nu_{\mathbf{w}}}.$$

The first quantity is the commitment seen by the non-rounded shifted scheme, whereas the second is the rounded commitment from which BRaccoon derives its challenge.

Lemma 4 (Rounded-to-non-rounded verification). *Let* $\mathsf{vk}_0 = (\mathbf{A}, \mathbf{t}_0)$ *and* $\mathsf{vk}_{\mathsf{rnd}} = (\mathbf{A}, \mathbf{t})$*, where* $\mathbf{t} = \lfloor \mathbf{t}_0 \rceil_{\nu_{\mathbf{t}}}$*. Set* $q_{\mathbf{w}} := \lfloor q/2^{\nu_{\mathbf{w}}} \rfloor$ *and* $q_{\perp,\mathbf{w}} := q - 2^{\nu_{\mathbf{w}}} q_{\mathbf{w}}$*. Let* $\iota : \mathcal{R}_{q_{\mathbf{w}}}^k \to \mathcal{R}_q^k$ *denote the canonical injection* $\iota(\mathbf{w}) := 2^{\nu_{\mathbf{w}}} \overline{\mathbf{w}} \bmod q$*. Define*

$$\Delta_{\mathsf{rnd}} := (\omega(2^{\nu_{\mathbf{t}}} - 1) + (2^{\nu_{\mathbf{w}}} - 1) + q_{\perp,\mathbf{w}}) \sqrt{nk}.$$

Let $(c, \mathbf{z}, \mathbf{h})$ *be accepted under* $\mathsf{vk}_{\mathsf{rnd}}$ *with bound* $B_2^\star$*, and let*

$$\mathbf{w} := \lfloor \mathbf{A}\mathbf{z} - 2^{\nu_{\mathbf{t}}} c\mathbf{t} \rceil_{\nu_{\mathbf{w}}} + \mathbf{h}.$$

Define $\mathbf{h}_0 := \iota(\mathbf{w}) - (\mathbf{A}\mathbf{z} - c\mathbf{t}_0) \pmod q$*. Then* $(c, \mathbf{z}, \mathbf{h}_0)$ *satisfies the corresponding non-rounded verification equation and* $\|(\mathbf{z}, \mathbf{h}_0)\|_2 \le B_2^\star + \Delta_{\mathsf{rnd}}$*.*

Lemma 5 (Entropy of a shifted Raccoon commitment). *Suppose that*

$$\sigma_{\mathbf{w}'} > 2n\, q^{\frac{1}{k+\ell} + \frac{2}{n\ell}} \qquad \text{and} \qquad \nu_{\mathbf{w}} < \log_2(q) - 2,$$

which are the hypotheses of Lemma 10. Except with probability at most 2^{-n+1} *over the initial choice of* $\mathbf{A}$*, the following holds simultaneously for every signing session. Condition on the protocol history immediately before the fresh sample* $(\mathbf{r}', \mathbf{e}')$ *is drawn, and let* $(\mathbf{r}, \mathbf{e})$ *be any admissible shift fixed by this history. Then both*

$$\mathbf{w}_{\mathsf{raw}} = \mathbf{A}(\mathbf{r}' + \mathbf{r}) + \mathbf{e}' + \mathbf{e}$$

and

$$\mathbf{w} = \lfloor \mathbf{w}_{\mathsf{raw}} \rceil_{\nu_{\mathbf{w}}}$$

have conditional min-entropy at least $n - 1$*. Both lemmas are proven in Appendix B.1.*

Theorem 6 (One-More Unforgeability). *Let $\mathcal{A}$ be a PPT adversary against the one-more unforgeability of* BRaccoon *(Definition 8) that initiates at most Q_s signing sessions. Let Q_H bound the number of non-signing evaluations of the challenge oracle, let Q_{setup} bound the number of setup-expansion oracle queries, and let κ denote the length of the public setup seeds.*

Assume that the commitment scheme CS *satisfies setup indistinguishability and extractability, that π_1 is sound, that π_2 is zero-knowledge, and that* HE *satisfies chosen-key circuit privacy as in Section A.1. Moreover, assume that the parameters satisfy the conditions of Lemma 5, and let*

$$B \geq \max\left\{B_2^\star + \Delta_{\mathsf{rnd}}, B_{2,\mathsf{corr}}^{(0)} + B_{\mathbf{w}}\right\}.$$

The first term accounts for the rounded-to-non-rounded forgery conversion of Lemma 4, whereas the second ensures that an honestly distributed shifted auxiliary signature satisfies the norm test of the $\mathsf{Raccoon}_0^\star$ signing oracle except with the correctness failure probability $\varepsilon_{\mathsf{corr}}$ of Theorem 4, that is, the probability that over the at most Q_s auxiliary signing queries some first signing sample fails this norm test.

Then there exist PPT adversaries $\mathcal{R}, \mathcal{B}, \mathcal{X}, \mathcal{D}, \mathcal{E}, \mathcal{V}$ such that $\mathcal{R}$ makes at most Q_s signing queries against $\mathsf{Raccoon}_0^\star$ and

$$\begin{aligned}\mathsf{Adv}^{\mathsf{OMUF}}_{\mathsf{BRaccoon},\mathcal{A}}(\lambda) \leq\ & \mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{Raccoon}_0^\star,\mathcal{R}}(\lambda) + \mathsf{Adv}^{\mathsf{SetupInd}}_{\mathsf{CS},\mathcal{B}}(\lambda) + \frac{Q_{\mathsf{setup}}}{2^\kappa} \\ & + Q_s \cdot \Big(\mathsf{Adv}^{\mathsf{Ext}}_{\mathsf{CS},\mathcal{X}}(\lambda) + \mathsf{Adv}^{\mathsf{ZK}}_{\pi_2,\mathcal{D}}(\lambda) + \mathsf{Adv}^{\mathsf{SND}}_{\pi_1,\mathcal{E}}(\lambda) \\ & \qquad + \mathsf{Adv}^{\mathsf{CP}}_{\mathsf{HE},\mathcal{V}}(\lambda)\Big) \\ & + \left(Q_s Q_H + \frac{Q_s(Q_s-1)}{2}\right) 2^{-(n-1)} + 2^{-n+1} + \varepsilon_{\mathsf{corr}} + \varepsilon_{\mathsf{hash}}.\end{aligned}$$

Here $\varepsilon_{\mathsf{hash}}$ bounds collisions of $\mathsf{msg} \mapsto \mathsf{H}(\mathsf{vk}_{\mathsf{rnd}}, \mathsf{msg})$ among the extracted signing queries and the output messages; it is the same event as in Theorem 2, and counting it in both bounds is conservative.

Proof (sketch). The reduction proceeds through the games $\mathbf{G}_0, \ldots, \mathbf{G}_8$ summarized in Figure 7. First, games $\mathbf{G}_1$–$\mathbf{G}_5$ replace the commitment setup by an extractable one, extract the committed tuple $(\mathbf{r}, \mathbf{e}, \mathsf{msg})$ from every accepted first-round commitment, simulate π_2, enforce soundness of π_1, and use chosen-key circuit privacy to sanitize the encrypted response. These transitions give the terms collected in the first two lines of the claimed bound.

Next, games $\mathbf{G}_6$ and $\mathbf{G}_7$ separate the rounded verification key from its non-rounded counterpart and lazily program the rounded challenge points. Lemma 5 ensures that each fresh rounded commitment has sufficient conditional min-entropy, so the probability that a programmed point collides with a previous challenge query is bounded by

$$\left(Q_s Q_H + \frac{Q_s(Q_s-1)}{2}\right) 2^{-(n-1)},$$

in addition to the global bad-matrix probability 2^{-n+1}.

Finally, in $\mathbf{G}_8$, the reduction answers each signing session using its $\mathsf{Raccoon}_0^\star$ signing oracle. Conditioned on the absence of an abort, this simulation is distributed as in $\mathbf{G}_7$. Among the $Q_s + 1$ valid signatures output on distinct messages, at least one message was not submitted to that oracle. Outside the prehash collision event, Lemma 4 converts the corresponding rounded signature into a valid $\mathsf{Raccoon}_0^\star$ forgery with verification bound B. Summing the transition losses, the correctness-failure probability, and $\varepsilon_{\mathsf{hash}}$ yields the stated inequality.

The complete game-by-game proof appears in Appendix E.2.

Corollary 1 (Composed reduction). *Under Theorem 6 and the premises of Theorem 4, there exists $\mathcal{R}_0$ such that*

$$\mathsf{Adv}^{\mathsf{OMUF}}_{\mathsf{BRaccoon},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{Raccoon}_0,\mathcal{R}_0}(\lambda) + \varepsilon_{\mathsf{pre}} + \varepsilon_{\mathsf{hash}} + \varepsilon_{\mathsf{br}} + \varepsilon_{\mathsf{corr}} + \varepsilon_{\mathsf{sim}}^{(0)}.$$

Here

$$\varepsilon_{\mathsf{pre}} := \mathsf{Adv}^{\mathsf{SetupInd}}_{\mathsf{CS},\mathcal{B}}(\lambda) + \frac{Q_{\mathsf{setup}}}{2^{\kappa}} + Q_s\Big(\mathsf{Adv}^{\mathsf{Ext}}_{\mathsf{CS},\mathcal{X}}(\lambda) + \mathsf{Adv}^{\mathsf{ZK}}_{\pi_2,\mathcal{D}}(\lambda) + \mathsf{Adv}^{\mathsf{SND}}_{\pi_1,\mathcal{E}}(\lambda) + \mathsf{Adv}^{\mathsf{CP}}_{\mathsf{HE},\mathcal{V}}(\lambda)\Big).$$

Moreover, $\varepsilon_{\mathsf{br}} := \Big(Q_sQ_H + \frac{Q_s(Q_s-1)}{2}\Big)\,2^{-(n-1)} + 2^{-n+1}$. *With* $Q_H^{(0)}$ *denoting the actual number of challenge queries made by the intermediate reduction,* $\varepsilon^{(0)}_{\mathsf{sim}} := 2Q_s(Q_H^{(0)} + 2Q_s)2^{-\kappa_w} + Q_s\varepsilon_{\mathsf{ent}}$. *There are at most* Q_s *signing queries.*

Proof. Apply Theorem 4 to the reduction $\mathcal{R}$ of Theorem 6, at the same verification bound B and with $\kappa_w = n-1$ and $\varepsilon_{\mathsf{ent}} = 0$ outside the bad-matrix event of Lemma 5, which is already counted in $\varepsilon_{\mathsf{br}}$. The first reduction absorbs the removal of rounding in Δ_{rnd}, while the second transfers the resulting $\mathsf{Raccoon}^{\star}_0$ forgery to $\mathsf{Raccoon}_0$ without enlarging the verification bound. Thus Theorem 2 applies to $\mathcal{R}_0$ with $B_2^{(0)} = B \geq B^{(0)}_{2,\mathsf{corr}}$ and SelfTargetMSIS norm parameter $B + \sqrt{\omega}$; its own losses are those of that theorem and are not included in the displayed bound.

Theorem 7 (Blindness). *Let* $\mathcal{A}$ *be any* PPT *adversary that wins the blindness experiment (Definition 7) instantiated with our blind signature in Figure 1 with advantage* $\mathsf{Adv}^{\mathsf{BLD}}_{\mathsf{BRaccoon},\mathcal{A}}(\lambda)$. *Then there exist PPT adversaries* $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$ *and* $\mathcal{E}$, *one for each game hop of the proof, such that, for the security parameter* λ, *we have*

$$\mathsf{Adv}^{\mathsf{BLD}}_{\mathsf{BRaccoon},\mathcal{A}}(\lambda) \leq 2\big(\mathsf{Adv}^{\mathsf{ZK}}_{\pi_1,\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{HID}}_{\mathsf{CS},\mathcal{C}}(\lambda) + \mathsf{Adv}^{\mathsf{KS}}_{\pi_2,\mathcal{D}}(\lambda) + \mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{HE},\mathcal{E}}(\lambda)\big) + \epsilon_{\mathsf{Stat}},$$

where the factor 2 *accounts for the two sessions of the blindness experiment, each hop being applied to one session at a time, and, for any* $\epsilon \leq 1/4$ *with* $\sigma_{\mathbf{w}} \geq \eta_{\epsilon}(\mathbb{Z}^{n(\ell+k)})$, $\epsilon_{\mathsf{Stat}} := 4\sqrt{\pi}\,B_{\mathbf{w}'}/\sigma_{\mathbf{w}} + 2\sqrt{6\epsilon}$ *is the statistical distance introduced by masking the signer's randomness with the user's blinding vector.*

Proof (Sketch). We proceed through hybrid games, applying each hop to the two sessions in turn, whence the factor 2. Starting from the real experiment, we replace the user proof π_1 by a simulated one (zero knowledge of π_1, adversary $\mathcal{B}$), then the commitment to $(\mathsf{msg}, \mathbf{r}, \mathbf{e})$ by a commitment to an independent message (hiding of CS, adversary $\mathcal{C}$). Next, after verifying π_2 the challenger extracts the signer's witness $(\mathbf{s}, \mathbf{r}')$ (knowledge soundness of π_2, adversary $\mathcal{D}$) and recomputes the response as $\mathbf{z} = c\mathbf{s} + \mathbf{r}' + \mathbf{r}$, which is distributed exactly as in the protocol and no longer uses the decryption key. This allows replacing ctx_c by an encryption of an independent challenge (IND-CPA security of HE, adversary $\mathcal{E}$). In the final game the only dependence on the signer's per-session randomness $(\mathbf{r}', \mathbf{e}')$ is through its sum with the user's blinding vector $(\mathbf{r}, \mathbf{e})$, which is statistically hidden by the width $\sigma_{\mathbf{w}}$ up to ϵ_{Stat}. The full proof appears in Appendix E.3.

5 Concrete Instantiations and Efficiency

We now instantiate the protocol of Section 4: we fix the linearly homomorphic encryption scheme, the hash functions, the extractable commitment and the proof system, then give parameters for 128 bits of security and estimate the communication cost.

5.1 Building Blocks and Proof Architecture

The main technical difficulty is to instantiate the zero-knowledge proofs for L_1 and L_2, whose statements contain two different types of constraints. The module-lattice relations (commitments, encryptions, linear evaluations, norm bounds) are naturally handled by lattice-based proof systems. The challenge generation, a hash evaluation followed by the deterministic procedure SampleInBall that

maps a hash value to a sparse challenge $c \in \mathcal{C}$, is non-linear and is more naturally expressed as arithmetic and Boolean circuit constraints.

We encode each proof statement as a mixed R1CS and use the zero-knowledge LaBRADOR toolkit of Biasioli *et al.* [BBL+26]. Modular lattice equations are lifted to bounded integer equations and placed in the arithmetic part of the R1CS; hashes, bit decompositions, range checks, and the challenge sampler are placed in its binary part. The two parts share one binary encoding of the witness and are reduced to a single LaBRADOR dot-product relation using the mixed-R1CS reduction of [BS23].

We use the linearly homomorphic MLWE encryption scheme described in Section A.2. Namely, from an encryption of the challenge representation c and an encryption of the first-round randomness $\mathbf{r}'$, the signer homomorphically obtains an encryption of

$$\mathbf{z}' = c \cdot \mathbf{s} + \mathbf{r}',$$

where $\mathbf{s}$ is the signing key. To obtain circuit privacy, the encryption of $\mathbf{r}'$ uses fresh randomness large enough to flood the noise already present in the evaluated ciphertext.

The challenge-generation relation is parameterized by a hash family $\mathcal{H} = (H_{\mathsf{msg}}, H_{\mathsf{chal}}, \mathsf{XOF})$. The user computes

$$\mu = H_{\mathsf{msg}}(\mathsf{seed}_{\mathsf{vk}}, \mathsf{msg}), \qquad \rho = H_{\mathsf{chal}}(\mu, \mathbf{w}), \qquad c = \mathsf{SampleInBall}_{\mathsf{XOF}}(\rho),$$

where $\mathsf{seed}_{\mathsf{vk}}$ is the public seed representing the verification key. The hash family is fixed by the public parameters. Our instantiation sets both hash functions to SHA3-256 and uses SHAKE256 as the XOF. The resulting Boolean and arithmetic constraints are proved as a mixed R1CS using ZK-LaBRADOR [BBL+26]. In the final signature verification algorithm, the verifier recomputes these values directly from msg, vk, and the signature commitment.

The protocol requires two proofs: one ZK-LaBRADOR proof for the complete user-side language L_1, and one for the complete signer-side language L_2. Appendix I gives the challenge-sampling algorithm, while Appendices G and H give the full relations and their mixed-R1CS encoding.

5.2 Instantiating L_1

The language L_1 proves correctness of the user's computation after receiving the signer's first-round value $\mathbf{w}'$. The user must prove that the blinded value $\mathbf{w}$ is correctly derived from $\mathbf{w}'$, that the challenge representation c is obtained from the committed values, and that c is correctly encrypted.

We write $\mathsf{BDLOP.Verify}(\mathsf{ck}, m, C) = 1$ to denote the relation stating that C is a valid extractable BDLOP commitment to m. The opening is part of the witness and is left implicit. Similarly, $\mathsf{HE.Enc}$ and $\mathsf{HE.Eval}$ denote the corresponding MLWE encryption and evaluation relations, including the randomness and noise witnesses. The fully expanded relations are given in Appendix G.

The user-side language is:

$$\mathsf{L}_1 := \left\{ \begin{array}{l|l} & \exists(\mathsf{msg}, \mu, \rho, c, \mathbf{r}, \mathbf{e}, \mathbf{w}, \mathbf{e_w}) \text{ s.t.} \\ & \mathsf{BDLOP.Verify}(\mathsf{ck}, (\mathsf{msg}, \mathbf{r}, \mathbf{e}), (\mathbf{t_A}, \mathbf{t_B})) = 1, \\ & \mathbf{w}' = 2^{\nu_{\mathbf{w}}}\mathbf{w} + \mathbf{e_w} - \mathbf{A}\mathbf{r} - \mathbf{e} \pmod{q_{\mathsf{sig}}}, \\ ((\mathbf{t_A}, \mathbf{t_B}), \mathsf{ctx}_c, \mathbf{w}', & \mathsf{ctx}_c = \mathsf{HE.Enc}(\mathsf{pp}, \mathsf{ek}, c), \\ \mathsf{vk} = (\mathbf{A}, \mathbf{t}), \mathsf{ek}, \mathsf{pp}, B_{\mathbf{w}}) & \mu = H_{\mathsf{msg}}(\mathsf{seed}_{\mathsf{vk}}, \mathsf{msg}), \\ & \rho = H_{\mathsf{chal}}(\mu, \mathbf{w}), \\ & c = \mathsf{SampleInBall}_{\mathsf{XOF}}(\rho), \\ & \|(\mathbf{r}, \mathbf{e})\|_2 \le B_{\mathbf{w}}, \\ & \|\mathbf{e_w}\|_\infty < 2^{\nu_{\mathbf{w}}-1}. \end{array} \right\}.$$

The error term $\mathbf{e_w}$ makes the rounding explicit. Equivalently, the proof shows that

$$\mathbf{w} = \lfloor \mathbf{w}' + \mathbf{A}\mathbf{r} + \mathbf{e} \rceil_{\nu_{\mathbf{w}}},$$

Table 3. Concrete BRaccoon parameters targeting 128-bit security for $Q_s \in \{2^{20}, 2^{32}, 2^{64}\}$. For parameters depending on Q_s, values are reported in this order.

Parameter	$Q_s = 2^{20},\ 2^{32},\ 2^{64}$
Ring degree n	256
Signature modulus $\log_2 q_{\mathsf{sig}}$	66
BGV modulus $\log_2 Q_{\mathsf{BGV}}$	82
Commitment modulus $\log_2 Q_{\mathsf{ext}}$	78
Public dimension k	12
Secret dimension ℓ	(14, 16, 22)
Challenge weight ω	23
$\sigma_{\mathbf{t}}$	2^{10}
$\sigma_{\mathbf{w}'}$	2^{17}
$\sigma_{\mathbf{w}}$	2^{57}
$\nu_{\mathbf{t}}$	52
$\nu_{\mathbf{w}}$	55
Verification key (KB)	5.39
Signature (KB)	(28.00, 31.78, 43.10)

but expressed as a linear equation plus a small coefficient-wise rounding bound.

Thus, the lattice, encryption, hash, and SampleInBall constraints share the same witness in one mixed proof; no auxiliary linking commitment is needed.

5.3 Instantiating L_2

The signer-side proof enforces honest computation of the encrypted response. After receiving ctx_c, the signer proves that the returned ciphertext encrypts

$$\mathbf{z}' = c \cdot \mathbf{s} + \mathbf{r}',$$

and that the first-round value $\mathbf{w}'$ was generated from short randomness. The statement is purely linear over module lattices once the encryption randomness and noise terms are made explicit. Compared with the definition of L_2 in Section 4, the only change is that the rounding relation $\mathbf{t} = \lfloor \mathbf{A}\mathbf{s} + \mathbf{e}_{\mathsf{kg}} \rceil_{\nu_{\mathbf{t}}}$ is linearized as $\mathbf{A}\mathbf{s} + \mathbf{e}_{\mathsf{kg}} = 2^{\nu_{\mathbf{t}}}\mathbf{t} + \mathbf{e}_{\mathbf{t}} \pmod{q_{\mathsf{sig}}}$ with $\|\mathbf{e}_{\mathbf{t}}\|_\infty < 2^{\nu_{\mathbf{t}}-1}$; the full relation is given in Appendix G. The flooding distribution D_{fl} is chosen so that the ciphertext of $\mathbf{r}'$ statistically hides the evaluation noise while preserving decryption correctness.

5.4 Concrete Parameters and Communication Cost

Table 3 gives concrete BRaccoon parameters[7] targeting 128 bits of security for several bounds Q_s on the number of signing queries. The dimensions are chosen so that key-recovery and forgery security reduce to concretely hard instances of MLWE and MSIS at the target security level.

The concrete parameters also satisfy the entropy requirement used in the OMUF reduction. Indeed, for all three parameter sets, $n = 256$, $\log_2(q_{\mathsf{sig}}) = 66$, $k = 12$, $\ell \in \{14, 16, 22\}$, $\sigma_{\mathbf{w}'} = 2^{17}$, and $\nu_{\mathbf{w}} = 55$. Hence

$$\sigma_{\mathbf{w}'} > 2n\, q_{\mathsf{sig}}^{\frac{1}{k+\ell}+\frac{2}{n\ell}} \qquad \text{and} \qquad \nu_{\mathbf{w}} < \log_2(q_{\mathsf{sig}}) - 2,$$

as required by Lemma 5 and Theorem 2: the right-hand side of the first condition is below 2^{12} for all three sets. Consequently, the shifted raw and rounded commitments have min-entropy at least $n - 1 = 255$, except with probability at most 2^{-255}.

[7] We provide a Sage script that verifies the security for our parameters, both for the signature scheme and the other lattice-based building blocks: : https://github.com/Heltorfx/BRaccoon-Parameters.

Table 4. Estimated communication cost of BRaccoon (KB). Proof sizes use the approximately 110 KB zero-knowledge size reported in [BBL+26]; the ciphertext column counts one encryption of c and ℓ component-wise encryptions of $\mathbf{z}'$, including the flooding bound $B_{\mathsf{fl}} = 2^{45}$.

Q_s	$\mathsf{NIZK}_{\mathsf{L}_1}$	$\mathsf{NIZK}_{\mathsf{L}_2}$	Ciphertexts	Extr. commitment	$\mathbf{w}'$	Total
2^{20}	≈ 110	≈ 110	$15 \cdot 28.78 = 431.70$	116.74	25.34	≈ 793.78
2^{32}	≈ 110	≈ 110	$17 \cdot 28.80 = 489.60$	126.76	25.34	≈ 861.70
2^{64}	≈ 110	≈ 110	$23 \cdot 28.84 = 663.32$	156.87	25.34	≈ 1065.53

How the parameters scale with Q_s. Across the three parameter sets, the signature modulus, the Gaussian widths, the rounding parameters and the challenge weight are shared; only the secret dimension ℓ changes, together with a one-bit adjustment of the encryption modulus Q_{BGV} that follows from it. The number of signing queries enters the analysis in two places. The union-bound terms $Q_s Q_H 2^{-(n-1)}$ of Theorems 2 and 6 are far from binding for $n = 256$ even at $Q_s = 2^{64}$. The binding constraint is the H-MLWE assumption of Theorem 2, whose instance carries Q_s hints $c_i\mathbf{s} + \mathbf{r}'_i$: the reduction from H-MLWE to plain MLWE loses a factor in the effective Gaussian width that grows with Q_s [KLSS23], and we compensate by increasing ℓ (14, 16, 22) rather than the widths. This keeps q_{sig}, $\nu_{\mathbf{t}}$, $\nu_{\mathbf{w}}$, the verification key and hence the verifier unchanged, and lets the signature grow linearly in ℓ, from 28.0 to 43.1 KB. Blindness is statistical and independent of Q_s; its level is set by $\sigma_{\mathbf{w}}/B_{\mathbf{w}'}$ (Theorem 7).

We treat $Q_s = 2^{20}$ as the primary parameter set for two reasons. First, it is the budget of the most efficient prior scheme [BLNS23a], which keeps Table 1 comparable. Second, a blind-signature key typically issues far fewer signatures than a general-purpose signing key, and deployments such as anonymous tokens and e-cash rotate issuer keys per epoch. The 2^{32} and 2^{64} sets show that meeting the 2^{64} budget that NIST requires for standard signatures costs about 54% in signature size and 32% in communication, with no change to the verification key or to the verification algorithm.

The zero-knowledge LaBRADOR implementation of Biasioli *et al.* reports proofs of approximately 110 KB across its case studies, with little dependence on the initial relation because the final recursive rounds dominate the proof size [BBL+26]. We use this value once for the complete user proof π_1 and once for the complete signer proof π_2. This is an analytical estimate for our relations.

Table 4 summarizes the resulting communication cost. The zero-knowledge machinery is confined to the interactive rounds; the final output remains a standard Raccoon-shaped signature. Note that our current instantiation uses a linearly homomorphic encryption scheme that encrypts each coefficient of a vector in $\mathcal{R}_q$ separately. This coefficient-wise representation is a major source of communication overhead. A more suitable MLWE-based vector linearly homomorphic encryption scheme, supporting direct encryption and linear operations on vectors over $\mathcal{R}_q$, could substantially reduce this cost. In such an idealized instantiation, we estimate that the total communication could decrease from roughly 800–1100 KB to about 400–600 KB.

Acknowledgments. We want to thank Patrick Hough for help with the Sage scripts and Guilhem Niot for his helpful suggestions regarding Raccoon*. This work was supported in part by the EU Digital Europe Programme under Grant Agreement No. 101190366 (PiQASO), the project LIONS within dtec.bw, funded by the European Commission under the NextGenerationEU program.

References

ABD16. Martin R. Albrecht, Shi Bai, and Léo Ducas. A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 153–178. Springer, Berlin, Heidelberg, August 2016.

ADK24. Shahla Atapoor, Cyprien Delpech de Saint Guilhem, and Al Kindi. STARK-based signatures from the RPO permutation. Cryptology ePrint Archive, Report 2024/1553, 2024.

AEB20. Nabil Alkeilani Alkadri, Rachid El Bansarkhani, and Johannes Buchmann. BLAZE: Practical lattice-based blind signatures for privacy-preserving applications. In Joseph Bonneau and Nadia Heninger, editors, *FC 2020*, volume 12059 of *LNCS*, pages 484–502. Springer, Cham, February 2020.

AGJ+24. Sven Argo, Tim Güneysu, Corentin Jeudy, Georg Land, Adeline Roux-Langlois, and Olivier Sanders. Practical post-quantum signatures for privacy. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024*, pages 1523–1537. ACM Press, October 2024.

AHJ21. Nabil Alkeilani Alkadri, Patrick Harasser, and Christian Janson. BlindOR: an efficient lattice-based blind signature scheme from OR-proofs. In Mauro Conti, Marc Stevens, and Stephan Krenn, editors, *CANS 21*, volume 13099 of *LNCS*, pages 95–115. Springer, Cham, December 2021.

AKSY22. Shweta Agrawal, Elena Kirshanova, Damien Stehlé, and Anshu Yadav. Practical, round-optimal lattice-based blind signatures. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 39–53. ACM Press, November 2022.

ALP26. Martin R. Albrecht, Russell W. F. Lai, and Eamonn W. Postlethwaite. Hardness of hinted ISIS from the space-time hardness of lattice problems. Cryptology ePrint Archive, Paper 2026/187, 2026.

ASY22. Shweta Agrawal, Damien Stehlé, and Anshu Yadav. Round-optimal lattice-based threshold signatures, revisited. Cryptology ePrint Archive, Report 2022/634, 2022.

BBB+26. Carsten Baum, Marvin Beckmann, Ward Beullens, Shibam Mukherjee, and Christian Rechberger. Concretely efficient blind signatures based on VOLE-in-the-head proofs and the MAYO trapdoor. Cryptology ePrint Archive, Paper 2026/109, 2026.

BBL+26. Beatrice Biasioli, Madalina Bolboceanu, Vadim Lyubashevsky, Antonio Merino-Gallardo, Michal Osadnik, Gregor Seiler, and Patrick Steuer. A toolkit for succinct lattice-based zero knowledge proofs. Cryptology ePrint Archive, Paper 2026/1289, 2026. To appear at ACM CCS 2026.

BDL+18. Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More efficient commitments from structured lattice assumptions. In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 368–385. Springer, Cham, September 2018.

BFM+25. Charles Bouillaguet, Thibauld Feneuil, Jules Maire, Matthieu Rivain, Julia Sauvage, and Damien Vergnaud. Blinding post-quantum hash-and-sign signatures. Cryptology ePrint Archive, Report 2025/895, 2025.

BGV12. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, January 2012.

BK26. Dan Boneh and Jaehyung Kim. A universal blinder: One-round blind signatures from FHE. Cryptology ePrint Archive, Paper 2026/574, 2026.

BL13. Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013*, pages 1087–1098. ACM Press, November 2013.

BLL+15. Shi Bai, Adeline Langlois, Tancrède Lepoint, Damien Stehlé, and Ron Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 3–24. Springer, Berlin, Heidelberg, November / December 2015.

BLL+21. Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, and Mariana Raykova. On the (in)security of ROS. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 33–53. Springer, Cham, October 2021.

BLNS23a. Ward Beullens, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Lattice-based blind signatures: Short, efficient, and round-optimal. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023*, pages 16–29. ACM Press, November 2023.

BLNS23b. Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Alessandro Sorniotti. A framework for practical anonymous credentials from lattices. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 384–417. Springer, Cham, August 2023.

BMW26. Olivier Blazy, Lola-Baie Mallordy, and Weiqiang Wen. Concurrently secure compact blind signatures from module-SIS. Cryptology ePrint Archive, Paper 2026/1690, 2026.

Bol03. Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 31–46. Springer, Berlin, Heidelberg, January 2003.

BS23. Ward Beullens and Gregor Seiler. LaBRADOR: Compact proofs for R1CS from module-SIS. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 518–548. Springer, Cham, August 2023.

BV11. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, October 2011.

CAHL+22. Rutchathon Chairattana-Apirom, Lucjan Hanzlik, Julian Loss, Anna Lysyanskaya, and Benedikt Wagner. PI-cut-choo and friends: Compact blind signatures via parallel instance cut-and-choose and more. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 3–31. Springer, Cham, August 2022.

Cha82. David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO'82*, pages 199–203. Plenum Press, New York, USA, 1982.

CJL16. Jung Hee Cheon, Jinhyuck Jeong, and Changmin Lee. An algorithm for ntru problems and cryptanalysis of the ggh multilinear map without a low-level encoding of zero. *LMS Journal of Computation and Mathematics*, 19(A):255–266, 2016.

CL03. Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN 02*, volume 2576 of *LNCS*, pages 268–289. Springer, Berlin, Heidelberg, September 2003.

dEK+23. Rafael del Pino, Thomas Espitau, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, Mélissa Rossi, and Markku-Juhani Saarinen. Raccoon. Technical report, National Institute of Standards and Technology, 2023. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures.

DGS+18. Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. Privacy pass: Bypassing internet challenges anonymously. *PoPETs*, 2018(3):164–180, July 2018.

DHP24. Khue Do, Lucjan Hanzlik, and Eugenio Paracucchi. M&M'S: Mix and match attacks on Schnorr-type blind signatures with repetition. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part VI*, volume 14656 of *LNCS*, pages 363–387. Springer, Cham, May 2024.

DKL+18. Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR TCHES*, 2018(1):238–268, 2018.

DKM+24. Rafaël Del Pino, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, and Markku-Juhani O. Saarinen. Threshold raccoon: Practical threshold signatures from standard lattice assumptions. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 219–248. Springer, Cham, May 2024.

dPK22. Rafaël del Pino and Shuichi Katsumata. A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 306–336. Springer, Cham, August 2022.

dPKPR24. Rafaël del Pino, Shuichi Katsumata, Thomas Prest, and Mélissa Rossi. Raccoon: A masking-friendly signature proven in the probing model. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part I*, volume 14920 of *LNCS*, pages 409–444. Springer, Cham, August 2024.

dPPRS23. Rafaël del Pino, Thomas Prest, Mélissa Rossi, and Markku-Juhani O. Saarinen. High-order masking of lattice signatures in quasilinear time. In *2023 IEEE Symposium on Security and Privacy*, pages 1168–1185. IEEE Computer Society Press, May 2023.

Fas26. Matteo Frigo and abhi shelat. Anonymous credentials from ECDSA. *IACR Communications in Cryptology*, 3(1), 2026.

FHKS16. Georg Fuchsbauer, Christian Hanser, Chethan Kamath, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model from weaker assumptions. In Vassilis Zikas and Roberto De Prisco, editors, *SCN 16*, volume 9841 of *LNCS*, pages 391–408. Springer, Cham, August / September 2016.

Fis06. Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 60–77. Springer, Berlin, Heidelberg, August 2006.

FPS20. Georg Fuchsbauer, Antoine Plouviez, and Yannick Seurin. Blind Schnorr signatures and signed ElGamal encryption in the algebraic group model. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 63–95. Springer, Cham, May 2020.

FR25. Thibauld Feneuil and Matthieu Rivain. CAPSS: A framework for SNARK-friendly post-quantum signatures. Cryptology ePrint Archive, Report 2025/061, 2025.

FW24. Georg Fuchsbauer and Mathias Wolf. Concurrently secure blind Schnorr signatures. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 124–160. Springer, Cham, May 2024.

Gen09. Craig Gentry. *A Fully Homomorphic Encryption Scheme*. Ph.d. dissertation, Stanford University, Stanford, CA, USA, September 2009.

GHK+25. Kamil Doruk Gur, Patrick Hough, Jonathan Katz, Caroline Sandsbråten, and Tjerand Silde. Olingo: Threshold lattice signatures with DKG and identifiable abort. Cryptology ePrint Archive, Report 2025/1789, 2025.

GKS24. Kamil Doruk Gür, Jonathan Katz, and Tjerand Silde. Two-round threshold lattice-based signatures from threshold homomorphic encryption. In Markku-Juhani Saarinen and Daniel Smith-Tone, editors, *PQCrypto 2024, Part II*, pages 266–300. Springer, Cham, June 2024.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

HAB+17. Ethan Heilman, Leen Alshenibr, Foteini Baldimtsi, Alessandra Scafuro, and Sharon Goldberg. TumbleBit: An untrusted Bitcoin-compatible anonymous payment hub. In *NDSS 2017*. The Internet Society, February / March 2017.

HKL19. Eduard Hauck, Eike Kiltz, and Julian Loss. A modular treatment of blind signatures from identification schemes. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 345–375. Springer, Cham, May 2019.

HKLN20. Eduard Hauck, Eike Kiltz, Julian Loss, and Ngoc Khanh Nguyen. Lattice-based blind signatures, revisited. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 500–529. Springer, Cham, August 2020.

HLM+25. Lucjan Hanzlik, Yi-Fu Lai, Marzio Mula, Eugenio Paracucchi, Daniel Slamanig, and Gang Tang. Tanuki: New frameworks for (concurrently secure) blind signatures from post-quantum group actions. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part IV*, volume 16248 of *LNCS*, pages 35–69. Springer, Singapore, December 2025.

HLPP26. Lucjan Hanzlik, Yi-Fu Lai, Eugenio Paracucchi, and Edoardo Persichetti. Post-quantum blind signature from standard group action assumptions and more. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology – EUROCRYPT 2026*, pages 277–307, Cham, 2026. Springer Nature Switzerland.

HLW23. Lucjan Hanzlik, Julian Loss, and Benedikt Wagner. Rai-choo! Evolving blind signatures to the next level. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 753–783. Springer, Cham, April 2023.

HSS24. Patrick Hough, Caroline Sandsbråten, and Tjerand Silde. More efficient lattice-based electronic voting from NTRU. *CiC*, 1(4):10, 2024.

IKSA03. S. Ibrahim, M. Kamat, M. Salleh, and S.R.A. Aziz. Secure e-voting with blind signature. In *4th National Conference of Telecommunication Technology, 2003. NCTT 2003 Proceedings.*, pages 193–197, 2003.

JRS23. Corentin Jeudy, Adeline Roux-Langlois, and Olivier Sanders. Lattice signature with efficient protocols, application to anonymous credentials. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 351–383. Springer, Cham, August 2023.

JS25a. Corentin Jeudy and Olivier Sanders. Improved lattice blind signatures from recycled entropy. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part I*, volume 16000 of *LNCS*, pages 477–513. Springer, Cham, August 2025.

JS25b. Corentin Jeudy and Olivier Sanders. Worst-case lattice sampler with truncated gadgets and applications. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part III*, volume 16247 of *LNCS*, pages 200–232. Springer, Singapore, December 2025.

KLR26. Michael Klooß, Russell W. F. Lai, and Michael Reichle. Blind signatures from arguments of inequality. In *Advances in Cryptology – CRYPTO 2026*. Springer-Verlag, 2026.

KLSS23. Duhyeong Kim, Dongwon Lee, Jinyeong Seo, and Yongsoo Song. Toward practical lattice-based proof of knowledge from hint-MLWE. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 549–580. Springer, Cham, August 2023.

LNP22a. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Efficient lattice-based blind signatures via gaussian one-time signatures. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part II*, volume 13178 of *LNCS*, pages 498–527. Springer, Cham, March 2022.

LNP22b. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 71–101. Springer, Cham, August 2022.

LPR13. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 35–54. Springer, Berlin, Heidelberg, May 2013.

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *DCC*, 75(3):565–599, 2015.

LSS24. Vadim Lyubashevsky, Gregor Seiler, and Patrick Steuer. The LaZer library: Lattice-based zero knowledge and succinct proofs for quantum-safe privacy. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024*, pages 3125–3137. ACM Press, October 2024.

Lyu09. Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Berlin, Heidelberg, December 2009.

MPD25. Jules Maire and Alan Pulval-Dady. Blind ECDSA from the ECDSA assumption. *CiC*, 2(3):38, 2025.

MR04. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 372–381, 2004.

Nat15. National Institute of Standards and Technology. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Technical Report Federal Information Processing Standards Publication 202, National Institute of Standards and Technology, 2015.

NS26. Ngoc Khanh Nguyen and Jan Niklas Siemer. Tight reductions for sis-with-hints assumptions with applications to anonymous credentials. In Shi Bai and Edoardo Persichetti, editors, *Public-Key Cryptography – PKC 2026*, pages 103–135, Cham, 2026. Springer Nature Switzerland.

Pei16. Chris Peikert. A decade of lattice cryptography. *Foundations and Trends® in Theoretical Computer Science*, 10(4):283–424, 2016.

PFH+22. Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

PKN+25. Rafaël Del Pino, Shuichi Katsumata, Guilhem Niot, Michael Reichle, and Kaoru Takemure. Unmasking TRaccoon: A lattice-based threshold signature with an efficient identifiable abort protocol. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 423–456. Springer, Cham, August 2025.

PPZ24. Christian Paquin, Guru-Vamsi Policharla, and Greg Zaverucha. Crescent: Stronger privacy for existing credentials. Cryptology ePrint Archive, Report 2024/2013, 2024.

PS96. David Pointcheval and Jacques Stern. Provably secure blind signature schemes. In Kwangjo Kim and Tsutomu Matsumoto, editors, *ASIACRYPT'96*, volume 1163 of *LNCS*, pages 252–265. Springer, Berlin, Heidelberg, November 1996.

PS00. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000.

PWFW23. Guru-Vamsi Policharla, Bas Westerbaan, Armando Faz-Hernández, and Christopher A Wood. Post-quantum privacy pass via post-quantum anonymous credentials. Cryptology ePrint Archive, Report 2023/414, 2023.

Rüc10. Markus Rückert. Lattice-based blind signatures. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 413–430. Springer, Berlin, Heidelberg, December 2010.

SU12. Dominique Schröder and Dominique Unruh. Security of blind signatures revisited. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 662–679. Springer, Berlin, Heidelberg, May 2012.

WOS+25. Anna P. Y. Woo, Alex Ozdemir, Chad Sharp, Thomas Pornin, and Paul Grubbs. Efficient proofs of possession for legacy signatures. Cryptology ePrint Archive, Report 2025/538, 2025.

ZSE+24. Xinyu Zhang, Ron Steinfeld, Muhammed F. Esgin, Joseph K. Liu, Dongxi Liu, and Sushmita Ruj. Loquat: A SNARK-friendly post-quantum signature based on the Legendre PRF with applications in ring and aggregate signatures. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part I*, volume 14920 of *LNCS*, pages 3–38. Springer, Cham, August 2024.

A Extended Background

A.1 Homomorphic Encryption Scheme

Homomorphic Encryption Scheme. A homomorphic (public key) encryption scheme HE is a 5-tuple of PPT algorithms $(\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ defined as follows. On input the security parameter $\lambda \in \mathbb{N}$, the setup algorithm $\mathsf{HE.Setup}(1^\lambda)$ outputs public parameters pp, which include a description of the message space $\mathcal{M}$. The key generation algorithm $\mathsf{HE.KGen}(\mathsf{pp})$ outputs a public/secret key pair $(\mathsf{ek}, \mathsf{dk})$. Given pp, a public key ek, and a message $\mathsf{msg} \in \mathcal{M}$, the encryption algorithm $\mathsf{HE.Enc}(\mathsf{pp}, \mathsf{ek}, \mathsf{msg})$ outputs a ciphertext $\mathsf{ctx}_{\mathsf{msg}}$. Given pp, ek, a function $f : \mathcal{M}^n \to \mathcal{M}$ from an admissible function class $\mathcal{F}$, and ciphertexts $(\mathsf{ctx}_1, \ldots, \mathsf{ctx}_n)$, the evaluation algorithm $\mathsf{HE.Eval}(\mathsf{pp}, \mathsf{ek}, f, (\mathsf{ctx}_1, \ldots, \mathsf{ctx}_n))$ outputs a ciphertext ctx_f. Finally, the decryption algorithm $\mathsf{HE.Dec}(\mathsf{pp}, \mathsf{dk}, \mathsf{ctx}_{\mathsf{msg}})$ outputs the message $\mathsf{msg} \in \mathcal{M}$ or $\perp$ in case of decryption failure.

Correctness. A homomorphic encryption scheme is *correct* if for all $\lambda \in \mathbb{N}$,

$$\Pr\left[\mathsf{Dec}(\mathsf{pp}, \mathsf{dk}, \mathsf{Enc}(\mathsf{pp}, \mathsf{ek}, \mathsf{msg})) = \mathsf{msg} \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda), \\ (\mathsf{ek}, \mathsf{dk}) \leftarrow\!\$\ \mathsf{KGen}(\mathsf{pp}), \\ \mathsf{msg} \leftarrow\!\$\ \mathcal{M} \end{array}\right] = 1$$

and for any function $f \in \mathcal{F}$,

$$\Pr\left[\begin{array}{r} \mathsf{Dec}(\mathsf{Eval}(\mathsf{pp}, \mathsf{ek}, f, (\mathsf{ctx}_1, \ldots, \mathsf{ctx}_n))) \\ = f(\mathsf{msg}_1, \ldots, \mathsf{msg}_n) \end{array} \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda), \\ (\mathsf{ek}, \mathsf{dk}) \leftarrow\!\$\ \mathsf{KGen}(\mathsf{pp}), \\ (\mathsf{msg}_1, \ldots, \mathsf{msg}_n) \leftarrow\!\$\ \mathcal{M}^n, \\ \forall i \in [n], \mathsf{ctx}_i \leftarrow\!\$\ \mathsf{Enc}(\mathsf{pp}, \mathsf{ek}, \mathsf{msg}_i) \end{array}\right] = 1 - \mathsf{negl}(\lambda).$$

IND-CPA Security. We recall the notion of IND-CPA (Indistinguishability under Chosen-Plaintext Attack), which is a standard security definition for encryption schemes. Let $t : \mathbb{N} \to \mathbb{N}$ and $\epsilon : \mathbb{N} \to [0, 1]$. The encryption scheme is said to be (t, ϵ)-IND-CPA secure if for all two-phase adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ running in time at most $t(\lambda)$, we have:

$$\Pr\left[b = b' \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda), (\mathsf{ek}, \mathsf{dk}) \leftarrow\!\$\ \mathsf{KGen}(\mathsf{pp}), \\ (\mathsf{msg}_0, \mathsf{msg}_1, \mathsf{state}) \leftarrow\!\$\ \mathcal{A}_1(\mathsf{pp}, \mathsf{ek}), b \leftarrow\!\$\ \{0, 1\}, \\ c \leftarrow\!\$\ \mathsf{Enc}(\mathsf{pp}, \mathsf{ek}, \mathsf{msg}_b), b' \leftarrow\!\$\ \mathcal{A}_2(\mathsf{pp}, c, \mathsf{state}) \end{array}\right] \leq \frac{1}{2} + \epsilon(\lambda).$$

Chosen-Key Circuit Privacy on Admissible Inputs. Circuit privacy is not part of the default requirements on a homomorphic encryption scheme; we single it out because blindness of our blind signature relies on it. The standard notion of circuit privacy [Gen09] considers honestly generated encryption keys and ciphertexts in the image of the encryption algorithm. In our setting, circuit privacy must hold against the party that supplies the encryption key. This party may know a corresponding decryption key, the key-generation coins, and the input-encryption coins. We therefore use the following chosen-key variant, restricted to admissible inputs.

Let $\mathsf{HE.Eval}_{\mathsf{priv}}$ denote the configured private evaluation algorithm, including its fresh flooding randomness. An instance

$$\big(\mathsf{pp}, \mathsf{ek}, f, (m_i, \mathsf{ctx}_i)_{i=1}^a\big)$$

is called admissible if ek has the prescribed syntactic shape, f belongs to the bounded function family, and there exist coins $(\theta_i)_{i=1}^a$ such that

$$\mathsf{ctx}_i = \mathsf{HE.Enc}(\mathsf{pp}, \mathsf{ek}, m_i; \theta_i) \qquad \text{for every } i \in [a],$$

where the plaintexts, coins, function coefficients, and induced evaluation shifts satisfy the prescribed admissibility bounds. Admissibility is a promise on the quantified instances and need not be efficiently decidable.

Let $\mathsf{Sim}_{\mathsf{HE}}$ be a randomized PPT algorithm whose output depends only on $(\mathsf{pp}, \mathsf{ek}, y)$. For a two-stage PPT distinguisher $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$ whose first-stage outputs are admissible, define $\mathsf{CP}_b^{\mathcal{D}}(1^\lambda)$ as follows:

1. Sample
$$\mathsf{pp} \leftarrow \mathsf{HE.Setup}(1^\lambda)$$
and
$$(\mathsf{ek}, f, (m_i, \mathsf{ctx}_i)_{i=1}^a, \mathsf{state}) \leftarrow \mathcal{D}_1(\mathsf{pp}).$$
Set $y := f(m_1, \ldots, m_a)$.
2. If $b = 0$, sample
$$\mathsf{ctx}^* \leftarrow \mathsf{HE.Eval}_{\mathsf{priv}}(\mathsf{pp}, \mathsf{ek}, f, (\mathsf{ctx}_i)_{i=1}^a).$$
If $b = 1$, sample
$$\mathsf{ctx}^* \leftarrow \mathsf{Sim}_{\mathsf{HE}}(\mathsf{pp}, \mathsf{ek}, y).$$
3. Return $\mathcal{D}_2(\mathsf{state}, \mathsf{ctx}^*)$.

The state may contain all first-stage information, including a decryption key, key-generation coins, the function description, and any known input-encryption coins. Define

$$\mathsf{Adv}^{\mathsf{CP}}_{\mathsf{HE},\mathcal{D}}(\lambda) := \left|\Pr[\mathsf{CP}_0^{\mathcal{D}}(1^\lambda) = 1] - \Pr[\mathsf{CP}_1^{\mathcal{D}}(1^\lambda) = 1]\right|.$$

We say that HE satisfies computational chosen-key circuit privacy on admissible inputs if there exists such a PPT simulator $\mathsf{Sim}_{\mathsf{HE}}$ for which this advantage is negligible for every PPT admissible distinguisher $\mathcal{D}$.

For the statistical property used in our flooding argument, we require that, for every fixed admissible key, function, and collection of input openings,

$$\Delta_{\mathrm{TV}}\Big(\mathsf{HE.Eval}_{\mathsf{priv}}\big(\mathsf{pp}, \mathsf{ek}, f, (\mathsf{HE.Enc}(\mathsf{pp}, \mathsf{ek}, m_i; \theta_i))_{i=1}^a\big),\ \mathsf{Sim}_{\mathsf{HE}}\big(\mathsf{pp}, \mathsf{ek}, f(m_1, \ldots, m_a)\big)\Big) \le \varepsilon_{\mathsf{cp}}(\lambda).$$

Since this bound holds pointwise, it continues to hold in the presence of arbitrary auxiliary information fixed before the fresh evaluation coins are sampled.

The admissible-input restriction is essential. By soundness, an accepting proof π_1 guarantees the existence of an admissible opening of the user ciphertext; neither the opening nor its extraction is required. We do not claim privacy for arbitrary ciphertexts without such an opening, nor decryption correctness under malformed keys. For the concrete BGV instantiation,

$$\mathsf{Sim}_{\mathsf{HE}}(\mathsf{pp}, \mathsf{ek}, y) := \mathsf{HE.Enc}(\mathsf{pp}, \mathsf{ek}, y; \mathcal{D}_{\mathsf{fl}}),$$

namely a fresh encryption using the flooding distribution rather than the ordinary input-encryption distribution; see Lemma 8.

A.2 MLWE-Based Linearly Homomorphic Encryption

This subsection specifies the linear homomorphic encryption primitive used in our construction. We present a lightweight MLWE-based encryption scheme, inspired by the BGV construction [BGV12] and

its module variant from [GHK+25], but intentionally restricted to the evaluation of linear functions in the following family:

$$\mathcal{F}_{B_\alpha} = \left\{ \begin{array}{c} f_{\boldsymbol{\alpha}} : \mathcal{M}^2 \to \mathcal{M}, \quad (C, M) \mapsto \boldsymbol{\alpha} \odot C + M, \\ \text{where } \boldsymbol{\alpha} \in \mathcal{R}_p^\ell, \quad \max\limits_{j \in [\ell]} \|\widetilde{\alpha}_j\|_1 \le B_\alpha \end{array} \right\}. \tag{2}$$

Here $\mathcal{M} \coloneqq \mathcal{R}_p^\ell$ is the message space, while $\odot$ denotes component-wise multiplication and $\widetilde{\alpha}_j \in \mathbb{Z}[X]$ is the centered coefficient lift of $\alpha_j \in \mathcal{R}_p$. In our application, $\boldsymbol{\alpha} = \mathbf{s}$ and the bound $\|\mathbf{s}\|_2 \le B_t$ implies $\max_{j \in [\ell]} \|\widetilde{s}_j\|_1 \le \sqrt{n} B_t$, so one may take $B_\alpha = \sqrt{n} B_t$. As in the original BGV construction, a key ingredient is the *modulus reduction* introduced in [BV11]. More precisely, ciphertexts are defined over the larger ring $\mathcal{R}_q$, while plaintexts are recovered modulo the smaller plaintext modulus p. Plaintexts are embedded using fixed centered coefficient lifts modulo p; integer evaluation expressions may require the carries made explicit in Lemma 8.

The scheme is defined in two layers. We first describe a scalar encryption algorithm encrypting messages in $\mathcal{R}_p$. We then extend it coordinate-wise to the message space $\mathcal{M} = \mathcal{R}_p^\ell$. To avoid ambiguity, we denote the scalar encryption and decryption algorithms by $\mathsf{BGV.Enc_{sc}}$ and $\mathsf{BGV.Dec_{sc}}$, respectively, while $\mathsf{BGV.Enc}$ and $\mathsf{BGV.Dec}$ denote their coordinate-wise vector extensions.

By default, encryption uses the noise distribution D specified in the public parameters. With an alternative noise distribution D', we write

$$\mathsf{BGV.Enc_{sc}}(\mathsf{pp}, \mathsf{ek}, m; \mathsf{D}') \quad \text{or} \quad \mathsf{BGV.Enc}(\mathsf{pp}, \mathsf{ek}, M; \mathsf{D}')$$

for scalar or vector encryption. The semicolon emphasizes that the distribution is an additional sampling parameter rather than part of the plaintext input.

- $\mathsf{pp} \leftarrow\!\$\ \mathsf{BGV.Setup}(1^\lambda)$: The setup algorithm outputs $\mathsf{pp} = (\mathcal{R}_q, p, k, \ell, \mathsf{D})$, where $1 < p < q$, $\gcd(p, q) = 1$, $p \ll q$ is the plaintext modulus, and D is a discrete Gaussian over $\mathcal{R}_q$. Parameters are chosen such that $\mathsf{MLWE}_{k,\ell,\mathsf{D}}$ is hard.

- $(\mathsf{dk}, \mathsf{ek}) \leftarrow \mathsf{BGV.KGen}(\mathsf{pp})$: Sample a uniform $\mathbf{A} \leftarrow\!\$\ \mathcal{R}_q^{k \times k}$, a secret $\mathbf{s} \leftarrow\!\$\ \mathsf{D}^k$, and an error $\mathbf{e} \leftarrow\!\$\ \mathsf{D}^k$, and set $\mathbf{b} \coloneqq \mathbf{A}\mathbf{s} + p\mathbf{e} \in \mathcal{R}_q^k$. The encryption key is $\mathsf{ek} = (\mathbf{A}, \mathbf{b})$ and the decryption key is $\mathsf{dk} = \mathbf{s}$.

- $\mathsf{ctx} \leftarrow \mathsf{BGV.Enc_{sc}}(\mathsf{pp}, \mathsf{ek}, m)$: On input the encryption key $\mathsf{ek} = (\mathbf{A}, \mathbf{b})$ and a scalar message $m \in \mathcal{R}_p$, sample encryption randomness $\mathbf{r} \leftarrow\!\$\ \mathsf{D}^k$ and noise terms $\mathbf{e}_1 \leftarrow\!\$\ \mathsf{D}^k$ and $e_2 \leftarrow\!\$\ \mathsf{D}$, and compute
$$\mathbf{u} \coloneqq \mathbf{A}^T\mathbf{r} + p\mathbf{e}_1 \in \mathcal{R}_q^k, \qquad v \coloneqq \mathbf{b}^T\mathbf{r} + pe_2 + m \in \mathcal{R}_q.$$
It outputs $\mathsf{ctx} = (\mathbf{u}, v)$. The notation $\mathsf{ctx} \leftarrow \mathsf{BGV.Enc_{sc}}(\mathsf{pp}, \mathsf{ek}, m; \mathsf{D}')$ denotes the same algorithm with $\mathbf{r}$, $\mathbf{e}_1$, and e_2 sampled from D'.

- $\mathbf{ctx} \leftarrow \mathsf{BGV.Enc}(\mathsf{pp}, \mathsf{ek}, M)$: For a vector message $M = (M_1, \ldots, M_\ell) \in \mathcal{M}$, define the vector encryption algorithm by applying the scalar encryption algorithm coordinate-wise:
$$\mathbf{ctx} \coloneqq \left(\mathsf{ctx}^{(1)}, \ldots, \mathsf{ctx}^{(\ell)}\right), \qquad \mathsf{ctx}^{(j)} \leftarrow \mathsf{BGV.Enc_{sc}}(\mathsf{pp}, \mathsf{ek}, M_j) \quad \text{for } j \in [\ell].$$
Similarly, $\mathbf{ctx} \leftarrow \mathsf{BGV.Enc}(\mathsf{pp}, \mathsf{ek}, M; \mathsf{D}')$ denotes coordinate-wise encryption using $\mathsf{BGV.Enc_{sc}}(\mathsf{pp}, \mathsf{ek}, \cdot; \mathsf{D}')$. For $c \in \mathcal{R}_p$, write $\overline{c} \coloneqq (c, \ldots, c) \in \mathcal{M}$. Its encryption may be represented by the vector ciphertext $\mathsf{BGV.Enc}(\mathsf{pp}, \mathsf{ek}, \overline{c})$ or compactly by one scalar ciphertext $\mathsf{ctx}_c \leftarrow \mathsf{BGV.Enc_{sc}}(\mathsf{pp}, \mathsf{ek}, c)$, reused in every coordinate during evaluation.

- $\mathbf{ctx}' \leftarrow \mathsf{BGV.Eval}(\mathsf{pp}, \mathsf{ek}, f_{\boldsymbol{\alpha}}, (\mathbf{ctx}_C, \mathbf{ctx}_M))$: On input $f_{\boldsymbol{\alpha}} \in \mathcal{F}_{B_\alpha}$, parse
$$\mathbf{ctx}_C = (\mathsf{ctx}_C^{(1)}, \ldots, \mathsf{ctx}_C^{(\ell)}), \qquad \mathbf{ctx}_M = (\mathsf{ctx}_M^{(1)}, \ldots, \mathsf{ctx}_M^{(\ell)}),$$

where

$$\mathsf{ctx}_C^{(j)} = (\mathbf{u}_C^{(j)}, v_C^{(j)}), \qquad \mathsf{ctx}_M^{(j)} = (\mathbf{u}_M^{(j)}, v_M^{(j)}).$$

If the first input is the compact scalar ciphertext $\mathsf{ctx}_c = (\mathbf{u}_c, v_c)$ encrypting $c \in \mathcal{R}_p$, set $\mathsf{ctx}_C^{(j)} \coloneqq \mathsf{ctx}_c$ for every $j \in [\ell]$. Write $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_\ell) \in \mathcal{R}_p^\ell$.
The evaluation algorithm outputs,

$$\mathbf{ctx}' = \big(\mathsf{ctx}'^{(1)}, \ldots, \mathsf{ctx}'^{(\ell)}\big),$$

where, for every $j \in [\ell]$,

$$\mathsf{ctx}'^{(j)} = (\mathbf{u}'^{(j)}, v'^{(j)}) \coloneqq \big(\boldsymbol{\alpha}_j \mathbf{u}_C^{(j)} + \mathbf{u}_M^{(j)},\ \boldsymbol{\alpha}_j v_C^{(j)} + v_M^{(j)}\big) \bmod q.$$

– $\mathbf{ctx}^{\mathsf{fl}} \leftarrow \mathsf{BGV.Eval}^{\mathsf{fl}}(\mathsf{pp}, \mathsf{ek}, f_{\boldsymbol{\alpha}}, (\mathbf{ctx}_C, \mathbf{ctx}_M))$: Compute

$$\begin{aligned} \mathbf{ctx}^{\mathsf{ev}} &\leftarrow \mathsf{BGV.Eval}(\mathsf{pp}, \mathsf{ek}, f_{\boldsymbol{\alpha}}, (\mathbf{ctx}_C, \mathbf{ctx}_M)), \\ \mathbf{ctx}_0 &\leftarrow \mathsf{BGV.Enc}(\mathsf{pp}, \mathsf{ek}, 0^\ell; \mathsf{D}_{\mathsf{fl}}), \\ \mathbf{ctx}^{\mathsf{fl}} &\coloneqq \mathbf{ctx}^{\mathsf{ev}} + \mathbf{ctx}_0. \end{aligned}$$

The encryption of zero is sampled independently, and addition is coordinate-wise over $\mathcal{R}_q$. The flooding distribution D_{fl} is chosen so that the noise of $\mathbf{ctx}_0$ statistically dominates that already present in $\mathbf{ctx}^{\mathsf{ev}}$.

– $m \leftarrow \mathsf{BGV.Dec}_{\mathsf{sc}}(\mathsf{pp}, \mathsf{dk}, \mathsf{ctx})$: For $\mathsf{dk} = \mathbf{s}$ and $\mathsf{ctx} = (\mathbf{u}, v)$, output $m \coloneqq v - \mathbf{s}^T\mathbf{u} \bmod q \bmod p$.
– $M \leftarrow \mathsf{BGV.Dec}(\mathsf{pp}, \mathsf{dk}, \mathbf{ctx})$: For a vector ciphertext $\mathbf{ctx} = (\mathsf{ctx}^{(1)}, \ldots, \mathsf{ctx}^{(\ell)})$, output

$$M \coloneqq \big(\mathsf{BGV.Dec}_{\mathsf{sc}}(\mathsf{pp}, \mathsf{dk}, \mathsf{ctx}^{(1)}), \ldots, \mathsf{BGV.Dec}_{\mathsf{sc}}(\mathsf{pp}, \mathsf{dk}, \mathsf{ctx}^{(\ell)})\big) \in \mathcal{M}.$$

Definition 9 (Decryption residual). *For a scalar ciphertext* $\mathsf{ctx} = (\mathbf{u}, v)$ *encrypting* $m \in \mathcal{R}_p$*, we define its decryption residual as*

$$\delta(\mathsf{ctx}) \coloneqq v - \mathbf{s}^T\mathbf{u} = m + pz \pmod q,$$

where z *is the noise term induced by the encryption randomness. For a vector ciphertext* $\mathbf{ctx} = (\mathsf{ctx}^{(1)}, \ldots, \mathsf{ctx}^{(\ell)})$*, we define,*

$$\delta(\mathbf{ctx}) \coloneqq \big(\delta(\mathsf{ctx}^{(1)}), \ldots, \delta(\mathsf{ctx}^{(\ell)})\big).$$

Lemma 6 (Correctness). *Let* $\mathbf{ctx}'$ *be the evaluation of* $f_{\boldsymbol{\alpha}}$ *on ciphertexts* $\mathbf{ctx}_C$ *and* $\mathbf{ctx}_M$ *encrypting* $C, M \in \mathcal{M}$*, respectively. Suppose that, except with negligible probability, for every coordinate* $j \in [\ell]$*, the decryption residual of* $\mathsf{ctx}'^{(j)}$ *can be written as*

$$\delta(\mathsf{ctx}'^{(j)}) = \alpha_j C_j + M_j + pz_{\mathsf{ev}}^{(j)} \pmod q$$

with

$$\left\| z_{\mathsf{ev}}^{(j)} \right\|_\infty \le B. \tag{3}$$

If $pB < q/2$*, then* BGV *is correct with respect to the family* $f_{\boldsymbol{\alpha}} \in \mathcal{F}_{B_\alpha}$*. Namely,*

$$\mathsf{BGV.Dec}(\mathsf{pp}, \mathsf{dk}, \mathbf{ctx}') = f_{\boldsymbol{\alpha}}(C, M) = \boldsymbol{\alpha} \odot C + M.$$

The same statement holds for the flooded evaluation $\mathsf{BGV.Eval}^{\mathsf{fl}}$*, with* B *replaced by a bound* B_{fl} *on the total decryption noise after flooding.*

Proof. It suffices to prove correctness coordinate-wise. Fix $j \in [\ell]$ and write $\mathsf{ctx}_C^{(j)} = (\mathbf{u}_C^{(j)}, v_C^{(j)})$ and $\mathsf{ctx}_M^{(j)} = (\mathbf{u}_M^{(j)}, v_M^{(j)})$. By definition of the decryption residual, there exist noise terms $z_C^{(j)}$ and $z_M^{(j)}$ such that

$$\delta(\mathsf{ctx}_C^{(j)}) = v_C^{(j)} - \mathbf{s}^T\mathbf{u}_C^{(j)} = C_j + pz_C^{(j)} \pmod q$$

and

$$\delta(\mathsf{ctx}_M^{(j)}) = v_M^{(j)} - \mathbf{s}^T\mathbf{u}_M^{(j)} = M_j + pz_M^{(j)} \pmod q.$$

The evaluated ciphertext in coordinate j is

$$\mathsf{ctx}'^{(j)} = \big(\boldsymbol{\alpha}_j\mathbf{u}_C^{(j)} + \mathbf{u}_M^{(j)}, \boldsymbol{\alpha}_j v_C^{(j)} + v_M^{(j)}\big) \bmod q.$$

Therefore,

$$\begin{aligned}\delta(\mathsf{ctx}'^{(j)}) &= \boldsymbol{\alpha}_j v_C^{(j)} + v_M^{(j)} - \mathbf{s}^T\big(\boldsymbol{\alpha}_j\mathbf{u}_C^{(j)} + \mathbf{u}_M^{(j)}\big) \pmod q\\ &= \boldsymbol{\alpha}_j\big(v_C^{(j)} - \mathbf{s}^T\mathbf{u}_C^{(j)}\big) + \big(v_M^{(j)} - \mathbf{s}^T\mathbf{u}_M^{(j)}\big) \pmod q\\ &= \boldsymbol{\alpha}_j C_j + M_j + p\big(\boldsymbol{\alpha}_j z_C^{(j)} + z_M^{(j)}\big) \pmod q.\end{aligned}$$

Hence the evaluated noise is $z_{\mathsf{ev}}^{(j)} = \boldsymbol{\alpha}_j z_C^{(j)} + z_M^{(j)}$. By assumption, $\|z_{\mathsf{ev}}^{(j)}\|_\infty \le B$ and $pB < q/2$, so the noise does not wrap modulo q. Reducing the residual modulo p removes it and yields

$$\mathsf{BGV.Dec}_{\mathsf{sc}}(\mathsf{pp}, \mathsf{dk}, \mathsf{ctx}'^{(j)}) = \boldsymbol{\alpha}_j C_j + M_j \bmod p.$$

Applying this argument coordinate-wise gives $\mathsf{BGV.Dec}(\mathsf{pp}, \mathsf{dk}, \mathbf{ctx}') = \boldsymbol{\alpha} \odot C + M$. For flooded evaluation, $\mathbf{ctx}^{\mathsf{fl}} = \mathbf{ctx}^{\mathsf{ev}} + \mathbf{ctx}_0$, where $\mathbf{ctx}_0$ encrypts 0^ℓ. Adding $\mathsf{ctx}_0^{(j)}$ preserves the plaintext and only adds a multiple of p to the residual. Thus the same argument applies with B replaced by a bound B_{fl} on the total noise after flooding.

Lemma 7 (IND-CPA Security). *Assuming the decisional* $\mathsf{MLWE}_{k,\ell,\mathsf{D}}$ *problem is hard for the parameters generated by* Setup*, the scalar encryption scheme* $\mathsf{BGV.Enc}_{\mathsf{sc}}$ *is* IND-CPA *secure. Consequently, its coordinate-wise vector extension* BGV.Enc *is also* IND-CPA *secure.*

More precisely, for any PPT adversary $\mathcal{A}$, there exist PPT distinguishers $\mathcal{B}$ and $\mathcal{C}$ such that

$$\mathsf{Adv}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda) \;\le\; \mathsf{Adv}_{\mathcal{B}}^{\mathsf{MLWE}_{\mathrm{q,n,k_{ek},\ell_{ek},D}}}(\lambda) + \mathsf{Adv}_{\mathcal{C}}^{\mathsf{MLWE}_{\mathrm{q,n,\ell_{ek}+1,k_{ek},D}}}(\lambda).$$

Proof. We first argue security of the scalar encryption scheme.

The public key is $\mathsf{ek} = (\mathbf{A}, \mathbf{b})$, where $\mathbf{b} = \mathbf{As}+p\mathbf{e}$. To account for the factor p multiplying the error, consider an MLWE sample $(\mathbf{A}_0, \mathbf{A}_0\mathbf{s}+\mathbf{e})$ and multiply both components by p. Since $\gcd(p, q) = 1$, multiplication by p is a permutation of $\mathcal{R}_q$; hence $p\mathbf{A}_0$ is uniform, and the resulting sample is distributed as $(\mathbf{A}, \mathbf{As} + p\mathbf{e})$. The same transformation maps a uniform sample to a uniform sample. Therefore, by the decisional MLWE assumption, the distribution of $(\mathbf{A}, \mathbf{b})$ is computationally indistinguishable from $(\mathbf{A}, \mathbf{b}^*)$, where $\mathbf{b}^*$ is uniformly random in $\mathcal{R}_q^k$. Thus, replacing the public key component $\mathbf{b}$ by a uniform vector changes the adversary's advantage by at most $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{MLWE}_{\mathrm{q,n,k_{ek},\ell_{ek},D}}}(\lambda)$ for some PPT distinguisher $\mathcal{B}$.

Now consider a challenge ciphertext under a public key $(\mathbf{A}, \mathbf{b}^*)$ with uniform $\mathbf{b}^*$. An honest scalar encryption of m_b has the form

$$\mathbf{u}^* = \mathbf{A}^T\mathbf{r} + p\mathbf{e}_1, \qquad v^* = (\mathbf{b}^*)^T\mathbf{r} + pe_2 + m_b.$$

Applying the same transformation to the corresponding block MLWE sample shows, by the decisional MLWE assumption, that the pair $(\mathbf{u}^*, v^*)$ is computationally indistinguishable from a uniformly random element of $\mathcal{R}_q^k \times \mathcal{R}_q$. Since adding the plaintext m_b to a uniform ring element does not change its distribution, the resulting challenge ciphertext is independent of the challenge bit b. This transition costs at most $\mathsf{Adv}_{\mathcal{C}}^{\mathsf{MLWE}_{\mathrm{q,n,\ell_{ek}+1,k_{ek},D}}}(\lambda)$ for some PPT distinguisher $\mathcal{C}$.

In the final game, the challenge ciphertext is independent of b, and hence the adversary wins with probability exactly 1/2. Therefore,

$$\mathsf{Adv}^{\text{IND-CPA}}_{\mathcal{A}}(\lambda) \;\leq\; \mathsf{Adv}^{\mathsf{MLWE}_{\mathrm{q},\mathrm{n},\mathrm{k}_{\mathsf{ek}},\ell_{\mathsf{ek}},\mathrm{D}}}_{\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{MLWE}_{\mathrm{q},\mathrm{n},\ell_{\mathsf{ek}}+1,\mathrm{k}_{\mathsf{ek}},\mathrm{D}}}_{\mathcal{C}}(\lambda).$$

The vector encryption algorithm $\mathsf{BGV.Enc}$ encrypts each coordinate using an independent invocation of the scalar encryption algorithm. Security of the vector scheme follows from a standard hybrid argument over the ℓ coordinates.

Lemma 8 (Chosen-key circuit privacy via full-randomness flooding). *Fix any syntactically valid encryption key $\mathsf{ek} = (\mathbf{A}, \mathbf{b})$, not necessarily generated by $\mathsf{BGV.KGen}$. For integer polynomial representatives, write an encryption as the deterministic map*

$$\mathcal{E}_{\mathsf{ek}}(a;\theta) := \left(\mathbf{A}^T\mathbf{r} + p\mathbf{e}_1,\, \mathbf{b}^T\mathbf{r} + pe_2 + a\right) \bmod q, \qquad \theta = (\mathbf{r}, \mathbf{e}_1, e_2).$$

The entire randomness tuple, including $\mathbf{r}$, is part of θ. Let $Z \leftarrow \mathcal{D}_{\mathsf{fl}}$ be the joint distribution of all fresh flooding coins for the output vector. Assume that, for every admissible integer shift $\mathbf{a}$ of these coins,

$$\Delta_{\mathsf{TV}}(Z + \mathbf{a}, Z) \leq \varepsilon_{\mathsf{fl}}. \tag{4}$$

Here Δ_{TV} denotes statistical distance. The admissible shifts include the evaluated input coins and any plaintext-representation carries defined below.

For input coordinates $\mathsf{ctx}^{(j)}_C = \mathcal{E}_{\mathsf{ek}}(\widetilde{C}_j; \theta_{C,j})$ and $\mathsf{ctx}^{(j)}_M = \mathcal{E}_{\mathsf{ek}}(\widetilde{M}_j; \theta_{M,j})$, let $\widetilde{y}_j$ be the fixed centered lift modulo p of $y_j = \alpha_j C_j + M_j$. Fix integer lifts of α_j and define

$$t_j = \frac{\alpha_j\widetilde{C}_j + \widetilde{M}_j - \widetilde{y}_j}{p}, \qquad \mathbf{a}_j = \alpha_j\theta_{C,j} + \theta_{M,j} + (0,0,t_j).$$

All operations in these two expressions are over the integer polynomial ring. The quotient is integral coefficient-wise. Then the joint flooded evaluation

$$\mathsf{ctx}^{(j)}_{\mathsf{fl}} = \alpha_j\mathsf{ctx}^{(j)}_C + \mathsf{ctx}^{(j)}_M + \mathcal{E}_{\mathsf{ek}}(0; Z_j) \pmod q$$

is at statistical distance at most $\varepsilon_{\mathsf{fl}}$ from a fresh vector encryption of y using the same joint flooding distribution. The bound holds even with the key, input-encryption coins, function, and any decryption key revealed to the distinguisher.

In the protocol, the private offset M is encrypted directly with fresh flooding coins:

$$\mathsf{ctx}^{(j)}_{\mathsf{out}} = \alpha_j\mathsf{ctx}^{(j)}_C + \mathcal{E}_{\mathsf{ek}}(\widetilde{M}_j; Z_j).$$

The same conclusion holds with $\mathbf{a}_j = \alpha_j\theta_{C,j} + (0,0,t_j)$. Thus no separate encryption of zero is required when the offset encryption already supplies the flooding.

Proof. Linearity of the full ciphertext map, for any fixed ek, gives

$$\mathsf{ctx}^{(j)}_{\mathsf{fl}} = \mathcal{E}_{\mathsf{ek}}\left(\widetilde{y}_j; Z_j + \mathbf{a}_j\right).$$

Indeed, the first component contains $\mathbf{A}^T(\mathbf{r}_{Z,j} + \alpha_j\mathbf{r}_{C,j} + \mathbf{r}_{M,j})$ and the corresponding first error terms. The second component has the same combined randomness, and pt_j accounts exactly for the difference between the integer plaintext expression and its fixed lift. The offset-encryption variant follows by omitting $\theta_{M,j}$.

Apply this deterministic map jointly to the two distributions in Eq. (4). Statistical distance cannot increase under a deterministic map, giving

$$\Delta_{\mathsf{TV}}\left((\mathsf{ctx}^{(j)}_{\mathsf{fl}})_{j=1}^{\ell}, (\mathcal{E}_{\mathsf{ek}}(\widetilde{y}_j; Z_j))_{j=1}^{\ell}\right) \leq \varepsilon_{\mathsf{fl}}.$$

This is a statement about the complete ciphertext vector, not only its decryption residual. It holds pointwise for every fixed key and admissible input opening, so it also holds with arbitrary retained key-generation information and after averaging over adversarial key choices. Neither a valid secret key nor an extraction algorithm for the input coins is used.

The simulator is therefore $\mathsf{Sim}_{\mathsf{HE}}(\mathsf{pp}, \mathsf{ek}, y) = \mathsf{BGV.Enc}(\mathsf{pp}, \mathsf{ek}, y; \mathsf{D}_{\mathsf{fl}})$, where D_{fl} specifies the joint coin law $\mathcal{D}_{\mathsf{fl}}$. For the protocol variant, $\alpha_j = s_j$, $C_j = c$, and $M_j = r'_j$. The accepted relation L_1 bounds θ_c, and the honest signing-key bound controls multiplication by each s_j. Under the plaintext no-wrap condition, the integer response $cs_j + r'_j$ is already its centered lift modulo p, hence $t_j = 0$. This proves the required chosen-key privacy on admissible inputs. Decryption correctness is a separate claim requiring an honestly generated key and the decryption no-wrap bounds.

Parameter selection. The parameters are chosen so that the decryption noise remains below the modulus-reduction threshold after evaluation. This bound depends on the coefficients of the evaluated function, on the randomness and error distributions used to generate the input ciphertexts, and, for flooded evaluation, on the flooding distribution. Namely, if the input ciphertext noises are $z_C^{(j)}$ and $z_M^{(j)}$, then evaluating $f_{\boldsymbol{\alpha}}(C, M) = \boldsymbol{\alpha} \odot C + M$ produces noise $z_{\mathsf{ev}}^{(j)} = \alpha_j z_C^{(j)} + z_M^{(j)}$. By the definition of $\mathcal{F}_{B_\alpha}$ and the convolution bound for polynomial multiplication,

$$\|z_{\mathsf{ev}}^{(j)}\|_\infty \le B_\alpha \|z_C^{(j)}\|_\infty + \|z_M^{(j)}\|_\infty.$$

For flooded evaluation, adding an encryption of zero with noise $z_0^{(j)}$ produces total noise $z_{\mathsf{fl}}^{(j)} = \alpha_j z_C^{(j)} + z_M^{(j)} + z_0^{(j)}$. Accordingly, the modulus q is chosen so that, except with negligible probability,

$$p\|z_{\mathsf{ev}}^{(j)}\|_\infty < \frac{q}{2} \quad \text{and, in the flooded case,} \quad p\|z_{\mathsf{fl}}^{(j)}\|_\infty < \frac{q}{2}$$

for every $j \in [\ell]$, with the plaintext contribution also included when applying the centered decryption no-wrap condition.

A.3 Commitment Schemes

A commitment scheme is a triple of PPT algorithms $\mathsf{CS} := (\mathsf{Setup}, \mathsf{Com}, \mathsf{Verify})$ such that:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{ck}$. On input λ, outputs a commitment key ck defining the message space $\mathcal{M}$. The commitment key is part of the public parameters ck of the blind signature scheme, and we pass it explicitly to the algorithms below.
- $\mathsf{Com}(\mathsf{ck}, \mathsf{msg}) \to (C, \mathsf{aux})$. On input the commitment key ck and $\mathsf{msg} \in \mathcal{M}$, outputs a commitment C together with opening information aux.
- $\mathsf{Verify}(\mathsf{ck}, \mathsf{msg}, (C, \mathsf{aux})) \to b \in \{0, 1\}$. Outputs 1 if (C, aux) is a valid opening of msg, and 0 otherwise.

Correctness. For all $\lambda \in \mathbb{N}$, we have

$$\Pr\left[\mathsf{Verify}(\mathsf{ck}, \mathsf{msg}, (C, \mathsf{aux})) = 1 \;\middle|\; \begin{array}{l} \mathsf{ck} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda),\ \mathsf{msg} \leftarrow\!\$\ \mathcal{M}, \\ (C, \mathsf{aux}) \leftarrow\!\$\ \mathsf{Com}(\mathsf{ck}, \mathsf{msg}) \end{array}\right] = 1.$$

Hiding. The scheme is computationally hiding if for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$:

$$\Pr\left[b = b' \;\middle|\; \begin{array}{l} \mathsf{ck} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda), \\ (M_0, M_1, \mathsf{state}) \leftarrow\!\$\ \mathcal{A}_1(\mathsf{ck}), \\ b \leftarrow\!\$\ \{0, 1\}, \\ (C, \mathsf{aux}) \leftarrow\!\$\ \mathsf{Com}(\mathsf{ck}, M_b), \\ b' \leftarrow\!\$\ \mathcal{A}_2(C, \mathsf{state}) \end{array}\right] \le \tfrac{1}{2} + \mathsf{Adv}_{\mathsf{CS}}^{\mathsf{HID}}(\lambda).$$

Binding. The scheme is computationally binding if for all PPT adversaries $\mathcal{A}$:

$$\Pr\left[\begin{array}{l} M_1 \neq M_2, \\ \mathsf{Verify}(\mathsf{ck}, M_1, (C, \mathsf{aux}_1)) = 1, \\ \mathsf{Verify}(\mathsf{ck}, M_2, (C, \mathsf{aux}_2)) = 1 \end{array} \middle| \begin{array}{l} \mathsf{ck} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda), \\ (M_1, M_2, \mathsf{aux}_1, \mathsf{aux}_2, C) \leftarrow\!\$\ \mathcal{A}(\mathsf{ck}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Extractable commitment schemes. A commitment scheme is extractable if there exists an alternative setup algorithm $\mathsf{Setup}_{\mathsf{Ext}}$ that outputs $(\mathsf{ck}, \mathcal{T})$ such that:

- **Setup indistinguishability.** For any PPT adversary $\mathcal{D}$, there exists a PPT algorithm $\mathcal{B}$ such that

$$\left|\Pr[\mathcal{D}(\mathsf{ck}) = 1 \mid \mathsf{ck} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda)] - \Pr[\mathcal{D}(\mathsf{ck}) = 1 \mid (\mathsf{ck}, \mathcal{T}) \leftarrow\!\$\ \mathsf{Setup}_{\mathsf{Ext}}(1^\lambda)]\right| \leq \mathsf{Adv}^{\mathsf{SetupInd}}_{\mathsf{CS},\mathcal{B}}(\lambda).$$

- **Extractability.** There exists a PPT extractor Ext such that for any PPT adversary $\mathcal{A}$:

$$\Pr\left[\begin{array}{l} \mathsf{Verify}(\mathsf{ck}, m, (C, \mathsf{aux})) = 1, \\ m \neq \mathsf{Ext}(\mathcal{T}, C) \end{array} \middle| \begin{array}{l} (\mathsf{ck}, \mathcal{T}) \leftarrow\!\$\ \mathsf{Setup}_{\mathsf{Ext}}(1^\lambda), \\ (C, m, \mathsf{aux}) \leftarrow\!\$\ \mathcal{A}(\mathsf{ck}) \end{array}\right] \leq \mathsf{Adv}^{\mathsf{Ext}}_{\mathsf{CS},\mathcal{A}}(\lambda).$$

Extraction recovers the committed payload m, not necessarily the opening randomness aux. This definition concerns valid openings output by an adversary; it does not require arbitrary malformed commitments to admit an opening. The OMUF proof uses the stronger uniform decoding property established algebraically for our bounded BDLOP instantiation in Section 4.

A.4 BDLOP Commitments

We now describe the BDLOP commitment scheme of Baum *et al.* [BDL+18] following the above formalism.

- $\mathsf{BDLOP.Setup}(1^\lambda) \to \mathsf{ck}$. Sample parameters (k, n_r, ℓ, q) as a function of the security parameter λ. Define a distribution D over $\mathcal{R}_q$. Sample $(\mathbf{A}, \mathbf{B}) \leftarrow\!\$\ \mathcal{R}_q^{k \times n_r} \times \mathcal{R}_q^{\ell \times n_r}$ and output

$$\mathsf{ck} = (k, n_r, \ell, q, \mathsf{D}, \mathbf{A}, \mathbf{B}).$$

- $\mathsf{BDLOP.Com}(\mathsf{ck}, (\mathbf{m}, \mathbf{r})) \to (\mathbf{t}_A, \mathbf{t}_B)$. On input a message vector $\mathbf{m} \in \mathcal{R}_q^\ell$, randomness $\mathbf{r} \leftarrow\!\$\ \mathsf{D}^{n_r}$, and the commitment key ck, output the commitment $(\mathbf{t}_A, \mathbf{t}_B) \in \mathcal{R}_q^k \times \mathcal{R}_q^\ell$ defined by

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{r} + \begin{bmatrix} 0 \\ \mathbf{m} \end{bmatrix} \bmod q.$$

- $\mathsf{BDLOP.Verify}(\mathsf{ck}, (\mathbf{m}, \mathbf{r}), (\mathbf{t}_A, \mathbf{t}_B)) \to b \in \{0, 1\}$. Check that

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{r} + \begin{bmatrix} 0 \\ \mathbf{m} \end{bmatrix} \bmod q,$$

  and that $\mathbf{r}$ is small. Output 1 if all conditions hold, and 0 otherwise.

Lemma 9 (Hiding and Binding of BDLOP). *Let* BDLOP *be the commitment scheme defined above. Then, for parameters chosen according to* λ*:*

- BDLOP *is computationally binding under the* $\mathsf{MSIS}_{k+\ell,n_r,B}$ *assumption.*
- BDLOP *is computationally hiding under the* $\mathsf{MLWE}_{n_r-(k+\ell),k+\ell,\mathsf{D}}$ *assumption.*

A.5 Extractable BDLOP Commitment

The BDLOP commitment scheme allows committing to a message $\mathbf{m} \in \mathcal{R}_q^{\ell}$ using a short randomness $\mathbf{r}$, together with public random matrices $\mathbf{A} \in \mathcal{R}_q^{k \times n_r}$ and $\mathbf{B} \in \mathcal{R}_q^{\ell \times n_r}$, as

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{r} + \begin{bmatrix} 0 \\ \mathbf{m} \end{bmatrix} \bmod q.$$

The binding property follows from the MSIS assumption on the matrix $\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$, as finding two distinct openings yields a short non-zero vector in its kernel. The hiding property follows from the MLWE assumption, provided that $\mathbf{r}$ is sampled from a distribution such that

$$\left(\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}, \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{r} \right)$$

is computationally indistinguishable from uniform.

Extractability via simulated setup. For the purpose of the security proof of our blind signature, we rely on a simulated setup for the BDLOP commitment scheme that enables extraction. Concretely, in the reduction, the challenger generates the public matrices $(\mathbf{A}, \mathbf{B})$ as follows. It samples $\mathbf{A} \leftarrow\!\!\$\, \mathcal{R}_q^{k \times n_r}$ together with short matrices $\mathbf{S} \leftarrow \mathsf{D}_{\mathbf{B}}^{\ell \times k}$ and $\mathbf{E} \leftarrow \mathsf{D}_{\mathbf{B}}^{\ell \times n_r}$ and sets

$$\mathbf{B} := \mathbf{S}\mathbf{A} + p\mathbf{E}.$$

The commitment algorithm itself remains unchanged, namely

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{r} + \begin{bmatrix} 0 \\ \mathbf{m} \end{bmatrix} \bmod q.$$

Using the relation $\mathbf{B} = \mathbf{S}\mathbf{A} + p\mathbf{E}$, we obtain

$$\mathbf{t}_B = \mathbf{S}\mathbf{t}_A + p\mathbf{E}\mathbf{r} + \mathbf{m} \bmod q.$$

Therefore, by reducing modulo p, the error term vanishes and the reduction can recover the committed message as

$$\mathbf{m} = \mathbf{t}_B - \mathbf{S}\mathbf{t}_A \bmod p.$$

Indistinguishability of the setup. We argue that the above simulated setup is computationally indistinguishable from the real one under the MLWE assumption. In particular, any distinguisher between $(\mathbf{A}, \mathbf{B})$ sampled uniformly at random and $(\mathbf{A}, \mathbf{S}\mathbf{A} + p\mathbf{E})$ for short $\mathbf{S}, \mathbf{E}$ can be used to construct an adversary against the MLWE problem. Therefore, this modification is indistinguishable to the adversary.

Remark 1 (Random-oracle-based setup). The simulated setup above is only used in the security reduction. In the real protocol, no trusted party is needed to generate the public matrices $(\mathbf{A}, \mathbf{B})$. Instead, these matrices can be derived from public seeds by using a hash function, modeled as a random oracle, to expand the seeds into pseudorandom elements of $\mathcal{R}_q$. Thus, all users can locally derive the same commitment parameters. The indistinguishability argument above ensures that replacing uniformly sampled matrices by programming the random oracle with matrices generated in the way above does not affect the adversary's view.

A.6 Non-Interactive Zero-Knowledge Proofs

We formally define NIZK and consider only transparent NIZK schemes without a trusted setup. Our definition is adapted from [BFM+25].
A non-interactive zero-knowledge proof system for a family of languages $\mathcal{L}$, which we denote by NIZK, is defined by the following algorithms:

- $\mathsf{pp} \leftarrow \mathsf{NIZK}_{\mathsf{L}}.\mathsf{Setup}(1^\lambda)$. Given a security parameter λ, outputs public parameters pp.
- $\pi \leftarrow \mathsf{NIZK}_{\mathsf{L}}.\mathsf{Prove}(\mathsf{pp}, w, x)$. Given a witness w, a language $\mathsf{L} \in \mathcal{L}$ and a statement x such that $(x, w) \in \mathsf{L}$, outputs a proof π.
- $1/0 \leftarrow \mathsf{NIZK}_{\mathsf{L}}.\mathsf{Verify}(\mathsf{pp}, \pi, x)$. Given a proof π, a language $\mathsf{L} \in \mathcal{L}$ and a statement x, outputs 1 ("accept") or 0 ("reject").

The algorithm Prove is probabilistic, while the algorithm Verify is deterministic. The proof system satisfies the following properties:

- **Completeness.** For any language $\mathsf{L} \in \mathcal{L}$ and any statement-witness pair $(x, w) \in \mathsf{L}$,

$$\Pr[\mathsf{NIZK}_{\mathsf{L}}.\mathsf{Verify}(\mathsf{pp}, \pi, x) = 1 \mid \pi \leftarrow \mathsf{NIZK}_{\mathsf{L}}.\mathsf{Prove}(\mathsf{pp}, w, x)] = 1.$$

- **ε-Zero-Knowledge.** There exists a probabilistic polynomial-time simulator Sim such that for any probabilistic polynomial-time distinguisher $\mathcal{A}$,

$$|\Pr[\mathcal{A}(\mathsf{pp}, \pi) = 1 : \pi \leftarrow \mathsf{NIZK}_{\mathsf{L}}.\mathsf{Prove}(\mathsf{pp}, w, x)] - \Pr[\mathcal{A}(\mathsf{pp}, \pi) = 1 : \pi \leftarrow \mathsf{Sim}(\mathsf{pp}, \mathsf{L}, x)]| \leq \varepsilon.$$

- **ε-Knowledge Soundness.** There exists a probabilistic polynomial-time extractor Ext such that for any probabilistic polynomial-time prover Prove^* and any language $\mathsf{L} \in \mathcal{L}$, if

$$\Pr[\mathsf{NIZK}_{\mathsf{L}}.\mathsf{Verify}(\mathsf{pp}, \pi, x) = 1 : \pi \leftarrow \mathsf{Prove}^*(\mathsf{pp}, \mathsf{L}, x)] \geq \varepsilon + \eta,$$

for some non-negligible function η, then

$$\Pr[(x, w) \in \mathsf{L} : w \leftarrow \mathsf{Ext}(\mathsf{pp}, \mathsf{Prove}^*, \mathsf{L}, x)]$$

is non-negligible.

B Deferred Lemmas

Definition 10. *For any $\mathbf{A} \in \mathcal{R}_q^{k \times \ell}$, let $\mathsf{D}^{\mathsf{bd-MLWE}}_{\ell,k,\mu,\nu}(\mathbf{A})$ be the distribution defined as $\{\lfloor \mathbf{As}+\mathbf{e} \rceil_\nu | (\mathbf{s}, \mathbf{e}) \leftarrow \mathsf{D}_\mu^\ell \times \mathsf{D}_\mu^k\}$, that is, it samples from the $\mathsf{MLWE}_{\ell,k,\mu}$ distribution with ν bits dropped.*

Lemma 10 (cf. [DKM+24]). *For any $\mu > 2n \cdot q^{\frac{1}{k+\ell} + \frac{2}{n\ell}}$ and $\nu < \log(q) - 2$, we have,*

$$\Pr\left[H_\infty(\mathsf{D}^{\mathsf{bd-MLWE}}_{\ell,k,\mu,\nu}(\mathbf{A})) \geq n - 1\right] > 1 - 2^{-n+1}$$

Lemma 11 (Extension of Lemma 3.2 [DKM+24]). *Let $\nu \in \mathbb{Z}_{\geq 0}$ and $q \in \mathbb{Z}_{>0}$ with $q > 2$, and set*

$$q_\nu = \left\lfloor \frac{q}{2^\nu} \right\rceil.$$

Moreover, assume that q and ν satisfy

$$q_\nu = \lfloor \frac{q}{2^\nu} \rceil,$$

that is, q can be decomposed as

$$q = 2^\nu q_\nu + q_\perp, \qquad q_\perp \in \left[0,\, 2^{\max(\nu-1,0)} - 1\right].$$

Then, for any $x \in \mathbb{Z}_q$, we have

$$\left| x - 2^\nu \cdot \overline{\lfloor x \rceil_\nu} \right| \le 2^\nu - 1.$$

Moreover, for all $\mathbf{x}, \boldsymbol{\delta} \in \mathbb{Z}_q^n$, we have

$$\left\| 2^\nu \cdot \left(\overline{\lfloor \mathbf{x} + \boldsymbol{\delta} \rceil_\nu} - \overline{\lfloor \mathbf{x} \rceil_\nu} \right) \bmod q \right\| \le \left\| 2^\nu \cdot \overline{\lfloor \boldsymbol{\delta} \rceil_\nu} \bmod q \right\| + \|\mathbf{1}\| \cdot f(\nu),$$

where

$$f(\nu) = \begin{cases} 0 & \text{if } \nu = 0, \\ 3 & \text{if } \nu = 1, \\ 6 & \text{if } \nu = 2, \\ 12 & \text{if } \nu = 3, \\ 2^\nu & \text{if } \nu \ge 4. \end{cases}$$

Proof. For $\nu \ge 1$, the statement follows directly from [DKM+24, Lemma 3.2], since in this case our definition of $f(\nu)$ coincides with theirs.

It remains to consider the case $\nu = 0$. In this case,

$$q_0 = \left\lfloor \frac{q}{2^0} \right\rceil = q,$$

and hence modulus rounding with parameter $\nu = 0$ is the identity map. In particular, for every $x \in \mathbb{Z}_q$,

$$\lfloor x \rceil_0 = x.$$

Therefore,

$$\left| x - 2^0 \cdot \overline{\lfloor x \rceil_0} \right| = 0 = 2^0 - 1,$$

which proves the first inequality.

Similarly, for all $\mathbf{x}, \boldsymbol{\delta} \in \mathbb{Z}_q^n$,

$$\lfloor \mathbf{x} + \boldsymbol{\delta} \rceil_0 - \lfloor \mathbf{x} \rceil_0 = \boldsymbol{\delta} \pmod q.$$

Since also $\lfloor \boldsymbol{\delta} \rceil_0 = \boldsymbol{\delta}$ and $f(0) = 0$, we obtain

$$\begin{aligned} \left\| 2^0 \cdot \overline{\lfloor \mathbf{x} + \boldsymbol{\delta} \rceil_0 - \lfloor \mathbf{x} \rceil_0} \bmod q \right\| &= \left\| \overline{\boldsymbol{\delta}} \bmod q \right\| \\ &= \left\| 2^0 \cdot \overline{\lfloor \boldsymbol{\delta} \rceil_0} \bmod q \right\| \\ &\le \left\| 2^0 \cdot \overline{\lfloor \boldsymbol{\delta} \rceil_0} \bmod q \right\| + \|\mathbf{1}\| \cdot f(0). \end{aligned}$$

Thus, the statement also holds for $\nu = 0$.

B.1 Technical Lemmas for the BRaccoon OMUF Reduction

Proof of Lemma 4

Proof. The non-rounded verification equation holds immediately from the definition of $\mathbf{h}_0$, since

$$\mathbf{A}\mathbf{z} - c\mathbf{t}_0 + \mathbf{h}_0 = \iota(\mathbf{w}) \pmod q.$$

It remains to bound the norm of the converted hint. Let

$$\delta_{\mathbf{t}} = 2^{\nu_{\mathbf{t}}}\overline{\mathbf{t}} - \mathbf{t}_0 \pmod q, \qquad a = \mathbf{A}\mathbf{z} - 2^{\nu_{\mathbf{t}}}c\overline{\mathbf{t}}.$$

By Lemma 11, we may choose representatives satisfying

$$\|\delta_{\mathbf{t}}\|_\infty \le 2^{\nu_{\mathbf{t}}} - 1, \qquad \delta_{\mathbf{w}} = \iota(\lfloor a \rceil_{\nu_{\mathbf{w}}}) - a \pmod q, \qquad \|\delta_{\mathbf{w}}\|_\infty \le 2^{\nu_{\mathbf{w}}} - 1.$$

Writing $\overline{\mathbf{h}}$ for the canonical representative of $\mathbf{h}$, there exists a coefficient vector d with entries in $\{0,1\}$ such that

$$\overline{\mathbf{w}} = \overline{\lfloor a \rceil_{\nu_{\mathbf{w}}}} + \overline{\mathbf{h}} - q_{\mathbf{w}} d.$$

Since $2^{\nu_{\mathbf{w}}} q_{\mathbf{w}} = q - q_{\perp,\mathbf{w}}$, it follows that

$$\mathbf{h}_0 = 2^{\nu_{\mathbf{w}}}\overline{\mathbf{h}} - c\delta_{\mathbf{t}} + \delta_{\mathbf{w}} + q_{\perp,\mathbf{w}} d \pmod q.$$

Using $\|c\|_1 = \omega$ and the triangle inequality therefore gives

$$\|(\mathbf{z}, \mathbf{h}_0)\|_2 \le \left\|(\mathbf{z}, 2^{\nu_{\mathbf{w}}}\overline{\mathbf{h}})\right\|_2 + \Delta_{\mathsf{rnd}} \le B_2^\star + \Delta_{\mathsf{rnd}},$$

as claimed.

Proof of Lemma 5

Proof. The proof of Lemma 3.8 of [DKM+24] (Lemma 10) first establishes, except with probability at most 2^{-n+1} over the choice of $\mathbf{A}$, the required regularity of the unrounded distribution

$$\mathbf{A}\mathbf{r}' + \mathbf{e}', \qquad (\mathbf{r}', \mathbf{e}') \leftarrow \mathsf{D}^{\ell}_{\mathbf{w}'} \times \mathsf{D}^{k}_{\mathbf{w}'}.$$

Importantly, this regularity event is independent of the rounding parameter.

Fix a matrix $\mathbf{A}$ for which this event holds, and consider an arbitrary signing session. Condition on the complete protocol history immediately before $(\mathbf{r}', \mathbf{e}')$ is sampled. Since $(\mathbf{r}', \mathbf{e}')$ is sampled freshly, its distribution is unchanged under this conditioning. Moreover, the shift $(\mathbf{r}, \mathbf{e})$ is already fixed, and hence

$$\Delta := \mathbf{A}\mathbf{r} + \mathbf{e}$$

is fixed.

Therefore

$$\mathbf{w}_{\mathsf{raw}} = (\mathbf{A}\mathbf{r}' + \mathbf{e}') + \Delta.$$

The regularity statement of [LPR13, Cor. 7.5] used in that proof bounds the statistical distance between $\langle \mathbf{a}, \mathbf{r}' \rangle + e'$ and the uniform distribution on $\mathcal{R}_q$, where $\mathbf{a}$ is the first row of $\mathbf{A}$. Translation by a fixed element is a bijection of $\mathcal{R}_q$, so the shifted variable satisfies the same bound, and the remaining steps of that proof (data processing, then the counting bound on the rounding map) apply verbatim to it. With $\nu = 0$ this gives

$$H_\infty(\mathbf{w}_{\mathsf{raw}} \mid \mathsf{history}) \ge n - 1.$$

Likewise,

$$\mathbf{w} = \lfloor \mathbf{A}\mathbf{r}' + \mathbf{e}' + \Delta \rceil_{\nu_{\mathbf{w}}},$$

where the shift is added *before* rounding, as in the protocol. The counting step of Lemma 3.8 bounds the largest probability of a rounded uniform element by $(2^{\nu_{\mathbf{w}}} + 1)/q$, which is unaffected by the translation, so the same argument with rounding parameter $\nu_{\mathbf{w}}$ gives

$$H_\infty(\mathbf{w} \mid \mathsf{history}) \ge n - 1.$$

Both conclusions rely on the same regularity event for $\mathbf{A}$.

C Security Proofs for the Non-Rounded Schemes

C.1 Security of Non-Rounded Gaussian Raccoon

Proof. For an honestly generated signature,

$$\begin{aligned}\mathbf{Az} - c\mathbf{t} &= \mathbf{A}(c\mathbf{s} + \mathbf{r}') - c(\mathbf{As} + \mathbf{e}_{\mathsf{kg}}) = \mathbf{Ar}' - c\mathbf{e}_{\mathsf{kg}},\\ \mathbf{h} &= \mathbf{w} - (\mathbf{Az} - c\mathbf{t}) = c\mathbf{e}_{\mathsf{kg}} + \mathbf{e}'.\end{aligned}$$

The verifier therefore reconstructs $\widetilde{\mathbf{w}} = \mathbf{Az} - c\mathbf{t} + \mathbf{h} = \mathbf{Ar}' + \mathbf{e}' = \mathbf{w}$, so the challenge-verification equation holds. Moreover, $(\mathbf{z}, \mathbf{h}) = c(\mathbf{s}, \mathbf{e}_{\mathsf{kg}}) + (\mathbf{r}', \mathbf{e}')$. Using the discrete-Gaussian tail bound and $\|c\|_1 = \omega$, we obtain, with overwhelming probability,

$$\|(\mathbf{z}, \mathbf{h})\|_2 \le e^{1/4}\big(\omega\sigma_{\mathbf{t}} + \sigma_{\mathbf{w}'}\big)\sqrt{n(k+\ell)} = B^{(0)}_{2,\mathsf{corr}} \le B^{(0)}_2.$$

This proves correctness.

For unforgeability, we adapt the hybrids of [DKM+24, Appendix C.3]. The random-oracle programming step is unchanged. In the absence of modulus rounding, the commitment programmed by the signing oracle is distributed as $\mathbf{Ar}' + \mathbf{e}'$. The proof of the min-entropy bound used in Hybrid 2 extends to rounding parameter zero, since the corresponding rounding map is the identity map.

The transition replacing $\mathbf{t} = \mathbf{As} + \mathbf{e}_{\mathsf{kg}}$ by a uniformly random vector is simulated using the Hint-MLWE assumption. For each signing query, the corresponding Hint-MLWE tuple is

$$(c_i, \mathbf{z}_i, \mathbf{z}'_i) = \big(c_i,\ c_i\mathbf{s} + \mathbf{r}'_i,\ c_i\mathbf{e}_{\mathsf{kg}} + \mathbf{e}'_i\big).$$

Thus, the real Hint-MLWE distribution corresponds to $\mathbf{t} = \mathbf{As} + \mathbf{e}_{\mathsf{kg}}$, whereas the random distribution replaces $\mathbf{t}$ by a uniformly random vector while preserving the same hint tuples.

For the i-th signing query, the signing oracle sets $\mathbf{w}_i := \mathbf{Az}_i - c_i\mathbf{t} + \mathbf{z}'_i$ and $\mathbf{h}_i := \mathbf{z}'_i$, and programs $H_c(\mu_i, \mathbf{w}_i) := c_i$. When $\mathbf{t} = \mathbf{As} + \mathbf{e}_{\mathsf{kg}}$, we have

$$\begin{aligned}\mathbf{w}_i &= \mathbf{A}(c_i\mathbf{s} + \mathbf{r}'_i) - c_i(\mathbf{As} + \mathbf{e}_{\mathsf{kg}}) + c_i\mathbf{e}_{\mathsf{kg}} + \mathbf{e}'_i\\ &= \mathbf{Ar}'_i + \mathbf{e}'_i,\end{aligned}$$

so the signing oracle is perfectly simulated.

It remains to reduce the final hybrid to SelfTargetMSIS. In this hybrid, $\mathbf{t}$ is uniformly random and independent of $\mathbf{A}$. Hence $M := [-\mathbf{t} \mid \mathbf{A}] \in \mathcal{R}_q^{k\times(\ell+1)}$ is itself uniformly distributed. From a valid forgery $(c^*, \mathbf{z}^*, \mathbf{h}^*)$ on a fresh digest μ^*, the reduction outputs

$$\mathbf{z}_{\mathsf{sol}} := \begin{pmatrix} c^* \\ \mathbf{z}^* \\ \mathbf{h}^* \end{pmatrix}.$$

The verification equation gives

$$\mathbf{Az}^* - c^*\mathbf{t} + \mathbf{h}^* = [-\mathbf{t} \mid \mathbf{A} \mid I_k]\begin{pmatrix} c^* \\ \mathbf{z}^* \\ \mathbf{h}^* \end{pmatrix} = [M \mid I_k]\mathbf{z}_{\mathsf{sol}},$$

so the Fiat–Shamir equation of the valid forgery is exactly the SelfTargetMSIS relation.

Finally, since $\|c^*\|_2 = \sqrt{\omega}$ and $\|(\mathbf{z}^*, \mathbf{h}^*)\|_2 \le B^{(0)}_2$,

$$\begin{aligned}\|\mathbf{z}_{\mathsf{sol}}\|_2 &\le \sqrt{\omega + \big(B^{(0)}_2\big)^2}\\ &\le B^{(0)}_2 + \sqrt{\omega} = B^{(0)}_{\mathsf{MSIS}}.\end{aligned}$$

Since $\|c^*\|_2 = \sqrt{\omega} > 0$, we also have $\mathbf{z}_{\mathsf{sol}} \neq 0$. Thus $\mathbf{z}_{\mathsf{sol}}$ is a valid SelfTargetMSIS solution.

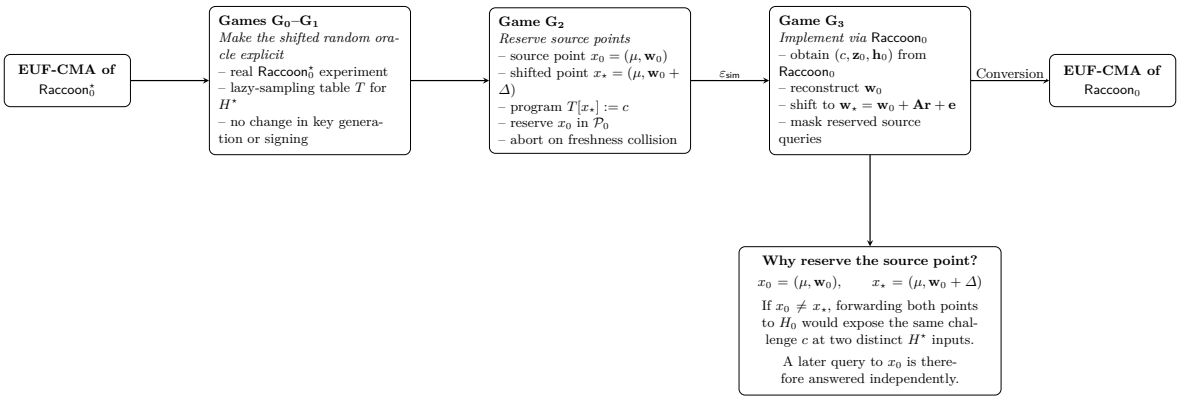


Fig. 8. Proof structure of the reduction from EUF-CMA security of $\mathsf{Raccoon}_0^\star$ to EUF-CMA security of $\mathsf{Raccoon}_0$. The main simulation loss $\varepsilon_{\mathsf{sim}}$ comes from the freshness tests and possible aborts when source or shifted challenge points collide with previously used points. The reservation set $\mathcal{P}_0$ ensures that distinct source and shifted points receive independent answers in the simulated shifted random oracle.

Accounting. The bound of Theorem 2 collects the losses of the three transitions. Programming $\mathsf{H}_c(\mu_i, \mathbf{w}_i) := c_i$ fails only if $(\mu_i, \mathbf{w}_i)$ was queried before, by the adversary or by an earlier signing query. By Lemma 10 with $\nu = 0$, except with probability 2^{-n+1} over $\mathbf{A}$, every $\mathbf{w}_i = \mathbf{Ar}'_i + \mathbf{e}'_i$ has min-entropy at least $n - 1$ given the history, so a union bound over the Q_H adversarial queries and the earlier programmed points gives $(Q_s Q_H + Q_s(Q_s - 1)/2)\, 2^{-(n-1)}$. The Hint-MLWE transition costs $\mathsf{Adv}^{\mathsf{H\text{-}MLWE}}_{\mathcal{B}}(\lambda)$, and the forgery yields a $\mathsf{SelfTargetMSIS}$ solution unless the digest μ^* of the forged message collides with the digest of a signed message, which is the event bounded by $\varepsilon_{\mathsf{hash}}$.

C.2 Reduction from $\mathsf{Raccoon}_0^\star$ to $\mathsf{Raccoon}_0$

We prove Theorem 4 under its stated correctness and entropy premises and with the common verification bound B.

Proof. We proceed by a series of hybrid games.

$\mathbf{G_0}$ This is the real EUF-CMA experiment for $\mathsf{Raccoon}_0^\star$. The challenger answers admissible signing queries $(\mu, \mathbf{r}, \mathbf{e})$ and maintains a lazy-sampling table for $H^\star$.

$\mathbf{G_1}$ We make the lazy-sampling implementation explicit. On a query x, return $T[x]$ if it is defined; otherwise sample a uniform challenge, store it as $T[x]$, and return it. Signing uses the same table. Neither key generation nor signing is modified, and therefore $\Pr[\mathsf{S}_1] = \Pr[\mathsf{S}_0]$.

$\mathbf{G_2}$ Alongside the public-oracle table T, we maintain an initially empty set $\mathcal{P}_0$ of reserved source points. A reserved point need not have an answer yet: on its first public query, its answer is sampled uniformly and stored in T, exactly as in $\mathbf{G}_1$.

For a signing query $(\mu, \mathbf{r}, \mathbf{e})$, let $\Delta := \mathbf{Ar} + \mathbf{e}$. Generate an ordinary signing sample $(\mathbf{r}', \mathbf{e}') \leftarrow \mathsf{D}^{\ell}_{\mathbf{w}'} \times \mathsf{D}^{k}_{\mathbf{w}'}$ as in $\mathsf{Raccoon}_0$, which has no rejection step; by the tail bound of Lemma 3 and $B \geq B^{(0)}_{2,\mathsf{corr}} + B_{\mathbf{w}}$, the resulting shifted signature passes the norm test of Figure 4 except with the probability $\varepsilon_{\mathsf{corr}}$ of the theorem statement, and we henceforth drop the restart step of the signing algorithm, which changes the adversary's view by at most $\varepsilon_{\mathsf{corr}}$. Set

$$\mathbf{w}_0 := \mathbf{Ar}' + \mathbf{e}', \qquad x_0 := (\mu, \mathbf{w}_0), \qquad x_\star := (\mu, \mathbf{w}_0 + \Delta).$$

If either x_0 or $x_\star$ belongs to $\mathrm{dom}(T) \cup \mathcal{P}_0$, the game aborts. Otherwise, sample $c \leftarrow\!\!\$\ \mathcal{C}$ and set $T[x_\star] := c$. If $x_0 \neq x_\star$, add x_0 to $\mathcal{P}_0$ without assigning it an answer.

Finally, return

$$(c, \mathbf{z}_\star, \mathbf{h}_\star) := \big(c,\ c\mathbf{s} + \mathbf{r}' + \mathbf{r},\ c\mathbf{e}_{\mathsf{kg}} + \mathbf{e}' + \mathbf{e}\big).$$

Reserving a fresh source point does not alter the public random-oracle distribution. If that point is subsequently queried, its answer is sampled independently of the challenge assigned to the distinct shifted point. For a fixed admissible shift, translation by Δ preserves the min-entropy of $\mathbf{w}_0$. Before each signing query, $\mathrm{dom}(T) \cup \mathcal{P}_0$ contains at most $Q_H + 2Q_s$ signing-related or public points. Thus, by the entropy premise and a union bound, the probability that a collision test causes an abort is bounded by $2Q_s(Q_H + 2Q_s)2^{-\kappa_w} + Q_s\varepsilon_{\mathsf{ent}}$. Conditioned on no abort, the distribution is identical to that in the preceding game, so $|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]| \le \varepsilon_{\mathsf{sim}}$ with $\varepsilon_{\mathsf{sim}}$ as in the theorem statement.

$\mathbf{G_3}$ We implement the preceding game using the non-rounded verification key vk_0 and access to the ordinary signing oracle and its external challenge oracle H_0. Here H_0 belongs to the external $\mathsf{Raccoon}_0$ challenger, whereas $H^\star$ is the oracle presented by the reduction to $\mathcal{A}$.

A public query x to $H^\star$ is answered as follows:

1. If $T[x]$ is defined, return it.
2. Otherwise, if $x \in \mathcal{P}_0$, sample an independent uniform value $d \in \mathcal{C}$, set $T[x] := d$, and return d without querying H_0.
3. Otherwise, query $H_0(x)$, store its answer in $T[x]$, and return it.

The order of these cases ensures that repeated queries always receive the same answer.

For an admissible signing query $(\mu, \mathbf{r}, \mathbf{e})$, the reduction obtains an auxiliary signature $(c, \mathbf{z}_0, \mathbf{h}_0)$ from the $\mathsf{Raccoon}_0$ signing oracle on μ and reconstructs

$$\mathbf{w}_0 := \mathbf{A}\mathbf{z}_0 - c\mathbf{t}_0 + \mathbf{h}_0, \qquad \mathbf{w}_\star := \mathbf{w}_0 + \mathbf{A}\mathbf{r} + \mathbf{e}.$$

If either $(\mu, \mathbf{w}_0)$ or $(\mu, \mathbf{w}_\star)$ belongs to $\mathrm{dom}(T) \cup \mathcal{P}_0$, the reduction aborts, exactly as in $\mathbf{G}_2$; in particular, it never makes more than one $\mathsf{Raccoon}_0$ signing query per $\mathsf{Raccoon}_0^\star$ signing query. Otherwise, set

$$T[(\mu, \mathbf{w}_\star)] := c.$$

If $\mathbf{w}_0 \neq \mathbf{w}_\star$, add $(\mu, \mathbf{w}_0)$ to $\mathcal{P}_0$. Finally, return

$$(c, \mathbf{z}_\star, \mathbf{h}_\star) := (c, \mathbf{z}_0 + \mathbf{r}, \mathbf{h}_0 + \mathbf{e}).$$

To compare $\mathbf{G}_2$ and $\mathbf{G}_3$, couple the ordinary signing samples used in both games. Conditioned on the freshness tests succeeding, the source point $(\mu, \mathbf{w}_0)$ has not previously been used by a public query or an earlier signing attempt. Hence $c = H_0(\mu, \mathbf{w}_0)$ is fresh and uniform. Moreover,

$$\mathbf{z}_\star = c\mathbf{s} + \mathbf{r}' + \mathbf{r}, \qquad \mathbf{h}_\star = c\mathbf{e}_{\mathsf{kg}} + \mathbf{e}' + \mathbf{e},$$

and

$$\mathbf{A}\mathbf{z}_\star - c\mathbf{t}_0 + \mathbf{h}_\star = \mathbf{w}_0 + \mathbf{A}\mathbf{r} + \mathbf{e} = \mathbf{w}_\star.$$

Thus, conditioned on no abort, the joint distribution of the returned signature agrees exactly with that in $\mathbf{G}_2$.

The source entry must be masked whenever the source and shifted points are distinct. Indeed, the adversary knows its shift and can reconstruct $\mathbf{w}_0$ from the returned signature. Forwarding a subsequent source query to H_0 would reveal the same challenge c at both points, creating an artificial correlation that is absent from the shifted experiment. The reserved-source rule prevents this correlation by sampling an independent answer if the adversary later queries the source point. It does not impose $d \neq c$: equality remains possible with probability $1/|\mathcal{C}|$, exactly as for two independent random-oracle answers. Only the shifted point is programmed to the signature challenge, and H_0 itself is never reprogrammed.

If $\mathbf{A}\mathbf{r} + \mathbf{e} = 0 \pmod q$, then $\mathbf{w}_0 = \mathbf{w}_\star$, so the source and shifted points coincide and no separate source reservation is made.

Transferring the final forgery. Suppose that the adversary outputs an accepting shifted signature $(c^*, \mathbf{z}^*, \mathbf{h}^*)$ on a fresh digest μ^*, and define

$$\mathbf{w}^* := \mathbf{A}\mathbf{z}^* - c^*\mathbf{t}_0 + \mathbf{h}^*.$$

Every reserved source point and every programmed shifted entry has a digest that was previously submitted to the signing oracle, including those arising from abandoned signing attempts. Since μ^* is fresh, $(\mu^*, \mathbf{w}^*)$ is neither a reserved source point nor a signing-related programmed point.

Therefore, if $(\mu^*, \mathbf{w}^*)$ is already present in T, its value was obtained by forwarding a public query to H_0. If it has not yet been queried, the reduction forwards the verification query to H_0. In either case,

$$c^* = H_0(\mu^*, \mathbf{w}^*), \qquad \|(\mathbf{z}^*, \mathbf{h}^*)\|_2 \le B.$$

The reduction therefore outputs the unchanged pair $\big(\mu^*, (c^*, \mathbf{z}^*, \mathbf{h}^*)\big)$ as its ordinary EUF-CMA forgery.

Final bound. Combining the abort bound, the exact simulation conditioned on no abort, and the transfer of every successful good-execution forgery gives

$$\Pr[\mathsf{S}_0] \le \mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{Raccoon}_0,\mathcal{B}}(\lambda) + \varepsilon_{\mathsf{sim}},$$

with $\varepsilon_{\mathsf{sim}}$ as in the theorem statement. Note that the forgery has norm at most $B = B_2^{(0)}$, so it is accepted by the $\mathsf{Raccoon}_0$ verifier. This concludes the proof.

D Security of the Rounded Variants

For reference, we retain the rounded constructions and their existing analysis. The main security chain uses the non-rounded schemes in Section 3.

D.1 Security of Rounded Raccoon

Lemma 12 (Theorem 5.1 [DKM+24]). *Let Q_s be the number of signing queries. The signature scheme described above is correct and* EUF-CMA *secure under the* $\mathsf{H\text{-}MLWE}_{q,\ell,k,Q_s,\sigma_{\mathbf{t}},\sigma_{\mathbf{w}'},\mathcal{C}}$ *and* $\mathsf{SelfTargetMSIS}_{q,\ell+1,k,\mathcal{C},B_{\mathsf{MSIS}}}$ *assumptions.*

D.2 Rounded-to-Non-Rounded Conversion

Before stating the reduction, we fix the notation used to relate rounded commitments to their non-rounded representatives. Let

$$q_{\mathbf{w}} := \left\lfloor \frac{q}{2^{\nu_{\mathbf{w}}}} \right\rfloor$$

and write

$$q = 2^{\nu_{\mathbf{w}}} q_{\mathbf{w}} + q_{\perp,w}, \qquad 0 \le q_{\perp,w} < 2^{\nu_{\mathbf{w}}}.$$

For $\mathbf{w} \in \mathcal{R}_{q_{\mathbf{w}}}^k$, define the coefficient-wise canonical embedding

$$\iota_{\nu_{\mathbf{w}}}(\mathbf{w}) := 2^{\nu_{\mathbf{w}}}\overline{\mathbf{w}} \bmod q \in \mathcal{R}_q^k,$$

where $\overline{\mathbf{w}}$ denotes the canonical lift of $\mathbf{w}$ to $\{0, \ldots, q_{\mathbf{w}} - 1\}^{nk}$. By the rounding convention of Section 2.1,

$$\lfloor \iota_{\nu_{\mathbf{w}}}(\mathbf{w}) \rceil_{\nu_{\mathbf{w}}} = \mathbf{w}.$$

Hence $\iota_{\nu_{\mathbf{w}}}$ provides a canonical non-rounded representative of every rounded commitment.

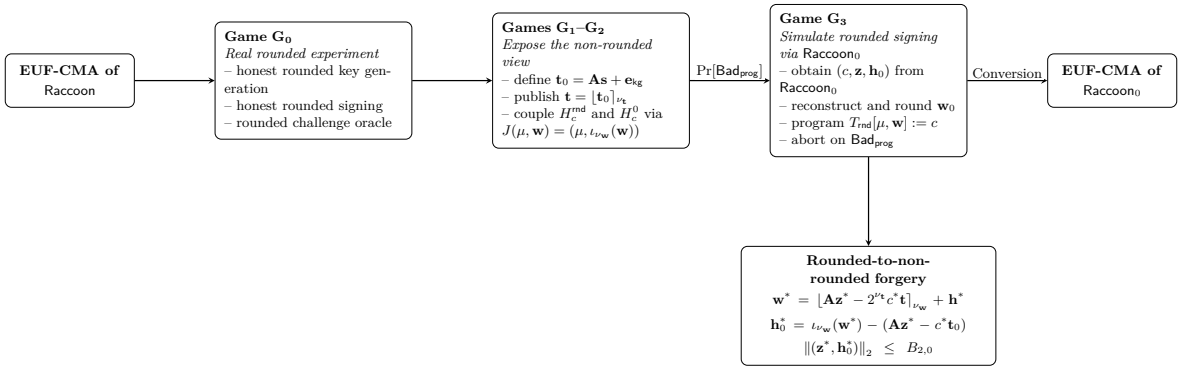


Fig. 9. Proof structure of the reduction from EUF-CMA security of rounded Raccoon to EUF-CMA security of $\mathsf{Raccoon}_0$. Arrows denote game transitions or reduction steps; the label $\Pr[\mathsf{Bad}_{\mathsf{prog}}]$ bounds the loss incurred when programming the rounded challenge oracle. The final step converts a rounded forgery into a valid non-rounded forgery using the canonical injection $\iota_{\nu_{\mathbf{w}}}$.

When converting rounded signatures to the non-rounded verification relation, scaled hints are interpreted using the same canonical lift followed by reduction modulo q as in the rounded verifier. The additional contribution to the Euclidean verification bound arising from public-key rounding, commitment rounding, and the possible remainder $q_{\perp,w}$ is therefore

$$\Delta_{\mathsf{rnd}} := \Big(\omega(2^{\nu_{\mathbf{t}}}-1) + (2^{\nu_{\mathbf{w}}}-1) + q_{\perp,w}\Big)\sqrt{nk}.$$

Theorem 8 (Reduction from Raccoon to $\mathsf{Raccoon}_0$). *Let $\mathsf{Raccoon}_{\nu_{\mathbf{t}},\nu_{\mathbf{w}}}$ denote Raccoon with modulus-rounding parameters $(\nu_{\mathbf{t}},\nu_{\mathbf{w}})$ and verification bound $B_{2,\mathsf{rnd}}$. Let $\mathsf{Raccoon}_0$ denote its non-rounded variant, with verification bound*

$$B_{2,0} := B_{2,\mathsf{rnd}} + \Big(\omega(2^{\nu_{\mathbf{t}}}-1) + (2^{\nu_{\mathbf{w}}}-1) + q_{\perp,w}\Big)\sqrt{nk},$$

where

$$q = 2^{\nu_{\mathbf{w}}} q_{\mathbf{w}} + q_{\perp,w}, \qquad q_{\mathbf{w}} = \left\lfloor \frac{q}{2^{\nu_{\mathbf{w}}}} \right\rfloor.$$

For every PPT adversary $\mathcal{A}$ making at most Q_s signing queries and Q_H non-signing challenge-oracle queries against $\mathsf{Raccoon}_{\nu_{\mathbf{t}},\nu_{\mathbf{w}}}$, there exists a PPT adversary $\mathcal{B}$ against $\mathsf{Raccoon}_0$ such that

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{Raccoon},\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{Raccoon}_0,\mathcal{B}}(\lambda) + \Pr[\mathsf{Bad}_{\mathsf{prog}}].$$

If every fresh rounded signing commitment, conditioned on the preceding history and chosen digest, has min-entropy at least κ_w, except with probability $\varepsilon_{\mathsf{ent}}$ per query, then

$$\Pr[\mathsf{Bad}_{\mathsf{prog}}] \le \left(Q_s Q_H + \frac{Q_s(Q_s-1)}{2}\right) 2^{-\kappa_w} + Q_s \varepsilon_{\mathsf{ent}}.$$

Proof. We prove the claim through a sequence of hybrid games. Let S_i denote the event that $\mathcal{A}$ outputs an accepting rounded signature on a digest μ^* not submitted to the signing oracle in game $\mathbf{G}_i$.

$\mathbf{G_0}$ This is the EUF-CMA experiment for Raccoon. The challenger generates an honest rounded verification key, answers signing queries honestly, and provides access to the rounded challenge oracle H_c^{rnd}. Thus $\Pr[\mathsf{S}_0]$ is the adversary's EUF-CMA success probability.

$\mathbf{G_1}$ We make explicit the non-rounded key underlying the public key:

$$\mathbf{t}_0 = \mathbf{A}\mathbf{s} + \mathbf{e}_{\mathsf{kg}}, \qquad \mathbf{t} = \lfloor \mathbf{t}_0 \rceil_{\nu_{\mathbf{t}}} .$$

The adversary receives only $\mathsf{vk}_{\mathsf{rnd}} = (\mathbf{A}, \mathbf{t})$. The Gaussian samples and the rounding map are unchanged, so

$$\Pr[\mathsf{S}_1] = \Pr[\mathsf{S}_0].$$

$\mathbf{G_2}$ Define

$$J(\mu, \mathbf{w}) := \big(\mu, \iota_{\nu_{\mathbf{w}}}(\mathbf{w})\big).$$

We implement H_c^{rnd} using a table T_{rnd} and the non-rounded random oracle H_c^0. On a query $y = (\mu, \mathbf{w})$, return $T_{\mathsf{rnd}}[y]$ if it is defined; otherwise set

$$T_{\mathsf{rnd}}[y] := H_c^0(J(y))$$

and return this value. Since $\iota_{\nu_{\mathbf{w}}}$ is injective on canonical representatives, J is injective. Hence distinct rounded oracle inputs map to distinct inputs of H_c^0, so the simulated oracle is distributed as a random function. Therefore

$$\Pr[\mathsf{S}_2] = \Pr[\mathsf{S}_1].$$

$\mathbf{G_3}$ We replace honest rounded signing by a simulator using the signing oracle of $\mathsf{Raccoon}_0$. On a signing query μ, it:

1. obtains a non-rounded signature $(c, \mathbf{z}, \mathbf{h}_0)$ on μ;
2. reconstructs its commitment and rounds it:

$$\mathbf{w}_0 := \mathbf{A}\mathbf{z} - c\mathbf{t}_0 + \mathbf{h}_0, \qquad \mathbf{w} := \lfloor \mathbf{w}_0 \rceil_{\nu_{\mathbf{w}}} ;$$

3. computes

$$\mathbf{y} := \lfloor \mathbf{A}\mathbf{z} - 2^{\nu_{\mathbf{t}}} c\mathbf{t} \rceil_{\nu_{\mathbf{w}}} , \qquad \mathbf{h} := \mathbf{w} - \mathbf{y};$$

4. if $T_{\mathsf{rnd}}[\mu, \mathbf{w}]$ is already defined, aborts and sets $\mathsf{Bad}_{\mathsf{prog}}$. Otherwise, it sets

$$T_{\mathsf{rnd}}[\mu, \mathbf{w}] := c$$

and returns $(c, \mathbf{z}, \mathbf{h})$.

For an honest non-rounded signature,

$$\mathbf{w}_0 = \mathbf{A}\mathbf{r}' + \mathbf{e}',$$

so the simulator uses exactly the same underlying sample as the honest rounded signer. Moreover, by construction,

$$\mathbf{w} = \lfloor \mathbf{A}\mathbf{z} - 2^{\nu_{\mathbf{t}}} c\mathbf{t} \rceil_{\nu_{\mathbf{w}}} + \mathbf{h}.$$

It remains to justify that, whenever $T_{\mathsf{rnd}}[\mu, \mathbf{w}]$ is undefined, the source challenge $c = H_c^0(\mu, \mathbf{w}_0)$ has not previously been exposed through the simulated rounded oracle. Suppose that an earlier rounded query $(\mu, \mathbf{w}')$ had been answered by querying H_c^0 at the source point $(\mu, \mathbf{w}_0)$. Then

$$J(\mu, \mathbf{w}') = (\mu, \iota_{\nu_{\mathbf{w}}}(\mathbf{w}')) = (\mu, \mathbf{w}_0),$$

and hence

$$\mathbf{w}' = \lfloor \iota_{\nu_{\mathbf{w}}}(\mathbf{w}') \rceil_{\nu_{\mathbf{w}}} = \lfloor \mathbf{w}_0 \rceil_{\nu_{\mathbf{w}}} = \mathbf{w}.$$

Therefore the earlier query would already have defined $T_{\mathsf{rnd}}[\mu, \mathbf{w}]$, contradicting the freshness test. Similarly, an earlier signing query using the same source commitment $\mathbf{w}_0$ would have produced the same rounded commitment $\mathbf{w}$ and defined the same table entry.

The same argument applies to future queries. If a rounded query can reach the source point $\mathbf{w}_0$ through J, then its rounded component must be exactly $\mathbf{w}$, whose programmed table entry takes priority. Thus there is no distinct public rounded point exposing the source challenge, and no source reservation is required.

Conditioned on $\neg\mathsf{Bad}_{\mathsf{prog}}$, the challenge c is fresh and uniform and the samples have exactly the same joint distribution as in honest rounded signing. Since both signing algorithms use the first sample, no additional conditioning on a norm test occurs. Consequently,

$$|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]| \le \Pr[\mathsf{Bad}_{\mathsf{prog}}].$$

Before the i-th signing query, T_{rnd} contains at most $Q_H + i - 1$ entries, accounting for public oracle queries and earlier signatures. Under the conditional min-entropy assumption, the new rounded commitment collides with this set with probability at most

$$(Q_H + i - 1)2^{-\kappa_w},$$

apart from the exceptional history event. Summing over all signing queries gives

$$\Pr[\mathsf{Bad}_{\mathsf{prog}}] \le \left(Q_s Q_H + \frac{Q_s(Q_s-1)}{2}\right) 2^{-\kappa_w} + Q_s \varepsilon_{\mathsf{ent}}.$$

Extracting a non-rounded forgery from $\mathbf{G_3}$. The reduction can implement $\mathbf{G}_3$ using only vk_0, the signing oracle of $\mathsf{Raccoon}_0$, and H_c^0. Suppose $\mathcal{A}$ outputs a successful rounded forgery $(\mu^*, c^*, \mathbf{z}^*, \mathbf{h}^*)$. Define

$$\mathbf{w}^* := \lfloor \mathbf{A}\mathbf{z}^* - 2^{\nu_{\mathbf{t}}} c^* \mathbf{t} \rceil_{\nu_{\mathbf{w}}} + \mathbf{h}^* \in \mathcal{R}_{q_{\mathbf{w}}}^k.$$

Since μ^* was never submitted to the signing oracle, the corresponding table entry was never programmed by a signing query. If it was queried previously, its value was obtained using the forwarding rule; otherwise the reduction evaluates it using that rule when verifying the forgery. Hence

$$c^* = H_c^0\big(\mu^*, \iota_{\nu_{\mathbf{w}}}(\mathbf{w}^*)\big). \tag{5}$$

Set $\widehat{\mathbf{w}}^* := \iota_{\nu_{\mathbf{w}}}(\mathbf{w}^*)$ and define

$$\mathbf{h}_0^* := \widehat{\mathbf{w}}^* - (\mathbf{A}\mathbf{z}^* - c^*\mathbf{t}_0) \in \mathcal{R}_q^k. \tag{6}$$

Then

$$\mathbf{A}\mathbf{z}^* - c^*\mathbf{t}_0 + \mathbf{h}_0^* = \widehat{\mathbf{w}}^*,$$

and Eq. (5) shows that $(c^*, \mathbf{z}^*, \mathbf{h}_0^*)$ satisfies the non-rounded challenge equation on the same fresh digest μ^*. It remains to bound its norm.

Norm of the extracted signature. Define the public-key rounding error

$$\delta_t := 2^{\nu_{\mathbf{t}}}\overline{\mathbf{t}} - \mathbf{t}_0 \pmod q.$$

By Lemma 11,

$$\|\delta_t\|_\infty \le 2^{\nu_{\mathbf{t}}} - 1. \tag{7}$$

Let

$$x^* := \mathbf{A}\mathbf{z}^* - 2^{\nu_{\mathbf{t}}} c^* \overline{\mathbf{t}} \in \mathcal{R}_q^k, \qquad \mathbf{u}^* := \lfloor x^* \rceil_{\nu_{\mathbf{w}}}.$$

Since $\mathbf{w}^* = \mathbf{u}^* + \mathbf{h}^*$ in $\mathcal{R}_{q_{\mathbf{w}}}^k$, there exists a coefficient vector $d \in \{0,1\}^{nk}$ such that, over the integers,

$$\overline{\mathbf{w}^*} = \overline{\mathbf{u}^*} + \overline{\mathbf{h}^*} - q_{\mathbf{w}} d. \tag{8}$$

Moreover, Lemma 11 gives $\delta_w \in \mathcal{R}_q^k$ such that

$$2^{\nu_{\mathbf{w}}}\overline{\mathbf{u}^*} = x^* + \delta_w \pmod q, \qquad \|\delta_w\|_\infty \le 2^{\nu_{\mathbf{w}}} - 1. \tag{9}$$

Since $2^{\nu_{\mathbf{t}}}\bar{\mathbf{t}} = \mathbf{t}_0 + \delta_t \pmod q$,

$$x^* = \mathbf{A}\mathbf{z}^* - c^*\mathbf{t}_0 - c^*\delta_t \pmod q.$$

Using Eqs. (8) and (9), we obtain

$$\begin{aligned}\widehat{\mathbf{w}}^* &= 2^{\nu_{\mathbf{w}}}\overline{\mathbf{w}^*} \pmod q \\ &= \mathbf{A}\mathbf{z}^* - c^*\mathbf{t}_0 - c^*\delta_t + 2^{\nu_{\mathbf{w}}}\overline{\mathbf{h}^*} - 2^{\nu_{\mathbf{w}}}q_w d + \delta_w \pmod q.\end{aligned}$$

Since $2^{\nu_{\mathbf{w}}}q_{\mathbf{w}} = q - q_{\perp,w}$, this yields

$$\mathbf{h}_0^* = 2^{\nu_{\mathbf{w}}}\overline{\mathbf{h}^*} - c^*\delta_t + \delta_w + q_{\perp,w}d \pmod q. \tag{10}$$

Since $\|c^*\|_1 = \omega$,

$$\|c^*\delta_t\|_2 \le \omega(2^{\nu_{\mathbf{t}}} - 1)\sqrt{nk}.$$

Furthermore,

$$\|\delta_w\|_2 \le (2^{\nu_{\mathbf{w}}} - 1)\sqrt{nk}, \qquad \|q_{\perp,w}d\|_2 \le q_{\perp,w}\sqrt{nk}.$$

Using the scaled-hint norm of the rounded verifier and Eq. (10), we conclude that

$$\begin{aligned}\|(\mathbf{z}^*, \mathbf{h}_0^*)\|_2 &\le \left\|\left(\mathbf{z}^*, 2^{\nu_{\mathbf{w}}}\overline{\mathbf{h}^*} \bmod q\right)\right\|_2 \\ &\quad + \left(\omega(2^{\nu_{\mathbf{t}}} - 1) + (2^{\nu_{\mathbf{w}}} - 1) + q_{\perp,w}\right)\sqrt{nk} \\ &\le B_{2,\mathsf{rnd}} + \left(\omega(2^{\nu_{\mathbf{t}}} - 1) + (2^{\nu_{\mathbf{w}}} - 1) + q_{\perp,w}\right)\sqrt{nk} \\ &= B_{2,0}.\end{aligned}$$

Thus $(c^*, \mathbf{z}^*, \mathbf{h}_0^*)$ is a valid non-rounded Gaussian Raccoon signature on the fresh digest μ^*.

Final bound. The reduction outputs $(\mu^*, c^*, \mathbf{z}^*, \mathbf{h}_0^*)$. The preceding argument shows that this signature is accepted by $\mathsf{Raccoon}_0$, and hence

$$\Pr[\mathsf{S}_3] \le \mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{Raccoon}_0,\mathcal{B}}(\lambda).$$

Combining the game transitions yields

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{Raccoon},\mathcal{A}}(\lambda) = \Pr[\mathsf{S}_0] \le \mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{Raccoon}_0,\mathcal{B}}(\lambda) + \Pr[\mathsf{Bad}_{\mathsf{prog}}],$$

which is the claimed bound.

E Security Reduction of our Blind Signature

E.1 Correctness

Proof. For honest executions of $\mathsf{BRaccoon}$, the resulting tuple $(c, \mathbf{z}, \mathbf{h})$ has the same algebraic form as a shifted Raccoon signature $\mathsf{Raccoon}_0^\star$, with the additional errors introduced by public-key and commitment rounding. The corresponding contribution to the verification norm is therefore bounded by:

$$e^{\frac{1}{4}}(\omega\sigma_{\mathbf{t}} + \sigma_{\mathbf{w}'} + \sigma_{\mathbf{w}})\sqrt{n(k+\ell)},$$

as in the non-rounded shifted scheme, while the rounding terms contribute at most

$$\left(\omega(2^{\nu_{\mathbf{t}}} - 1) + 2^{\nu_{\mathbf{w}}} - 1 + f(\nu_{\mathbf{w}})\right)\sqrt{nk}.$$

Hence, by the triangle inequality, the verification norm is at most $B_2^\star$ except with negligible probability, and the verification equation holds by construction. Observe that when $\nu_{\mathbf{t}} = \nu_{\mathbf{w}} = 0$, all rounding-error terms vanish (and $f(0) = 0$), so the bound reduces exactly to the correctness bound of the non-rounded shifted scheme $\mathsf{Raccoon}_0^\star$. Correctness of the auxiliary primitives ensures that honest executions do not abort before producing the signature.

E.2 One-More Unforgeability

Proof (Proof of Theorem 6). We prove one-more unforgeability via a sequence of hybrid games.

$\mathbf{G_0}$: We start with the initial game $\mathbf{G_0}$, in which the protocol is executed honestly and the adversary $\mathcal{A}$ interacts with the OMUF game instantiated with our protocol shown in Figure 5 and Figure 6. By definition of the OMUF game, the adversary $\mathcal{A}$ issues at most Q_s signing queries, which it may open in parallel and interleave arbitrarily; concurrency is therefore a property of the one-more unforgeability game, while blindness is the standard two-session notion of Definition 7. The reduction $\mathcal{R}$ forwards all messages from the signer to the adversary without any modification. Let S_0 denote the event that the adversary wins in game $\mathbf{G}_0$.

$\mathbf{G_1}$: In game $\mathbf{G}_1$, we modify the way the public parameters of the commitment scheme are generated. Recall that, in the real protocol, the public matrices used by CS are not sampled explicitly by a trusted party, but are deterministically expanded from public seeds. We model this seed expansion as a random oracle $\mathcal{O}_{\mathsf{setup}}$. Instead of answering the setup-expansion queries honestly, the challenger first samples the public seeds that will define the parameters of CS and then programs $\mathcal{O}_{\mathsf{setup}}$ on these seeds so that the resulting matrices are distributed as in the simulated setup

$$(\mathsf{ck}, \mathcal{T}) \leftarrow \mathsf{Setup}_{\mathsf{Ext}}(1^\lambda).$$

All subsequent queries to $\mathcal{O}_{\mathsf{setup}}$ are answered consistently with this programming, while all other oracle queries are answered as in the real game. The only way in which this programming can be detected before the public seeds are revealed is if the adversary queries $\mathcal{O}_{\mathsf{setup}}$ on one of these seeds in advance. Let $\mathsf{Bad}_{\mathsf{setup}}$ denote this event. If the seeds have length κ and the adversary makes at most Q_{setup} setup-oracle queries, then

$$\Pr[\mathsf{Bad}_{\mathsf{setup}}] \leq \frac{Q_{\mathsf{setup}}}{2^\kappa}.$$

Conditioned on $\neg\mathsf{Bad}_{\mathsf{setup}}$, the adversary's view differs from the real setup only in that the public matrices are generated according to the simulated setup distribution. Hence, by setup indistinguishability of the commitment parameters, there exists a PPT adversary $\mathcal{B}$ such that

$$\left|\Pr[\mathsf{S}_0] - \Pr[\mathsf{S}_1]\right| \leq \mathsf{Adv}^{\mathsf{SetupInd}}_{\mathsf{CS},\mathcal{B}}(\lambda) + \frac{Q_{\mathsf{setup}}}{2^\kappa}.$$

For our BDLOP-based instantiation, setup indistinguishability follows from the MLWE assumption, since the simulated commitment parameters are computationally indistinguishable from honestly generated parameters.

$\mathbf{G_2}$: In game $\mathbf{G}_2$, upon receiving a commitment $C_{\mathsf{msg},\mathbf{r},\mathbf{e}}$, the reduction uses the trapdoor $\mathcal{T}$ to extract $(\mathsf{msg}, \mathbf{r}, \mathbf{e})$. If extraction fails, the game aborts. Let F_2 denote this event. Games $\mathbf{G}_1$ and $\mathbf{G}_2$ are identical unless F_2 occurs. Therefore, by extractability of CS and a union bound over the at most Q_s initiated sessions, there exists a PPT adversary $\mathcal{X}$ such that

$$\left|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]\right| \leq Q_s \cdot \mathsf{Adv}^{\mathsf{Ext}}_{\mathsf{CS},\mathcal{X}}(\lambda).$$

$\mathbf{G_3}$: In game $\mathbf{G}_3$, replace each signer proof π_2 by its simulated proof. The signer knows a witness for L_2: its signing key, first-round randomness, and the randomness used in the prescribed encryption and evaluation. These relations certify its computation on the received ciphertext; they do not require that ciphertext to encrypt the intended challenge. By zero-knowledge,

$$\left|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]\right| \leq Q_s \mathsf{Adv}^{\mathsf{ZK}}_{\pi_2,\mathcal{D}}(\lambda).$$

$\mathbf{G_4}$: Verify π_1 as in the protocol and reject the continuation if it fails. Otherwise, using the tuple $(\mathsf{msg}, \mathbf{r}, \mathbf{e})$ extracted from the commitment C, the reduction recomputes the values that the adversary was required to certify through π_1, namely

$$\mu = \mathsf{H}(\mathsf{vk}, \mathsf{msg}), \qquad \mathbf{w} = \lfloor \mathbf{w}' + \mathbf{A}\mathbf{r} + \mathbf{e} \rceil_{\nu_{\mathbf{w}}}, \qquad c' = H_{\mathsf{rnd}}(\mu, \mathbf{w}).$$

The reduction then checks that the accepted proof is consistent with these recomputed values and with the relations defining L_1. In particular, the extracted tuple must satisfy the prescribed domain and shift bounds, $\mathbf{w}$ must be the corresponding rounded commitment, c' must be the challenge derived from $(\mu, \mathbf{w})$, and ctx_c must be an encryption of c'.

If π_1 is accepted while any of these consistency conditions fails, then the adversary has produced an accepting proof for a false L_1 statement. Game $\mathbf{G}_4$ aborts on this event. The event is determined by the transcript, so it need not be efficiently recognisable: the soundness adversary $\mathcal{E}$ simply runs the game and outputs the statement and the accepted proof, and the soundness experiment decides membership in L_1.

Record c' for the next hybrid; the homomorphic ciphertext is still formed as before. By soundness and a union bound over the at most Q_s signing sessions, there exists a PPT adversary $\mathcal{E}$ such that

$$\left|\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4]\right| \leq Q_s \mathsf{Adv}^{\mathsf{SND}}_{\pi_1,\mathcal{E}}(\lambda).$$

$\mathbf{G_5}$: After an accepting continuation, let c' be the challenge fixed in $\mathbf{G}_4$. The signer's private affine evaluation produces a ciphertext encrypting

$$y := c'\mathbf{s} + \mathbf{r}'.$$

We replace this evaluated ciphertext by

$$\widetilde{\mathsf{ctx}}_{z'} \leftarrow \mathsf{Sim}_{\mathsf{HE}}(\mathsf{pp}, \mathsf{ek}, y) = \mathsf{HE.Enc}(\mathsf{pp}, \mathsf{ek}, y; \rho_{\mathsf{fl}}), \qquad \rho_{\mathsf{fl}} \leftarrow \mathsf{D}_{\mathsf{fl}}.$$

Thus, $\widetilde{\mathsf{ctx}}_{z'}$ is a fresh encryption of the same output plaintext under the flooding distribution, rather than the result of the private affine evaluation. Outside the soundness bad event already charged in $\mathbf{G}_4$, the input ciphertext ctx_c has an admissible opening as an encryption of c'. Therefore, the chosen-key circuit-privacy guarantee of Lemma 8, applied to the bounded affine function

$$f_{\mathbf{s},\mathbf{r}'}(c) = c\mathbf{s} + \mathbf{r}',$$

gives

$$\mathsf{HE.Eval}_{\mathsf{priv}}(\mathsf{pp}, \mathsf{ek}, f_{\mathbf{s},\mathbf{r}'}, \mathsf{ctx}_c) \approx \mathsf{Sim}_{\mathsf{HE}}(\mathsf{pp}, \mathsf{ek}, c'\mathbf{s} + \mathbf{r}').$$

Replacing the responses one at a time and applying a hybrid argument over at most Q_s signing responses therefore yields

$$\left|\Pr[\mathsf{S}_4] - \Pr[\mathsf{S}_5]\right| \leq Q_s \mathsf{Adv}^{\mathsf{CP}}_{\mathsf{HE},\mathcal{V}}(\lambda).$$

$\mathbf{G_6}$: We now rewrite key generation in the form required by the target $\mathsf{Raccoon}^{\star}_0$ experiment. Sample

$$\mathbf{t}_0 = \mathbf{A}\mathbf{s} + \mathbf{e}_{\mathsf{kg}}$$

and publish

$$\mathsf{vk}_{\mathsf{rnd}} = (\mathbf{A}, \mathbf{t}), \qquad \mathbf{t} = \lfloor \mathbf{t}_0 \rceil_{\nu_{\mathbf{t}}}.$$

This is only a change in the order in which the honestly distributed verification key is generated and therefore does not alter the adversary's view. All message digests continue to be computed as

$$\mu = \mathsf{H}(\mathsf{vk}_{\mathsf{rnd}}, \mathsf{msg}).$$

The auxiliary signing oracle of $\mathsf{Raccoon}^{\star}_0$ used in the subsequent games is invoked directly on this digest μ. We additionally implement the rounded challenge oracle H_{rnd} by lazy sampling. The challenger

maintains an initially empty table T. On input u, if $T[u]$ is already defined, it returns $T[u]$; otherwise it samples

$$c \leftarrow\!\$\, \mathcal{C},$$

sets $T[u] := c$, and returns c. We write

$$\mathrm{dom}(T) := \{u : T[u] \text{ is defined}\}$$

for the set of challenge-oracle inputs that have already been assigned an output.

These changes are purely representational and therefore

$$\Pr[\mathsf{S}_5] = \Pr[\mathsf{S}_6].$$

$\mathbf{G_7}$: Using the canonical injection $\iota : \mathcal{R}_{q_{\mathbf{w}}}^k \to \mathcal{R}_q^k$ from Lemma 4, define $J(\mu, \mathbf{u}) := (\mu, \iota(\mathbf{u}))$. For a signing session with extracted admissible shift $(\mathbf{r}, \mathbf{e})$, the signer samples fresh $(\mathbf{r}', \mathbf{e}') \leftarrow \mathsf{D}_{\mathbf{w}'}^{\ell} \times \mathsf{D}_{\mathbf{w}'}^{k}$ and computes

$$\mathbf{w}' = \mathbf{A}\mathbf{r}' + \mathbf{e}', \qquad \mathbf{w}_{\mathsf{raw}} = \mathbf{w}' + \mathbf{A}\mathbf{r} + \mathbf{e}, \qquad \mathbf{w} = \lfloor \mathbf{w}_{\mathsf{raw}} \rceil_{\nu_{\mathbf{w}}}.$$

Let $y := (\mu, \mathbf{w})$. If $y \in \mathrm{dom}(T)$, the challenger aborts; let $\mathsf{Bad}_{\mathsf{prog}}$ denote the event that such an abort occurs. Otherwise, it samples $c \leftarrow\!\$\, \mathcal{C}$, sets $T[y] := c$, and sends $\mathbf{w}'$ to the adversary. The remainder of the session proceeds as in $\mathbf{G}_5$, using the stored value c whenever y is queried. We now bound $\Pr[\mathsf{Bad}_{\mathsf{prog}}]$. By Lemma 5, except for a global bad-matrix event $\mathsf{Bad}_{\mathbf{A}}$ with $\Pr[\mathsf{Bad}_{\mathbf{A}}] \le 2^{-n+1}$, every fresh rounded commitment $\mathbf{w}$ has conditional min-entropy at least $\kappa_{\mathsf{br}} := n - 1$, conditioned on the complete history before $(\mathbf{r}', \mathbf{e}')$ is sampled.

Before the i-th signing session, T contains at most $Q_H + i - 1$ entries. Hence, conditioned on $\neg\mathsf{Bad}_{\mathbf{A}}$,

$$\Pr[\mathbf{w} \in \mathrm{dom}(T)] \le (Q_H + i - 1)2^{-\kappa_{\mathsf{br}}}.$$

Summing over at most Q_s signing sessions gives

$$\Pr[\mathsf{Bad}_{\mathsf{prog}} \mid \neg\mathsf{Bad}_{\mathbf{A}}] \le \left(Q_s Q_H + \frac{Q_s(Q_s-1)}{2}\right) 2^{-\kappa_{\mathsf{br}}}.$$

Therefore,

$$\Pr[\mathsf{Bad}_{\mathsf{prog}}] \le \varepsilon_{\mathsf{br}} := \left(Q_s Q_H + \frac{Q_s(Q_s-1)}{2}\right) 2^{-(n-1)} + 2^{-n+1}.$$

Games $\mathbf{G}_6$ and $\mathbf{G}_7$ are identical unless $\mathsf{Bad}_{\mathsf{prog}}$ occurs, so

$$|\Pr[\mathsf{S}_6] - \Pr[\mathsf{S}_7]| \le \varepsilon_{\mathsf{br}}.$$

$\mathbf{G_8}$: We now implement the preceding game using the signing oracle of the non-rounded shifted scheme $\mathsf{Raccoon}_0^{\star}$ and its challenge oracle H_0. The reduction maintains the same lazy-sampling table T as in $\mathbf{G}_7$.

A public rounded challenge query y is answered by returning $T[y]$ if defined; otherwise the reduction sets $T[y] := H_0(J(y))$ and returns this value.

If the tuple $(\mathsf{msg}, \mathbf{r}, \mathbf{e})$ extracted from the first-round commitment is not admissible, i.e., $\|(\mathbf{r}, \mathbf{e})\|_2 > B_{\mathbf{w}}$, the reduction does not query the auxiliary signing oracle: it samples $(\mathbf{r}', \mathbf{e}')$ itself, sends $\mathbf{w}' = \mathbf{A}\mathbf{r}' + \mathbf{e}'$, and will reject the continuation, since by binding of the commitment the L_1 statement of that session is false and an accepting π_1 is the soundness event already charged in $\mathbf{G}_4$. Consider now a signing session for which the reduction has extracted an admissible tuple $(\mathsf{msg}, \mathbf{r}, \mathbf{e})$, and let $\mu = \mathsf{H}(\mathsf{vk}_{\mathsf{rnd}}, \mathsf{msg})$. Before sending the first signer message, the reduction makes one query to the signing oracle of $\mathsf{Raccoon}_0^{\star}$ on $(\mu, \mathbf{r}, \mathbf{e})$ and obtains $(c, \mathbf{z}, \mathbf{h}_0)$. It reconstructs

$$\mathbf{w}_{\mathsf{raw}} := \mathbf{A}\mathbf{z} - c\mathbf{t}_0 + \mathbf{h}_0, \qquad \mathbf{w}' := \mathbf{w}_{\mathsf{raw}} - \mathbf{A}\mathbf{r} - \mathbf{e}, \qquad \mathbf{w} := \lfloor \mathbf{w}_{\mathsf{raw}} \rceil_{\nu_{\mathbf{w}}}. \tag{11}$$

Set $x := (\mu, \mathbf{w}_{\mathsf{raw}})$ and $y := (\mu, \mathbf{w})$.

If $T[y]$ is already defined, the reduction aborts, defining the same event $\mathsf{Bad}_{\mathsf{prog}}$ as in $\mathbf{G}_7$. Otherwise, it sets $T[y] := c$, stores $(\mu, \mathbf{r}, \mathbf{e}, c, \mathbf{z}, \mathbf{w}')$ as the session state, and sends $\mathbf{w}'$ to the adversary.

After receiving (ctx_c, π_1) and performing the same checks as in the preceding games, the reduction returns $\mathsf{ctx}_{z'} \leftarrow \mathsf{Sim}_{\mathsf{HE}}(\mathsf{pp}, \mathsf{ek}, \mathbf{z} - \mathbf{r})$ together with the simulated proof π_2.

We compare this execution with $\mathbf{G}_7$ by coupling each call to the $\mathsf{Raccoon}_0^\star$ signing oracle with its first internal signing sample $(\mathbf{r}', \mathbf{e}')$. If this sample passes the norm test, then

$$\mathbf{z} = c\mathbf{s} + \mathbf{r}' + \mathbf{r}, \qquad \mathbf{h}_0 = c\mathbf{e}_{\mathsf{kg}} + \mathbf{e}' + \mathbf{e},$$

and hence Eq. (11) yields

$$\mathbf{w}_{\mathsf{raw}} = \mathbf{A}(\mathbf{r}' + \mathbf{r}) + \mathbf{e}' + \mathbf{e}, \qquad \mathbf{w}' = \mathbf{A}\mathbf{r}' + \mathbf{e}', \qquad \mathbf{z} - \mathbf{r} = c\mathbf{s} + \mathbf{r}'.$$

Thus the reconstructed first message and the simulated encrypted response have exactly the same joint distribution as in $\mathbf{G}_7$.

It remains to compare the challenge oracles. The auxiliary signing oracle uses $c = H_0(x)$. Suppose that a previous public rounded query u had been forwarded to the same source point x. Then $J(u) = x$, so

$$u = \lfloor \iota(u) \rceil_{\nu_{\mathbf{w}}} = \lfloor \mathbf{w}_{\mathsf{raw}} \rceil_{\nu_{\mathbf{w}}} = \mathbf{w}.$$

Hence the corresponding rounded point is exactly y, which would already belong to $\mathrm{dom}(T)$, contradicting the freshness test. Likewise, an earlier signing query to the $\mathsf{Raccoon}_0^\star$ auxiliary oracle using the same source point x would have produced the same rounded commitment $\mathbf{w}$ and already defined $T[y]$.

Conversely, any later public rounded query reaching x through J must again be the point y, whose programmed table entry takes priority. Thus no distinct rounded challenge point exposes the source challenge $H_0(x)$.

It remains only to account for internal restarts of the $\mathsf{Raccoon}_0^\star$ signing oracle. Let $\mathsf{Bad}_{\mathsf{corr}}$ denote the event that, in one of the at most Q_s auxiliary signing queries, the first internal signing sample fails the prescribed norm bound. By the choice $B \geq B^{(0)}_{2,\mathsf{corr}} + B_{\mathbf{w}}$ and correctness of the non-rounded shifted signing distribution,

$$\Pr[\mathsf{Bad}_{\mathsf{corr}}] \leq \varepsilon_{\mathsf{corr}}.$$

Conditioned on $\neg\mathsf{Bad}_{\mathsf{corr}}$ and $\neg\mathsf{Bad}_{\mathsf{prog}}$, the joint distribution of the signing response and the simulated rounded challenge oracle is exactly the same as in $\mathbf{G}_7$. Therefore,

$$|\Pr[\mathsf{S}_7] - \Pr[\mathsf{S}_8]| \leq \varepsilon_{\mathsf{corr}}.$$

Transferring a rounded forgery. Let $\mathcal{M}_{\mathsf{qry}}$ contain all messages used in auxiliary signing queries, including those from abandoned sessions. Since $|\mathcal{M}_{\mathsf{qry}}| \leq Q_s$, among the $Q_s + 1$ valid outputs on distinct messages there exists $\mathsf{msg}^\dagger \notin \mathcal{M}_{\mathsf{qry}}$. Outside the prehash-collision event, $\mu^\dagger := \mathsf{H}(\mathsf{vk}_{\mathsf{rnd}}, \mathsf{msg}^\dagger)$ is therefore fresh with respect to the auxiliary signing oracle.

Let $(c^\dagger, \mathbf{z}^\dagger, \mathbf{h}^\dagger)$ be the corresponding accepted rounded signature, and let

$$\mathbf{w}^\dagger := \left\lfloor \mathbf{A}\mathbf{z}^\dagger - 2^{\nu_{\mathbf{t}}} c^\dagger \mathbf{t} \right\rceil_{\nu_{\mathbf{w}}} + \mathbf{h}^\dagger \in \mathcal{R}_{q_{\mathbf{w}}}^k$$

be the value reconstructed by the rounded verifier.

Outside the prehash-collision event, $\mu^\dagger$ was never used in a signing query to the $\mathsf{Raccoon}_0^\star$ auxiliary oracle. Hence the corresponding rounded challenge point was never programmed by the signing simulation. If it was queried previously, its value was obtained by forwarding through J; otherwise, the reduction performs this forwarding query when checking the final forgery. Therefore,

$$c^\dagger = H_0\big(\mu^\dagger, \iota(\mathbf{w}^\dagger)\big).$$

Applying Lemma 4, define

$$\mathbf{h}_0^\dagger := \iota(\mathbf{w}^\dagger) - (\mathbf{A}\mathbf{z}^\dagger - c^\dagger \mathbf{t}_0) \pmod q.$$

Then $(c^\dagger, \mathbf{z}^\dagger, \mathbf{h}_0^\dagger)$ satisfies the non-rounded verification equation and

$$\left\|(\mathbf{z}^\dagger, \mathbf{h}_0^\dagger)\right\|_2 \leq B_2^\star + \Delta_{\mathsf{rnd}} \leq B.$$

Thus $\left(\mu^\dagger, (c^\dagger, \mathbf{z}^\dagger, \mathbf{h}_0^\dagger)\right)$ is a valid EUF-CMA forgery for $\mathsf{Raccoon}_0^\star$.

Therefore, in $\mathbf{G}_8$, outside the prehash-collision event, every successful OMUF output yields a valid EUF-CMA forgery for $\mathsf{Raccoon}_0^\star$. Combining the preceding game transitions with the losses $\varepsilon_{\mathsf{br}}$, $\varepsilon_{\mathsf{corr}}$, and $\varepsilon_{\mathsf{hash}}$ proves the claimed bound. □

E.3 Blindness

Proof. We prove blindness of our protocol via a sequence of hybrid games.

$\mathbf{G_0}$. This game corresponds to the blindness experiment defined in Section 2.5. The adversary $\mathcal{A}$ selects two messages and a verification key. A random bit $b \in \{0,1\}$ is chosen, and the challenger executes the protocol with the adversary using msg_b. At the end, the adversary receives signatures $(\mathsf{sig}_b, \mathsf{sig}_{1-b})$ and outputs a guess for b. Let S_0 denote the event that $\mathcal{A}$ correctly guesses b.

$\mathbf{G_1}$. In game $\mathbf{G_1}$, we modify the subroutine User_2 by replacing the proof π_1 with a simulated proof generated by the simulator of the NIZK scheme. Here and in the following hops, the modification is applied to both sessions of the blindness experiment, one session at a time, which accounts for the factor 2 in each bound.

Let S_1 denote the event that $\mathcal{A}$ correctly guesses b in $\mathbf{G_1}$. By the zero-knowledge property of π_1, there exists a PPT adversary $\mathcal{B}$ such that

$$|\Pr[\mathsf{S}_0] - \Pr[\mathsf{S}_1]| \leq 2\mathsf{Adv}^{\mathsf{ZK}}_{\pi_1,\mathcal{B}}(\lambda).$$

$\mathbf{G_2}$. In game $\mathbf{G_2}$, we modify the subroutine User_1 so that the commitment $C_{\mathsf{msg},\mathbf{r},\mathbf{e}}$ is replaced by a commitment C^* to a uniformly random message from the message space of the commitment scheme.

Let S_2 denote the event that $\mathcal{A}$ correctly guesses b in $\mathbf{G_2}$. By the hiding property of the commitment scheme, there exists a PPT adversary $\mathcal{C}$ such that

$$|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]| \leq 2\mathsf{Adv}^{\mathsf{HID}}_{\mathsf{CS},\mathcal{C}}(\lambda).$$

$\mathbf{G_3}$. In game $\mathbf{G_3}$, we modify the subroutine User_3 so that it attempts to extract a witness from the proof π_2 after verification. If extraction fails, the subroutine aborts; otherwise it proceeds as in $\mathbf{G_2}$.

Let S_3 denote the event that $\mathcal{A}$ correctly guesses b in $\mathbf{G_3}$. The view differs from $\mathbf{G_2}$ only if extraction fails. By knowledge soundness of π_2, there exists a PPT adversary $\mathcal{D}$ such that

$$|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]| \leq 2\mathsf{Adv}^{\mathsf{KS}}_{\pi_2,\mathcal{D}}(\lambda).$$

$\mathbf{G_4}$. In game $\mathbf{G_4}$, we modify the subroutine User_3 so that, using the extracted witness $(\mathbf{s}, \mathbf{r}')$, it recomputes

$$\mathbf{z} = c\mathbf{s} + \mathbf{r}' + \mathbf{r},$$

where $\mathbf{r}$ is the blinding randomness chosen in User_1.

By the relation L_2, the extracted witness satisfies

$$\mathsf{ctx}_{\mathbf{z}'} = \mathsf{HE.Eval}(\mathsf{pp}, \mathsf{ek}, f_{\mathbf{s}}, (\mathsf{ctx}_c, \mathsf{ctx}_{\mathbf{r}'})),$$

where $\mathsf{ctx}_{\mathbf{r}'}$ is an encryption of $\mathbf{r}'$ under the flooding distribution and all randomness lies within the bounds of L_2, so decryption correctness of HE gives $\mathsf{HE.Dec}(\mathsf{pp}, \mathsf{dk}, \mathsf{ctx}_{\mathbf{z}'}) = c\mathbf{s} + \mathbf{r}'$. The recomputed value therefore equals the one computed in the honest execution, and the distribution of $\mathbf{z}$ is unchanged.

Let S_4 denote the event that $\mathcal{A}$ correctly guesses b in $\mathbf{G_4}$. We therefore have

$$\Pr[\mathsf{S}_3] = \Pr[\mathsf{S}_4].$$

$\mathbf{G_5}$. In game $\mathbf{G_5}$, we modify the subroutine User_2 by replacing the ciphertext ctx_c with an encryption of an independently sampled value c^*. This hop is possible only because of the previous game: since $\mathbf{G_4}$, the challenger no longer decrypts $\mathsf{ctx}_{\mathbf{z}'}$, so the decryption key dk is never used, and the reduction to IND-CPA can embed the challenge encryption key ek from its own experiment. The proof π_1 for the (now false) statement is still simulated as in $\mathbf{G_1}$.

Let S_5 denote the event that $\mathcal{A}$ correctly guesses b in $\mathbf{G_5}$. By IND-CPA security of the homomorphic encryption scheme HE, there exists a PPT adversary $\mathcal{E}$ such that

$$|\Pr[\mathsf{S}_4] - \Pr[\mathsf{S}_5]| \leq 2\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{HE},\mathcal{E}}(\lambda).$$

$\mathbf{G_6}$. In the final game $\mathbf{G_6}$, all components generated by the user are independent of the challenger's bit b: the proofs are simulated, the commitment hides the message, the ciphertext encrypts an independent value, and the response $\mathbf{z}$ is computed from extracted values and fresh blinding randomness, independently of the challenge bit.

It remains to bound the statistical dependence of the two output signatures on the signer's per-session randomness. Write $\mathbf{u}_j := (\mathbf{r}'_j, \mathbf{e}'_j)$ for the signer's randomness of session j, as extracted from π_2, and $\mathbf{v} := (\mathbf{r}, \mathbf{e}) \leftarrow\!\$\ \mathsf{D}^{\ell+k}_{\mathbf{w}}$ for the user's blinding vector of that session. In $\mathbf{G_6}$ the signature produced in session j is a deterministic function of the public values, of $(\mathbf{s}, \mathbf{e}_{\mathsf{kg}})$ and of the sum $\mathbf{u}_j + \mathbf{v}$: indeed $\mathbf{z} = c\mathbf{s} + \mathbf{r}'_j + \mathbf{r}$, the rounded commitment is $\mathbf{w} = \lfloor \mathbf{A}(\mathbf{r}'_j + \mathbf{r}) + (\mathbf{e}'_j + \mathbf{e}) \rceil_{\nu_{\mathbf{w}}}$, the challenge is $c = \mathsf{H}_c(\mu, \mathbf{w})$ and $\mathbf{h}$ is determined by $(\mathbf{z}, c, \mathbf{w})$. Swapping the bit b swaps which session's $\mathbf{u}_j$ is added to the blinding vector of the signature on msg_i. By the data-processing inequality, the statistical distance between the two views is therefore at most

$$2\,\Delta\big(\mathbf{u}_0 + \mathsf{D}^{\ell+k}_{\mathbf{w}},\ \mathbf{u}_1 + \mathsf{D}^{\ell+k}_{\mathbf{w}}\big) = 2\,\Delta\big(\mathsf{D}^{\ell+k}_{\mathbf{w}},\ (\mathbf{u}_1 - \mathbf{u}_0) + \mathsf{D}^{\ell+k}_{\mathbf{w}}\big).$$

Since the extracted witnesses satisfy $\|\mathbf{u}_j\|_2 \leq B_{\mathbf{w}'}$ (this is where the norm bound of L_2 is used), we have $\|\mathbf{u}_1 - \mathbf{u}_0\|_2 \leq 2B_{\mathbf{w}'}$. For a shift $\mathbf{c}$ with $2\pi\|\mathbf{c}\|_2^2 \leq \sigma_{\mathbf{w}}^2$ and $\sigma_{\mathbf{w}} \geq \eta_\epsilon(\mathbb{Z}^{n(\ell+k)})$ with $\epsilon \leq 1/4$, the Rényi-divergence bound of [BLL+15, Lem. 4.2] for order $a = 2$ gives

$$R_2\big(\mathbf{c} + \mathsf{D}^{\ell+k}_{\mathbf{w}} \,\big\|\, \mathsf{D}^{\ell+k}_{\mathbf{w}}\big) \leq \left(\tfrac{1+\epsilon}{1-\epsilon}\right)^2 \exp\!\left(\tfrac{2\pi\|\mathbf{c}\|_2^2}{\sigma_{\mathbf{w}}^2}\right) \leq 1 + \frac{2\pi\|\mathbf{c}\|_2^2}{\sigma_{\mathbf{w}}^2} + 24\epsilon,$$

and since $R_2 - 1$ is the χ^2-divergence, the statistical distance is at most

$$\tfrac{1}{2}\sqrt{R_2 - 1} \leq \sqrt{\pi}\,\|\mathbf{c}\|_2/\sigma_{\mathbf{w}} + \sqrt{6\epsilon}.$$

Altogether,

$$\left|\Pr[\mathsf{S}_5] - \frac{1}{2}\right| \leq \epsilon_{\mathsf{Stat}} := 4\sqrt{\pi}\,\frac{B_{\mathbf{w}'}}{\sigma_{\mathbf{w}}} + 2\sqrt{6\epsilon}.$$

Final bound. Combining the bounds from all hybrids, we conclude that for any PPT adversary $\mathcal{A}$,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{BLD}}_{\mathsf{BRaccoon},\mathcal{A}}(\lambda) \leq 2\big(&\mathsf{Adv}^{\mathsf{ZK}}_{\pi_1,\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{HID}}_{\mathsf{CS},\mathcal{C}}(\lambda) \\ &+ \mathsf{Adv}^{\mathsf{KS}}_{\pi_2,\mathcal{D}}(\lambda) + \mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{HE},\mathcal{E}}(\lambda)\big) + \epsilon_{\mathsf{Stat}}.\end{aligned}$$

This concludes the proof of blindness.

F Discussion on Table 1

Table 1 combines values taken directly from the literature with a small number of estimates and recomputations. We detail below, for each compared work, the source of the reported data.

[AKSY22]. We consider the "full-fledged blind" variant described in [AKSY22, Appendix A, Fig. 5]. In this variant the public values $\mathbf{A}$ and $\mathsf{PKE.pk}$ are derived from random oracles. For the concrete size estimates, we use the instantiation of [AKSY22, Section 4.4]. The issuing transcript is the exchange of $\mathbf{t}$ and $\mathbf{y}$; concretely, only the first component $\mathbf{y}_1$ of $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2)$ is sent, since $\mathbf{y}_2$ is publicly recoverable from $\mathbf{t}$ and the verification key $\mathsf{vk} = C$. This gives a communication cost of 1.37 KB. The final signature is (π, ct), whose size is estimated as 45.19 KB. Finally, vk is one element of $\mathbb{Z}[X]/(7213, X^{512} + 1)$, so that

$$|\mathsf{vk}| = 512 \cdot 13/8 = 832 \text{ bytes} \approx 0.8 \text{ KB}.$$

The scheme supports an unbounded, i.e., polynomially many, number of signing queries per public key.

[dPK22]. We take the concrete parameters re-evaluated in [HSS24, §4.2]: the original NTRU parameter set was later observed to fall short of the claimed 128-bit security level due to attacks on overstretched-NTRU parameters [ABD16, CJL16], while the updated parameters restore the intended security level. This leaves the pre-signature and final signature sizes unchanged and only mildly increases the communication cost. We also account for the induced increase in the verification key, which consists of three elements of $\mathcal{R}_q$.

[AGJ+24]. For this anonymous-credential system, which refines the approach of [JRS23], we use the size estimates reported in [AGJ+24, Tables 8.1, 8.4–8.6 and Algorithm 6.3]. Since revealing the underlying SEP signature would make presentations linkable to the issuing session, we count the non-interactive credential-showing proof as the blind signature analogue, i.e., as the object given as input to public verification; its size is 79.58 KB. The communication cost counts the issuing/signing transcript exchanged between the user and the issuer, and is estimated as

$$2.38 + 35.99 + 6.81 = 45.18 \text{ KB},$$

where 2.38 KB estimates the commitment by the size of an element of the same type, namely upk; 35.99 KB is the issuance proof size; and 6.81 KB is the size of the emitted SEP signature sent by the issuer in [AGJ+24, Algorithm 6.3].

[BLNS23a]. The relevant data are reported in [BLNS23a, Tab. 3 and Fig. 4]. We estimate the communication cost by counting the full signing transcript exchanged between the user and the signer, namely the user's first message $(\mathbf{c}, \mathsf{ct}, \pi_1)$ and the signer's response $\mathbf{s}$. Thus, the communication cost is

$$|\pi_1| + |\mathsf{ct}| + |\mathbf{c}| + |\mathbf{s}|.$$

For the parameters of [BLNS23a, Tab. 3], with $q = 7933$ and ring degree $d = 512$, an element of $\mathcal{R}_q$ requires approximately $512 \cdot 13/8 \simeq 832$ bytes, i.e., about 0.8 KB. Hence $|\mathbf{c}| \simeq 0.8$ KB, while $\mathbf{s} \in \mathcal{R}_q^2$ contributes about 1.6 KB. The ciphertext size $|\mathsf{ct}|$ depends on the chosen IND-CPA encryption scheme. The bound on the number Q_s of signing queries is governed by [BLNS23a, Thms. 3.2 and 3.3]; in our table, we set $Q_s = 2^{20}$ as a conservative choice. The proof π_1 dominates the transcript. Its size in [BLNS23a] is an analytical estimate that gives a total of about 63 KB; the ZK-LaBRADOR implementation of Biasioli *et al.* [BBL+26] includes this statement as a case study and reports about 110 KB for the proof, and Table 1 reports the measured figure.

[JS25b]. We used the authors' parameter-estimation script[8]. The script confirms the transcript and blind signature sizes reported in [JS25b, Table 5.5], and additionally yields the verification-key size and the maximum number of signatures per public key.

[HKLN20]. We used the signature size from [HKLN20, Fig. 9], the number of signatures per public key specified at the end of [HKLN20, §6], and the sizes computed in [AHJ21, Table 1].

[KLR26]. The sizes are those of the lattice instantiation, computed using a script that the authors kindly provided in a private communication.

[8] https://github.com/truncatedsampler/truncated-sampler/tree/main/scripts

[BMW26]. We use the $n = 1296$ parameter set of [BMW26, Tables 2 and 3]: verification key 4.85 KB, blind signature 4.70 KB and transcript 110 KB, at a reported core-SVP hardness of 116 bits. Since the protocol restarts when the rejection step fails, the number of moves is an expectation: each round costs two moves and the reported expected number of rounds is 1.1 for honest users, giving ≈ 2.2 moves.

We also used [JS25a, Table 5.4] and [KLR26, Table 1] to cross-check some of the entries in our table.

G Detailed Description of L_1 and L_2

We write $\mathsf{vk} = (A_{\mathsf{sig}}, t)$ for the Raccoon verification key and $\mathsf{ek} = (A_{\mathsf{bgv}}, b_{\mathsf{bgv}})$ for the BGV encryption key. The Raccoon, BGV, and extractable-commitment rings use respective moduli

$$q_{\mathsf{sig}}, \qquad Q_{\mathsf{BGV}}, \qquad Q_{\mathsf{ext}}.$$

All their modular equations are lifted into the arithmetic R1CS modulo $M = 2^{256} + 1$. Concretely, an equation $F = 0 \pmod{Q}$ is represented as

$$\mathrm{clift}_Q(F) = Q\kappa,$$

where the signed quotient κ has a bounded binary decomposition. The bounds are chosen so that

$$|\,\mathrm{clift}_Q(F)| + Q|\kappa| < M/2;$$

hence an accepting R1CS assignment implies the intended equality over the integers and therefore modulo Q.

The challenge-generation relation is parameterized by

$$\mathcal{H} = (H_{\mathsf{msg}}, H_{\mathsf{chal}}, \mathsf{XOF}),$$

fixed by the public parameters. We use the 2λ-bit public seed $\mathsf{seed}_{\mathsf{vk}}$ as the circuit representation of the verification key and instantiate H_{msg} and H_{chal} with SHA3-256 and XOF with SHAKE256.

For the extractable BDLOP commitment with public matrices $(A_{\mathsf{ext}}, B_{\mathsf{ext}})$, we write

$$\mathsf{BDLOP.Com}_{\mathsf{ext}}(m; \boldsymbol{\gamma}) = (t_A, t_B) \iff \begin{cases} t_A = A_{\mathsf{ext}}\boldsymbol{\gamma} \pmod{Q_{\mathsf{ext}}}, \\ t_B = B_{\mathsf{ext}}\boldsymbol{\gamma} + m \pmod{Q_{\mathsf{ext}}}. \end{cases}$$

The complete user relation is

$$\mathcal{L}_1 = \left\{ \begin{array}{l} ((t_A^{\mathsf{ext}}, t_B^{\mathsf{ext}}), \mathsf{ctx}_c, w', \mathsf{vk}, \mathsf{ek}, \mathsf{pp}) : \\ \exists\ \mathsf{msg}, \mu, \rho, r, e, w, c, e_w, \boldsymbol{\gamma}_{\mathsf{ext}}, \rho_c, \eta_{c,1}, \eta_{c,2}, \boldsymbol{\kappa}_1 \text{ s.t.} \\ (t_A^{\mathsf{ext}}, t_B^{\mathsf{ext}}) = \mathsf{BDLOP.Com}_{\mathsf{ext}}((\mathsf{msg}, r, e); \boldsymbol{\gamma}_{\mathsf{ext}}), \\ \|\boldsymbol{\gamma}_{\mathsf{ext}}\| \le B_{\boldsymbol{\gamma},\mathsf{ext}}, \\ w' + A_{\mathsf{sig}} r + e = 2^{\nu_w} w + e_w \pmod{q_{\mathsf{sig}}}, \\ \|(r, e)\|_2 \le B_w, \qquad \|e_w\|_\infty < 2^{\nu_w - 1}, \\ \mu = H_{\mathsf{msg}}(\mathsf{seed}_{\mathsf{vk}}, \mathsf{msg}), \\ \rho = H_{\mathsf{chal}}(\mu, w), \\ c = \mathsf{SampleInBall}_{\mathsf{XOF}}(\rho), \\ \mathsf{ctx}_c = (u_c, v_c), \\ u_c = A_{\mathsf{bgv}}^T \rho_c + h\eta_{c,1} \pmod{Q_{\mathsf{BGV}}}, \\ v_c = b_{\mathsf{bgv}}^T \rho_c + h\eta_{c,2} + c \pmod{Q_{\mathsf{BGV}}}, \\ (\rho_c, \eta_{c,1}, \eta_{c,2}) \in \mathcal{B}_{\mathsf{enc}}, \\ \boldsymbol{\kappa}_1 \text{ contains the bounded quotients of all modular equations above.} \end{array} \right\}.$$

The binary hash and sampler constraints and the arithmetic lattice constraints use one common encoding of $(\mathsf{msg}, w, \mu, \rho, c)$; no auxiliary linking commitment is required.

The complete signer relation is

$$\mathcal{L}_2 = \left\{ \begin{array}{l} (\mathsf{ctx}_{\mathbf{z}'}, \mathsf{ctx}_c, w', \mathsf{vk}, \mathsf{ek}, \mathsf{pp}) : \\ \exists\, r', e', s, e_{\mathsf{kg}}, e_t, (u_{r,j}, v_{r,j}, \rho_j, \eta_{j,1}, \eta_{j,2})_{j=1}^{\ell}, \boldsymbol{\kappa}_2 \text{ s.t.} \\ w' = A_{\mathsf{sig}} r' + e' \pmod{q_{\mathsf{sig}}}, \\ \|(r', e')\|_2 \le B_{w'}, \\ A_{\mathsf{sig}} s + e_{\mathsf{kg}} = 2^{\nu_t} t + e_t \pmod{q_{\mathsf{sig}}}, \\ \|e_t\|_\infty < 2^{\nu_t - 1}, \\ u_{r,j} = A_{\mathsf{bgv}}^T \rho_j + h\eta_{j,1} \pmod{Q_{\mathsf{BGV}}}, \\ v_{r,j} = b_{\mathsf{bgv}}^T \rho_j + h\eta_{j,2} + r'_j \pmod{Q_{\mathsf{BGV}}}, \\ (\rho_j, \eta_{j,1}, \eta_{j,2}) \in \mathcal{B}_{\mathsf{fl}}, \\ \mathsf{ctx}_{\mathbf{z}'} = (\mathsf{ctx}_{\mathbf{z}'}^{(1)}, \ldots, \mathsf{ctx}_{\mathbf{z}'}^{(\ell)}), \\ \mathsf{ctx}_{\mathbf{z}'}^{(j)} = s_j \cdot \mathsf{ctx}_c + (u_{r,j}, v_{r,j}) \pmod{Q_{\mathsf{BGV}}}, \qquad j \in [\ell], \\ \boldsymbol{\kappa}_2 \text{ contains the bounded quotients of all modular equations above.} \end{array} \right\}.$$

Both languages are proved with one execution of the zero-knowledge LaBRADOR toolkit: π_1 proves L_1 and π_2 proves L_2.

H BRaccoon Relations via Mixed ZK-LaBRADOR

This section describes the mixed-R1CS encoding of the complete user relation L_1. It combines the lattice equations, SHA3-256, SHAKE256, and the challenge sampler in one succinct lattice-based zero-knowledge proof.

The construction combines two types of constraints. SHA3, SHAKE256, and the challenge sampler are expressed as a binary R1CS over $\mathbb{F}_2$, whereas the Raccoon, BGV, and extractable-commitment equations are lifted to bounded integer equations and embedded in an arithmetic R1CS. The two systems share a common binary encoding of their witnesses and are reduced to one dot-product relation using the mixed-R1CS reduction of LaBRADOR [BS23]. We instantiate the resulting relation with the zero-knowledge LaBRADOR toolkit of Biasioli *et al.* [BBL+26].

The construction proves

$$\mu = \mathsf{SHA3\text{-}256}(\mathsf{seed}_{\mathsf{vk}} \parallel \mathsf{Encode}(\mathsf{msg}))$$

and

$$\rho = \mathsf{SHA3\text{-}256}(\mu \parallel \mathsf{Encode}_{\mathbf{w}}(\mathbf{w})),$$

and derives the challenge by expanding ρ with SHAKE256 and applying the truncated Fisher–Yates sampler:

$$c = \mathsf{SampleInBall}(\rho).$$

H.1 The mixed relation and integer lifting

The complete relation L_1 is the one given in Appendix G. Its binary witness contains the encodings of $\mathsf{msg}, \mathbf{r}, \mathbf{e}, \mathbf{w}, \mu, \rho, c$, the extractable-commitment opening, the BGV randomness and noise, and all intermediate SHA3, SHAKE256, and sampler values. The same bits encode the bounded integers used by the arithmetic constraints; hence no commitment is needed to link separate proof systems.

The source relations use three moduli:

$$q_{\mathsf{sig}}, \qquad Q_{\mathsf{BGV}}, \qquad Q_{\mathsf{ext}}.$$

For any coefficient equation $F = 0 \pmod{Q}$ over one of these rings, let $\mathrm{clift}_Q(F)$ denote its centered integer lift. The arithmetic R1CS proves

$$\mathrm{clift}_Q(F) = Q\kappa \tag{12}$$

for a signed quotient κ represented by a bounded binary decomposition. If $|\operatorname{clift}_Q(F)| \le B_F$, we use

$$B_\kappa = \left\lceil \frac{B_F}{Q} \right\rceil, \qquad \ell_\kappa = \lceil \log_2(2B_\kappa + 1) \rceil .$$

All lifted equations are embedded in the common arithmetic R1CS modulo

$$M = 2^{256} + 1.$$

The parameter bounds are chosen so that

$$B_F + QB_\kappa < \frac{M}{2}. \tag{13}$$

Consequently, an equality accepted modulo M is the intended equality over the integers, and reducing Equation (12) modulo Q recovers the original source relation. The signer relation L_2 is encoded in the same way, but has no hash or sampler constraints.

H.2 Binary R1CS for SHA3

We instantiate both hashes with SHA3-256 as standardized in FIPS 202 [Nat15]. SHA3-256 uses the permutation Keccak-$f[1600]$, whose state consists of 1600 bits and whose permutation has 24 rounds. Each round applies the five steps θ, ρ, π, χ, and ι. The first, second, third, and fifth steps are affine over $\mathbb{F}_2$. The only nonlinear step is χ.

For state bits $a, b, c \in \mathbb{F}_2$, one output bit of χ has the form

$$y = a \oplus ((\neg b) \wedge c).$$

Over $\mathbb{F}_2$, this is equivalently written as

$$b\,c = y + a + c.$$

Thus one binary rank-1 constraint suffices for each output bit of χ. The affine steps before and after χ are absorbed into the linear forms of the R1CS matrices. It follows that one Keccak-$f[1600]$ permutation requires

$$N_{\mathsf{Keccak}} = 24 \cdot 1600 = 38\,400 \tag{14}$$

quadratic constraints, apart from linear input/output consistency equations.

SHA3-256 has rate $r_{\mathsf{SHA3}} = 1088$ bits and capacity 512 bits. Let $P_{256}(L)$ denote the number of padded 1088-bit blocks required for an L-bit input under the SHA3 domain suffix and the pad-10*1 rule. The number of nonlinear constraints for an L-bit SHA3-256 input is therefore

$$N_{\mathsf{SHA3}}(L) = 38\,400\, P_{256}(L). \tag{15}$$

Absorption, padding, the standard SHA3 suffix, and digest extraction are affine over $\mathbb{F}_2$ once the message length is fixed.

The protocol message is one polynomial $\mathsf{msg} = \sum_{i=0}^{n-1} \mathsf{msg}_i X^i$ whose coefficients satisfy $\mathsf{msg}_i \in \{0, 1\}$. Hence, for $n = 256$, its canonical representation contains exactly 256 bits. Since $\mathsf{seed}_{\mathsf{vk}}$ and μ also have $2\lambda = 256$ bits, the two input lengths are

$$L_{\mathsf{dig}} = 256 + 256 = 512,$$
$$L_{\mathsf{chal}} = 256 + L_{\mathbf{w}}.$$

The rounded vector $\mathbf{w} \in R_{q_{\mathbf{w}}}^k$ is serialized coefficient-wise in canonical form. Writing

$$b_{\mathbf{w}} = \lceil \log_2 q_{\mathbf{w}} \rceil, \qquad q_{\mathbf{w}} \simeq \left\lceil \frac{q_{\mathsf{sig}}}{2^{\nu_{\mathbf{w}}}} \right\rceil,$$

we obtain

$$L_{\mathbf{w}} = knb_{\mathbf{w}}.$$

For our concrete parameters $k = 12$, $n = 256$, $\log_2 q_{\mathsf{sig}} = 66$, and $\nu_{\mathbf{w}} = 55$, we have $b_{\mathbf{w}} = 11$ and hence

$$L_{\mathbf{w}} = 12 \cdot 256 \cdot 11 = 33\,792 \text{ bits}.$$

The first hash requires one Keccak permutation and the second requires 32 permutations, including SHA3 padding. The two SHA3 computations therefore use 33 Keccak permutations and

$$N_{\mathsf{hash}}^{\mathsf{quad}} = 33 \cdot 38\,400 = 1\,267\,200 \approx 2^{20.27}$$

quadratic constraints.

This count shows that the serialization of $\mathbf{w}$, rather than the digest of the message, dominates the Boolean circuit. It also shows why the use of a succinct proof backend is important.

H.3 Zero-knowledge proof backend

The mixed-R1CS reduction of LaBRADOR [BS23] uses a common binary witness that both satisfies the binary R1CS and encodes the arithmetic witness modulo $2^{256} + 1$. Both systems are reduced to the same dot-product relation and proved by one execution of the main protocol.

We use the zero-knowledge LaBRADOR toolkit of Biasioli *et al.* [BBL+26]; its hiding and binding commitments rely on the corresponding Module-LWE and Module-SIS assumptions.

The reported experiments of Biasioli *et al.* produce proofs of approximately 110 KB across their zero-knowledge case studies. We use this value for each of π_1 and π_2 in Section 5.4.

H.4 Binary encoding of the challenge sampler

We instantiate the XOF used by SampleInBall with SHAKE256. Let $\kappa_{\mathsf{idx}} = 160$. The circuit expands ρ into

$$\omega\kappa_{\mathsf{idx}} + \omega$$

bits. The first $\omega\kappa_{\mathsf{idx}}$ bits are parsed as ω integers $x_1, \ldots, x_\omega \in [0, 2^{\kappa_{\mathsf{idx}}})$, and the remaining ω bits are the signs $\mathsf{sg}_1, \ldots, \mathsf{sg}_\omega$.

For $a \in \{1, \ldots, \omega\}$, set $i_a = n - \omega + a - 1$ and $m_a = i_a + 1$. The witness contains an integer quotient q_a and an index j_a satisfying

$$x_a = m_a q_a + j_a, \qquad 0 \le j_a < m_a.$$

The range of j_a is encoded by selector bits $\delta_{a,0}, \ldots, \delta_{a,m_a-1}$ satisfying

$$\sum_{u=0}^{m_a-1} \delta_{a,u} = 1, \qquad j_a = \sum_{u=0}^{m_a-1} u\delta_{a,u}.$$

The quotient equation, the sum-to-one constraint, and the reconstruction of j_a are linear in the arithmetic R1CS once the shared bits are interpreted as bounded integers. We bound

$$0 \le q_a \le \left\lfloor \frac{2^{\kappa_{\mathsf{idx}}} - 1}{m_a} \right\rfloor,$$

so the absolute value of the lifted quotient expression is less than $2^{\kappa_{\mathsf{idx}}+1} + m_a$. The common arithmetic modulus M is selected so that

$$2^{\kappa_{\mathsf{idx}}+1} + m_a < \frac{M}{2}$$

for every a, in addition to the modular no-wrap conditions in Equation (13).

The circuit then verifies the truncated Fisher–Yates update. The selector bits identify the entry j_a, and the circuit performs the simultaneous assignments $C_{i_a} \leftarrow C_{j_a}$ and $C_{j_a} \leftarrow (-1)^{\mathsf{sg}_a}$. If $j_a = i_a$, the second assignment determines the final value of C_{i_a}. After ω iterations, the resulting polynomial has exactly ω nonzero coefficients in $\{-1, 1\}$.

The circuit does not enforce the rejection conditions used to remove the negligible bias of the modular reductions. If the prescribed public SampleInBall algorithm aborts, the resulting signature does not verify; omitting these comparisons therefore does not help produce a valid forgery.

For $n = 256$, $\omega = 23$, and $\kappa_{\mathsf{idx}} = 160$, SHAKE256 must produce

$$23 \cdot 160 + 23 = 3\,703$$

bits. SHAKE256 has rate 1088, and therefore requires four Keccak-f[1600] permutations. Its nonlinear cost is

$$N^{\mathsf{quad}}_{\mathsf{SHAKE}} = 4 \cdot 38\,400 = 153\,600 \approx 2^{17.23}.$$

Using a conservative selector-and-swap accounting, the Fisher–Yates updates and sign assignments require at most

$$N^{\mathsf{quad}}_{\mathsf{FY}} = \omega(3n + 4) = 17\,756$$

additional quadratic constraints. Some selector-booleanity constraints are native in the binary R1CS and may disappear in a concrete implementation, so this is an upper bound.

H.5 Total constraint count

Excluding the lifted linear BDLOP equations, quotient reconstructions, and other linear consistency constraints, the complete binary computation uses at most

$$\begin{aligned} N^{\mathsf{quad}}_{\mathsf{L}_1} &= N^{\mathsf{quad}}_{\mathsf{hash}} + N^{\mathsf{quad}}_{\mathsf{SHAKE}} + N^{\mathsf{quad}}_{\mathsf{FY}} \\ &= 1\,267\,200 + 153\,600 + 17\,756 \\ &= 1\,438\,556 \approx 2^{20.46}. \end{aligned}$$

The arithmetic part additionally contains the lifted Raccoon and BGV equations, the extractable-commitment equations, the 3ω linear index and selector equations described above, and the binary reconstructions of their bounded quotients. These linear constraints do not change the nonlinear count above.

I SampleInBall Procedure from [PWFW23]

The SampleInBall procedure deterministically maps a seed ρ to a challenge vector $C \in \{-1, 0, 1\}^n \subset \mathbb{F}_q^n$ containing exactly ω non-zero entries. Conditioned on no abort, its output is uniform over all such vectors.

We set $\kappa_{\mathsf{idx}} = 160$ and expand ρ with SHAKE256 into $\omega\kappa_{\mathsf{idx}} + \omega$ bits. The first $\omega\kappa_{\mathsf{idx}}$ bits are parsed as integers $x_1, \ldots, x_\omega \in [0, 2^{\kappa_{\mathsf{idx}}})$; the remaining bits are the signs $\mathsf{sg}_1, \ldots, \mathsf{sg}_\omega$.

The two assignments are simultaneous; if $j = i$, the second assignment determines the final value of C_i. At each iteration, rejection sampling makes j uniform in $\{0, \ldots, i\}$. Hence the procedure implements a truncated inside-out Fisher–Yates shuffle and produces uniform locations and signs.

The abort probability at one iteration is less than $m/2^{\kappa_{\mathsf{idx}}} \leq n/2^{\kappa_{\mathsf{idx}}}$. By a union bound, for $n = 256$, $\omega = 23$, and $\kappa_{\mathsf{idx}} = 160$, the overall abort probability is less than

$$\frac{\sum_{m=234}^{256} m}{2^{160}} = \frac{5635}{2^{160}} < 2^{-147}.$$

Algorithm 1 SampleInBall using SHAKE256

function SampleInBall(ρ)
  $C \leftarrow 0 \in \{-1, 0, 1\}^n \subset \mathbb{F}_q^n$
  $(x_1, \ldots, x_\omega, \mathsf{sg}_1, \ldots, \mathsf{sg}_\omega) \leftarrow \mathsf{Parse}(\mathsf{SHAKE256}(\rho), \omega\kappa_{\mathsf{idx}} + \omega)$
  for $a \leftarrow 1$ **to** ω **do**
    $i \leftarrow n - \omega + a - 1$
    $m \leftarrow i + 1$
    $T \leftarrow 2^{\kappa_{\mathsf{idx}}} - (2^{\kappa_{\mathsf{idx}}} \bmod m)$
    if $x_a \geq T$ **then**
      Abort
    end if
    $j \leftarrow x_a \bmod m$
    $C_i \leftarrow C_j$
    $C_j \leftarrow (-1)^{\mathsf{sg}_a}$
  end for
  return C
end function

J Methodology for the Instantiation

Bounding methodology. All bounds used in the proof statements are chosen so that the relations can be checked by the underlying lattice proof system while preserving correctness of the protocol computations. The notation follows Appendix G: ρ denotes a number of ring elements, and B_{bin} and B_{gauss} below are the ℓ_2-bounds used in the tables of Section 5.4. We distinguish two kinds of bounds. First, global ℓ_2-bounds are used for short module vectors sampled from binary or Gaussian-like distributions, such as signing randomness, blinding randomness, and BDLOP openings. These bounds are used in the mixed ZK-LaBRADOR statements and in the corresponding SIS/MLWE parameter selection. Second, coefficient-wise ℓ_∞-bounds are used whenever correctness depends on the absence of modular wrap-around, for example in rounding errors, plaintext decoding, extractable commitments, and BGV decryption.

For a vector of ρ ring elements over $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$, binary randomness is bounded by

$$B_{\mathsf{bin}}(\rho) = \sqrt{n\rho},$$

while Gaussian randomness with width σ is bounded by

$$B_{\mathsf{gauss}}(\rho, \sigma) = e^{1/4}\sigma\sqrt{n\rho}.$$

When a binding argument compares two valid openings, the relevant SIS witness is their difference. Thus, for Gaussian openings we use the conservative bound $2B_{\mathsf{gauss}}$, whereas for binary openings we use the binary bound implemented in our parameter search.

The user computes

$$\mu = \mathsf{H}(\mathsf{seed}_{\mathsf{vk}}, \mathsf{msg}),$$

where $\mathsf{seed}_{\mathsf{vk}}$ is a public 2λ-bit seed representing the verification key inside the circuit, and μ is a 2λ-bit digest. In the first-round extractable commitment, the user commits to

$$m_0 = (\mathsf{msg}, r, e),$$

The challenge-generation proof checks

$$\mu = H_{\mathsf{msg}}(\mathsf{seed}_{\mathsf{vk}}, \mathsf{msg}), \qquad \rho = H_{\mathsf{chal}}(\mu, \mathbf{w}), \qquad c = \mathsf{SampleInBall}_{\mathsf{XOF}}(\rho).$$

The concrete profile instantiates these functions with SHA3-256 and SHAKE256. The binary and arithmetic constraints use a common binary encoding of these values inside one mixed proof.

Rounding relations are rewritten linearly by introducing explicit rounding-error variables. For example,

$$\mathbf{w} = \lfloor \mathbf{w}' + A_{\mathsf{sig}} r + e \rceil_{\nu_{\mathbf{w}}}$$

is represented as

$$\mathbf{w}' + A_{\mathsf{sig}} r + e = 2^{\nu_{\mathbf{w}}} \mathbf{w} + e_{\mathbf{w}} \pmod{q_{\mathsf{sig}}}, \qquad \|e_{\mathbf{w}}\|_\infty < 2^{\nu_{\mathbf{w}}-1}.$$

The public-key rounding relation is handled analogously:

$$A_{\mathsf{sig}} s + e_{\mathsf{kg}} = 2^{\nu_{\mathbf{t}}} \mathbf{t} + e_{\mathbf{t}} \pmod{q_{\mathsf{sig}}}, \qquad \|e_{\mathbf{t}}\|_\infty < 2^{\nu_{\mathbf{t}}-1}.$$

These equations are lifted into the arithmetic R1CS used by ZK-LaBRADOR.

For the BGV component, the plaintext bound is derived from the value recovered after decryption. In our protocol this value is

$$z' = c \cdot s + r',$$

so we use

$$B_{\mathsf{plain}} = B_c B_s + B_{r'},$$

where $B_c = \|c\|_1 = \omega$, B_s bounds the signing-key component, and $B_{r'}$ bounds the signer randomness. The plaintext scale h is then chosen so that

$$h > 2B_{\mathsf{plain}},$$

ensuring that the plaintext does not wrap modulo h. The ciphertext modulus is chosen so that the BGV decryption residual does not wrap:

$$B_{\mathsf{plain}} + hB_{\mathsf{final}} < Q_{\mathsf{BGV}}/2,$$

where B_{final} includes the evaluated encryption noise and the flooding noise.

We use separate moduli for the three algebraic components:

$$q_{\mathsf{sig}}, \qquad Q_{\mathsf{BGV}}, \qquad Q_{\mathsf{ext}}.$$

A coefficient equation $F = 0 \pmod{Q}$ over any of these rings is represented in the arithmetic R1CS as

$$\mathrm{clift}_Q(F) = Q\kappa,$$

with a bounded signed quotient κ. All quotient bits are included in the common binary witness. The arithmetic modulus is $M = 2^{256} + 1$, and the coefficient bounds are chosen so that

$$|\,\mathrm{clift}_Q(F)| + Q|\kappa| < M/2.$$

Table 5. Bounds used in the lattice relations and BDLOP commitments.

Symbol	Meaning	Bound / condition
B_t	Bound for the signing-key opening (s, e_{kg})	$B_t = e^{1/4}\sigma_t\sqrt{n(k+\ell)}$
$B_{w'}$	Bound for the signer first-round randomness (r', e')	$B_{w'} = e^{1/4}\sigma_{w'}\sqrt{n(k+\ell)}$
B_w	Bound for the user blinding randomness (r, e)	$B_w = e^{1/4}\sigma_w\sqrt{n(k+\ell)}$
$B_{e_{\mathbf{w}}}$	Infinity-norm bound for the commitment-rounding error	$\|e_{\mathbf{w}}\|_\infty < 2^{\nu_{\mathbf{w}}-1}$
$B_{e_{\mathbf{t}}}$	Infinity-norm bound for the public-key rounding error	$\|e_{\mathbf{t}}\|_\infty < 2^{\nu_{\mathbf{t}}-1}$
$B_2^\star$	Final Raccoon verification bound	$\|(z, 2^{\nu_{\mathbf{w}}}h)\|_2 \le B_2^\star$
$B_{\boldsymbol{\gamma},\mathsf{ext}}$	Extractable BDLOP opening bound	Binary: $\sqrt{n\rho_i}$, Gaussian: $e^{1/4}\sigma_{\boldsymbol{\gamma}}\sqrt{n\rho_i}$
$B_{\Delta\boldsymbol{\gamma},\mathsf{ext}}$	Extractable BDLOP binding/MSIS difference bound	Binary: $B_{\boldsymbol{\gamma},i}$, Gaussian: $2B_{\boldsymbol{\gamma},i}$
B_{msg}	Coefficient bound for the external message representation	In our R1CS estimate, msg is represented by 256 field elements
B_μ	Coefficient bound for the digest μ	μ is a 2λ-bit string encoded over the proof field
B_{m_0}	Coefficient bound for the extractable round-1 committed message	For $m_0 = (\mathsf{msg}, r, e)$, conservatively $B_{m_0} = \max\{B_{\mathsf{msg}}, B_\mu, B_w\} = B_w$
$B_{E\boldsymbol{\gamma}}$	Extraction noise in the simulated BDLOP setup	$B_{E\boldsymbol{\gamma}} = B_E^{\mathsf{rpw}} B_{\boldsymbol{\gamma},\mathsf{ext}}$
h_{ext}	Plaintext scale for extractable BDLOP	$h_{\mathsf{ext}} > 2B_{m_0}$
Q_{ext}	Extractable-commitment modulus	$B_{m_0} + h_{\mathsf{ext}}B_{E\boldsymbol{\gamma}} < Q_{\mathsf{ext}}/2$

Table 6. Bounds used for the MLWE-BGV encryption relations.

Symbol	Meaning	Bound / condition
B_c	Challenge bound	$B_c = \omega$, where $\omega = \|c\|_1$
B_s	Plaintext-side signing-key bound	Usually $B_s = B_t$
$B_{r'}$	Plaintext-side signer randomness bound	Usually $B_{r'} = B_{w'}$
B_{plain}	Bound on decrypted plaintext $c \cdot s + r'$	$B_{\mathsf{plain}} = B_c B_s + B_{r'}$
h	BGV plaintext scale	$h > 2B_{\mathsf{plain}}$
B_{enc}	Fresh BGV encryption noise	$B_{\mathsf{enc}} = B_{e_{\mathsf{pk}}}B_\rho + B_{\eta_2} + B_{s_{\mathsf{he}}}B_{\eta_1}$
B_{eval}	Noise after evaluating $f_s(C, M) = s \odot C + M$	$B_{\mathsf{eval}} = B_s B_{\mathsf{in}} + B_{\mathsf{add}}$
B_{fl}	Noise-flooding bound	Chosen so that flooding statistically hides evaluation noise
B_{final}	Final BGV decryption-noise bound	$B_{\mathsf{final}} = B_{\mathsf{eval}} + B_{\mathsf{fl}}$
Q_{BGV}	BGV ciphertext modulus	$B_{\mathsf{plain}} + hB_{\mathsf{final}} < Q_{\mathsf{BGV}}/2$