"Preview Writeup": In anticipation of a package submission to the NIST Threshold Call

Title: Amber: a Family of Efficient Lattice-Based IND-CCA Threshold KEMs

Version: 0.1 (2026-01-20)[1]

Team name: Amber Team

Team members: Katharina Boudgoust, Rafael del Pino, Oleksandra Lapiha, Thomas Prest

Abstract: This document previews Amber, our submission to the NIST MPTC call. Amber is a threshold key encapsulation mechanism (TKEM) whose security relies on standard structured lattice assumptions. More precisely, the decapsulation key is shared among N parties, and any subset of *at least* T-out-of-N decapsulators can recover a message that was encapsulated to the encapsulation key. A prominent feature of Amber is that it natively achieves IND-CCA security (indistinguishability against chosen-ciphertext attacks) using only relatively simple techniques.

At the technical level, Amber is based on the BCHK+ transform, an adaptation of the BCHK (Boneh-Canetti-Halevi-Katz) transform, well suited for the lattice context. The BCHK+ framework is very modular and allows some flexibility in terms of the underlying building blocks. We reflect this in our document by breaking down our scheme in distinct building blocks: (i) a secret sharing scheme, or SSS, (ii) a (threshold) identity-based encryption scheme, or (T)IBE, and (iii) a one-time signature scheme, or OTS.

Proposed crypto-systems: Amber: a Family of Efficient Lattice-Based IND-CCA Threshold KEMs. **Category:** S2 (PKE).

Keywords: Threshold Cryptography; NIST Threshold Call; BCHK+ Transform; IND-CCA; Lattice Cryptography

(Logo AI-generated with ChatGPT)

[1]Preliminary version submitted to NIST-MPTC for review

Preview writeup. This document is provided to NIST for online publication, to foster public awareness and support public discussion within the scope of the NIST First Call for Multi-Party Threshold Schemes [NIST-IR8214C]. This "preview writeup" represents a good-faith plan for a subsequent "package submission". However, until the deadline for package submission, the team may still modify its own composition and the submission plan, including possible changes to the technical scope, and/or the used techniques or achieved results.

Team members: Katharina Boudgoust [i1,a3], Rafael del Pino [i2,a1], Oleksandra Lapiha [i3,a2], Thomas Prest [i4,a1]

Open Researcher and Contributor Identifiers (ORCID):

i1 (0000-0002-3971-9368); i2 (0009-0001-8638-787X); i3 (0009-0001-5089-989X); i4 (0000-0003-1445-6212)

Affiliations:

[a1] PQShield @ Paris, France

[a2] Royal Holloway University London @ London, United Kingdom

[a3] CNRS, Univ Montpellier, LIRMM @ Montpellier, France

Main contacts:

- **Team mailing list:**

  amber-tkem @ googlegroups com

- **Primary technical contact person:**

  Oleksandra Lapiha – sasha lapiha 2021 @ live rhul ac uk

- **Secondary contact persons:**

  Katharina Boudgoust – katharina boudgoust @ lirmm fr

  Rafael del Pino – rafael del pino @ pqshield com

  Thomas Prest – thomas prest @ pqshield com

Produced by humans. The team hereby confirms that the content in this preview writeup: (i) was produced by the team members, and (ii) was not produced by generative artificial intelligence (AI), with the possible exception of AI-proposed grammar improvements, minor integrated suggestions, or some well-identified and short localized portions of auxiliary content (e.g., some illustration); and (iii) was proofread by the team members.

1. Introduction

This preview writeup introduces Amber, a threshold key encapsulation mechanism (TKEM) whose security is based on standard lattice problems.

The design template of Amber is based on the BCHK+ transform introduced in [LP25] and well suited for the lattice context. Several optimization strategies as proposed in [KLPP26] helped achieving a practically efficient scheme.

Our main design goal was to achieve security against *active* adversaries, while avoiding heavy tools such as non-interactive zero-knowledge proofs or generic multi-party computations. The BCHK+ is a very modular and flexible tool to achieve this goal. As will be detailed below, it is obtained from several building blocks whose concrete instantiations can be done in a modular way.

We encounter a trade-off between robustness guarantees and security against adaptive corruptions, which we see as our main technical limitation. However, this trade-off reflects a more general problem in lattice-based threshold cryptography, and does not seem to depend on the BCHK+ transform itself.

2. Specification

2.1. Organization

This specification proposes two schemes for the Category S2 of threshold schemes for primitives of regular public-key encryption scehmes (PKE) and key-encapsulation mechanisms (KEM) that are not standardized by NIST. The two schemes are variants of the same underlying framework but exhibit markedly different trade-offs in terms of efficiency, scalability, and security. We aim to identify the stronger candidate through interaction and feedback from the community. We differentiate them based on the underlying secret sharing scheme: Shamir or Vandermonde.

Shamir. As summarized in Table 1, the first scheme achieves high scalability (values of $N = 1024$ or higher are easily achieved) at the cost of robustness; however, it does provide adaptive security.

Vandermonde. In contrast, the second scheme sacrifices scalability (for $N > 64$ the number of secret shares becomes prohibitively large) in exchange for strong robustness guarantees: malicious aborts are detected, and in the absence of aborts, decryption is correct with overwhelming probability. It is worth noting that for small sets this scheme is computationally more efficient.

Table 1: Trade-offs between our two schemes. Computation for (I) is technically $O(T)$, only because the identity of all other parties has to be read.

Variant	Security	Robustness	Group size	Computation per party
I (Shamir)	Adaptive	No	Enormous	$O(1)$*
II (Vandermonde)	Selective	Yes	Medium	$O(T)$

The specification document will be organized as follows:

- It will first introduce the syntax and definitions of the required cryptographic primitives and then give concrete instantiations of them.
- When doing so, we will give intuitions on the choices underlying our design. Various tweaks, such as balanced noise flooding, and approximate calculations, allowed us to significantly improve on the ciphertext size.

Figure 1 gives an overview of our construction template and further information is given in the sections below.

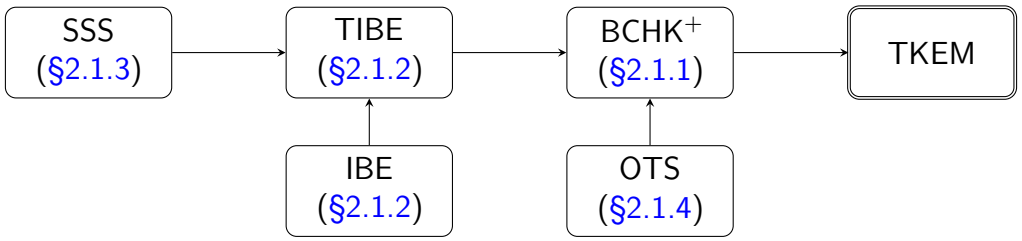

Figure 1: Structure of our TKEM construction

We emphasize the following points:

- The cryptosystem we submit is a threshold key encapsulation mechanism (TKEM). The other families of functions presented in this document (SSS, IBE, TIBE, OTS and BCHK$^+$) are building blocks that are assembled in a modular manner to construct our TKEM.
- Our TKEM comes in two main variations: Shamir and Vandermonde. These correspond to the underlying choice of SSS (§2.1.3), and this choice impacts several fundamental properties of the scheme: implementation, efficiency and security features.

2.1.1. The BCHK$^+$ transform

The framework that underlies our entire construction is the BCHK$^+$ transform [LP25], an adaptation of the BCHK transform [CHK04; BCHK07] to the lattice setting. The BCHK$^+$ transform is a generic transform that takes as input a one-time signature scheme OTS and a threshold identity-based encryption scheme TIBE, and outputs a threshold key encapsulation mechanism TKEM. In short,

$$\text{BCHK}^+(\text{OTS}, \text{TIBE}) = \text{TKEM}$$

Similarly to the Fujisaki-Okamoto (FO) transform, the BCHK$^+$ transform only requires mild properties from the OTS and the TIBE (in particular, TIBE must only satisfy a variation of IND-CPA security), and outputs a TKEM that satisfies the stronger notion of IND-CCA security. However, thresholdizing the FO transform requires the distribution of random oracles, whereas thresholdizing the BCHK$^+$ transform only requires the distribution of TIBE operations.

Note that, as the FO transform, the BCHK$^+$ transform contains a decrypt-and-re-encrypt step. In contrast to the FO, this only happens at the time of combining decapsulation shares together, a step that is public and does not require thresholdization. This makes the BCHK$^+$ transform a compelling method to construct an efficiently scalable and actively secure TKEM.

2.1.2. The underlying (Threshold) IBE

The main ingredient required by the BCHK$^+$ transform is a (threshold) identity-based encryption scheme, or (T)IBE. In our current proposal, the choice of TIBE accounts for 100% of the encapsulation key size, and between 88% and 97% of the ciphertext size.

We start from the "ROHIBE" (non-threshold) lattice-based IBE from Cash et al. [CHKP10], which we tweak in two ways: (i) we use Eagle trapdoors [YJW23; EENPSS24], (ii) for trapdoor sampling, we replace GPV sampling [GPV08] with noise flooding à la Plover/Raccoon [EENPSS24; PKPR24]. These two tweaks give a linear structure to the main IBE operations, making it straightforward to thresholdize. The actual thresholdization of the IBE is performed using a (linear) secret sharing scheme SSS. In short,

TIBE =	ROHIBE	[CHKP10]
+	Eagle trapdoors	[YJW23; EENPSS24]
+	Noise flooding	[EENPSS24; PKPR24]
+	SSS	(§2.1.3)

2.1.3. The underlying SSS

The TIBE construction outlined in (§2.1.2) requires a threshold-friendly IBE and a (linear) secret sharing scheme, or SSS. In our case, there are two main contenders:

- **Shamir secret sharing [Sha79].** Based on Lagrange interpolation, this SSS has excellent efficiency properties. In particular, it scales well with large numbers of parties. To be used securely in the lattice setting, it requires the online computation of ephemeral "zero-shares" [DKMMPS24; KRT24]. Zero-shares represent an efficiency bottleneck during the decapsulation procedure, since each party needs to perform $O(T)$ calls to a PRF (for example SHAKE). On the other hand, they can provide adaptive security at a moderate cost [KRT24].
- **Vandermonde secret sharing [DDB95; BCPENP25].** The second choice is to pick a secret sharing with "shortness" properties: the individual shares and reconstruction coefficients should be short. This enables the security proof to go through without resorting to zero-shares. The absence of zero-shares has two main consequences: (i) partial decryption shares can be publicly verified to be valid, which provides a form of robustness, (ii) adaptive security now requires a complexity leveraging argument, which entails a gap $\binom{N}{T}$ between selective and adaptive security.

  There are several possible "short" SSS – for example replicated secret sharing can be used, but it scales poorly with up to $\binom{N}{T-1}$ shares per party. We choose instead to use a

SSS based on the Vandermonde identity [DDB95; BCPENP25], which scales only slightly superpolynomially in N. Realistically, it can support $N \leq 64$ [BCPENP25, Figure 2].

2.1.4. The underlying OTS

A minor ingredient of the BCHK$^+$ transform is a one-time signature scheme OTS. A standard signature scheme is also an OTS, therefore it is a viable option to pick ML-DSA, SLH-DSA or FN-DSA as an OTS. Alternatively, WOTS+ or a one-time variant of Raccoon would also be viable options. Note that the OTS key generation and signing are performed by the encapsulation procedure, while verification is performed (in the clear) during the decapsulation procedure.

2.1.5. The resulting TKEM

The syntax of our resulting KEM is given in Fig. 2. All parties share the same encapsulation key ek, each party $i \subset \{1, \ldots, N\}$ is given a share dk_i of the decapsulation key.

Any third party can encapsulate a key to ek. A subset act $\subset \{1, \ldots, N\}$ of at least T decapsulators may participate in an interactive decapsulation protocol in order to recover a key', that will satisfy key $=$ key' if all parties follow the protocol.

The decapsulation protocol may fail if one member or more of act does not follow the protocol; in this case, for the Vandermonde variant of our scheme, it also outputs a non-empty set act$' \subseteq$ act of parties that did not follow the protocol.

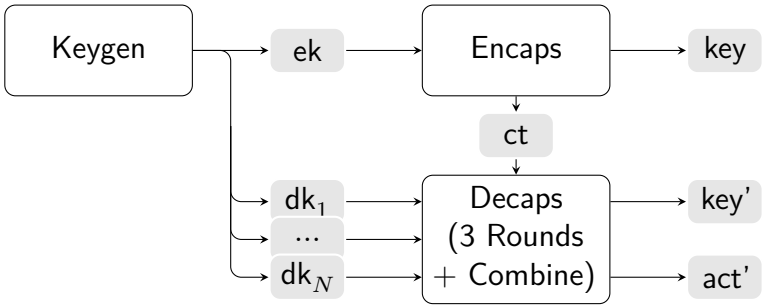

Figure 2: Structure of our TKEM construction

2.2. System and Security Models

2.2.1. System model

We consider a system with a trusted setup for the key generation. We also assume the strongest network model with synchrony, broadcast and guaranteed message delivery. We conjecture that our construction may be secure in a weaker model. We also assume a trusted setup, although it seems plausible that recent works on distributed key generation (DKG) [ENP24; BCPENP25]

could be adapted to our setting. We believe these two topics (weaker communication models, and DKG) are promising research directions.

The protocol supports any threshold $T \leq N$, hence allows a dishonest majority. In order to decapsulate a key any group of T parties executes an interactive protocol to compute their final decryption shares. In the end of the protocol, the decryption shares are either sent to all parties or to a dedicated (public) decryptor. Given the shares, one runs a public function to decapsulate the final key.

In this specification, we present two variants (a Shamir-based and a Vandermonde-based protocol) of our threshold KEM that satisfy different security properties. The Shamir-based variant can easily cover up to $N \leq 1024$ parties. The Vandermonde-based variant can only cover up to $N \leq 64$ parties.

2.2.2. Security

Both protocols we present are secure against an active adversary in the game-based model (i.e. CCA2 security for TKEM). We also prove that an active adversary cannot cause inconsistent behaviour of the system, the property that appears in the literature as Decryption Consistency. Informally, it states that the same ciphertext always either fails or decapsulates to the same key.

The corruption model differs between the two variants. For the Vandermonde-based scheme, we consider a model with static corruptions. Here the adversary declares the set of $T-1$ corrupt parties before receiving the public setup information from the challenger and the start of the decapsulation query phase. For small thresholds, we can achieve security with adaptive corruptions using standard complexity leveraging. However, for larger thresholds the security loss is probihitive.

The Shamir-based system natively allows for adaptive corruptions. Thus, the adversary can request key material of a total of $T-1$ parties, stretched over the security experiment.

When using the Vandermonde secret sharing, the (intermediate) TIBE fulfils a stronger security property which we term extraction share robustness. Even if the adversary maliciously generates the key extraction shares, they can not cause the reconstruction of honestly generated ciphertexts to fail, without being caught. If any share is deemed invalid the corresponding party is flagged, removed and the session restarts from the beginning. This robust TIBE then further leads to a TKEM with improved security guarantees. Its Shamir-based counterpart does not satisfy those stronger security property, as the one-time masks make individual share verification difficult.

Lastly, assuming honest protocol execution, both schemes decapsulate a given key correctly with overwhelming probability over the encryption and decryption randomness. When the decryption fails due to a malicious or corrupted ciphertext the KEM protocol aborts explicitly returning a special symbol.

The overall security of both our schemes relies on the hardness of the Ring Learning With Errors (RLWE) assumption [LPR10] in the Random Oracle model. The security proof actually relies on the intermediate assumption of RLWE with (coset) hints, but the latter can be reduced from standard RLWE [KLSS23; EENPSS24; LP25].

3. Open-Source Implementation

As our submission is based on a very recent line of works [LP25; KLPP26], we do not have an open-source implementation at the moment. Our plan moving forward is to augment our team with an implementer in order to have an optimized open-source implementation for the final submission. Based on toy implementations in Python, we do not foresee specific challenges for our proposal, including testing and reproducibility challenges.

4. Experimental Performance Evaluation

Sizes and communication. The specification document will contain detailed analysis of the concrete parameters sets, aiming for different security levels. Moreover, we will detail the sizes of the public key ek and the ciphertext ct, as well as the communication cost. Preliminary numbers based on can be found in [KLPP26].

Running time. We expect our schemes to be fairly fast as they only rely on symmetric cryptography and linear algebra over polynomials.

- For the variant based on Vandermonde SSS, each party needs to store $\left(\frac{N}{\log N}\right)^{O(\log N)}$ secret values, therefore we expect the actual bottleneck to be the storage cost.
- For the variant based on Shamir SSS, each party needs to perform $O(T)$ calls to SHAKE during the distributed decapsulation protocol, and we expect these calls to become the computational bottleneck for large T.

5. Licensing, Patent Claims, and Funding

Licensing. We do not have a reference implementation yet, therefore open sources licenses are not relevant to our submission at the moment.

Patents. To our knowledge, two patent families may be applicable to our cryptosystem: WO2024228005A1 and GB202410596D0. Both patents families are owned by PQShield, of which the co-submitters Rafaël del Pino and Thomas Prest are employees of.

Funding. Katharina Boudgoust is supported by the French National Research Agency (ANR), under the projects ANR-21-ASTR-0016 AMIRAL, ANR-22-PECY-003 SecureCompute and ANR-25-CE39-4214-01 RELATE. Oleksandra Lapiha was supported by the EPSRC and the UK Government as part of the Centre for Doctoral Training in Cyber Security for the Everyday at Royal Holloway, University of London (EP/S021817/1). Thomas Prest and Rafaël del Pino are supported by the ANR, under the project ANR-25-CE39-4214-01 RELATE.

References

[KLPP26] Katharina Boudgoust, Oleksandra Lapiha, Rafaël del Pino, and Thomas Prest. *IND-CCA Lattice Threshold KEM under 30 KiB*. Cryptology ePrint Archive, Paper 2026/021. https://eprint.iacr.org/2026/021. 2026.

[BCHK07] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. "Chosen-Ciphertext Security from Identity-Based Encryption". In: *SIAM J. Comput.* 36.5 (2007), pp. 1301–1328. DOI: 10.1137/S009753970544713X.

[BCPENP25] Giacomo Borin, Sofía Celi, Rafaël del Pino, Thomas Espitau, Guilhem Niot, and Thomas Prest. "Threshold Signatures Reloaded: ML-DSA and Enhanced Raccoon with Identifiable Aborts". In: *IACR Cryptol. ePrint Arch.* (2025), p. 1166. URL: https://eprint.iacr.org/2025/1166.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. "Chosen-Ciphertext Security from Identity-Based Encryption". In: 2004, pp. 207–222. DOI: 10.1007/978-3-540-24676-3_13.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. "Bonsai Trees, or How to Delegate a Lattice Basis". In: 2010, pp. 523–552. DOI: 10.1007/978-3-642-13190-5_27.

[DDB95] Yvo Desmedt, Giovanni Di Crescenzo, and Mike Burmester. "Multiplicative Non-abelian Sharing Schemes and their Application to Threshold Cryptography". In: 1995, pp. 21–32. DOI: 10.1007/BFb0000421.

[DKMMPS24] 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: 2024, pp. 219–248. DOI: 10.1007/978-3-031-58723-8_8.

[EENPSS24] Muhammed F. Esgin, Thomas Espitau, Guilhem Niot, Thomas Prest, Amin Sakzad, and Ron Steinfeld. "Plover: Masking-Friendly Hash-and-Sign Lattice Signatures". In: 2024, pp. 316–345. DOI: 10.1007/978-3-031-58754-2_12.

[ENP24] Thomas Espitau, Guilhem Niot, and Thomas Prest. "Flood and Submerse: Distributed Key Generation and Robust Threshold Signature from Lattices". In: 2024, pp. 425–458. DOI: 10.1007/978-3-031-68394-7_14.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions". In: 2008, pp. 197–206. DOI: 10.1145/1374376.1374407.

[KLSS23] Miran Kim, Dongwon Lee, Jinyeong Seo, and Yongsoo Song. "Accelerating HE Operations from Key Decomposition Technique". In: 2023, pp. 70–92. DOI: 10.1007/978-3-031-38551-3_3.

[KRT24] Shuichi Katsumata, Michael Reichle, and Kaoru Takemure. "Adaptively Secure 5 Round Threshold Signatures from MLWE/MSIS and DL with Rewinding". In: 2024, pp. 459–491. DOI: 10.1007/978-3-031-68394-7_15.

[LP25] Oleksandra Lapiha and Thomas Prest. "A Lattice-Based IND-CCA Threshold KEM from the BCHK+ Transform". In: https://eprint.iacr.org/2025/1958. Springer-Verlag, 2025.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: 2010, pp. 1–23. DOI: 10.1007/978-3-642-13190-5_1.

[PKPR24] Rafaël del Pino, Shuichi Katsumata, Thomas Prest, and Mélissa Rossi. "Raccoon: A Masking-Friendly Signature Proven in the Probing Model". In: 2024, pp. 409–444. DOI: 10.1007/978-3-031-68376-3_13.

[Sha79] Adi Shamir. "How to Share a Secret". In: 22.11 (November 1979), pp. 612–613. DOI: 10.1145/359168.359176.

[YJW23] Yang Yu, Huiwen Jia, and Xiaoyun Wang. "Compact Lattice Gadget and Its Applications to Hash-and-Sign Signatures". In: 2023, pp. 390–420. DOI: 10.1007/978-3-031-38554-4_13.

[NIST-IR8214C] Luís T. A. N. Brandão and René Peralta. *NIST First Call for Multi-Party Threshold Schemes*. (National Institute of Standards and Technology) NIST Internal Report (NISTIR) 8214C. 2026. DOI: 10.6028/NIST.IR.8214C.