

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

Title: PANTHERIA: Threshold FHE for RLWE-Based Cryptosystems

Subtitle: Threshold FHE

Version: 1.0 (2026-01-21)¹

Team name: PANTHERIA: PANTHERIA Threshold FHE Team

Team members: Andreea Alexandru, Ahmad Al Badawi, Daniel Apon, Jean-Philippe Bossuat, Sylvain Chatel, Ben Fisch, Nicholas Genise, Shai Halevi, Loïs Huguenin-Dumittan, Guy Itzhaki, Andrey Kim, Yongwoo Lee, Zeyu Liu, Janmajaya Mall, Christian Mouchet, Carlo Pascoe, Chris Peikert, Kabir Peshawaria, Yuriy Polyakov, Saraswathy R.V., Sarabjeet Singh, Yongsoo Song, Eran Tromer, Vinod Vaikuntanathan, Vincent Zucca, Guy Zyskind

Abstract: We first describe the conventional (non-threshold) FHE cryptosystems already implemented in OpenFHE, Lattigo, and Poulpy, which serve as the basis for thresholdization. The following conventional FHE schemes are included: BFV, BGV, CGGI/TFHE, CKKS, DM/FHEW, and LMK+. All of these schemes are based on the hardness of (Ring) Learning With Errors and support various native homomorphic operations. Next, we summarize the thresholdized variants of BFV, BGV, and CKKS implemented in OpenFHE and Lattigo, which use homomorphic addition for distributed key generation and noise flooding for distributed decryption, in the passively secure model for the small and medium categories and dishonest majority. Then, we propose Th-(d)BFV, Th-FHEW, and Th-BGV as thresholdized extensions of (decomposed) BFV, LMK+, and BGV, respectively, that use a Multi-Party Computation (MPC)-based protocol for distributed decryption to support small lattice parameters and achieve active security in the small and medium categories for both dishonest and honest majority settings. We also propose an improved distributed key generation protocol for Th-FHEW, which minimizes the key generation noise. Moreover, we will consider the Laminate verifiable computation method to achieve active security for homomorphic evaluation in the Th-(d)BFV and Th-BGV cryptosystems.

Proposed crypto-systems: (I)Th-(d)BFV: Threshold (decomposed) BFV (Category S5); (II) Th-FHEW: Thresholdized LMK+ variant of FHEW (Category S5). (III)Th-BGV: Threshold BGV (Category S5);

Keywords: Threshold Cryptography; NIST Threshold Call

¹Version submitted to NIST-MPTC for publication

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: Andreea Alexandru^{i1,a1,a2}, Ahmad Al Badawi^{i2,a1,a2}, Daniel Apon^{i3,a3}, Jean-Philippe Bossuat^{i4,a4a9}, Sylvain Chatel^{i5,a5}, Ben Fisch^{i6,a6}, Nicholas Genise^{i7,a2,*}, Shai Halevi^{i8,a7,a2}, Loïs Huguenin-Dumittan^{i9,a8,a9}, Guy Itzhaki^{i10,a11}, Andrey Kim^{i11,a2}, Yongwoo Lee^{i12,a13,a14,a2}, Zeyu Liu^{i13,a6,a2}, Janmajaya Mall^{i14,a4}, Christian Mouchet^{i15,a15,a9}, Carlo Pascoe^{i16,a1,a2}, Chris Peikert^{i17,a10,a11}, Kabir Peshawaria^{i18,a12}, Yuriy Polyakov^{i19,a1,a2}, Saraswathy R.V.^{i20,a2,*}, Sarabjeet Singh^{i21,a1,a2}, Yongsoo Song^{i22,a16}, Eran Tromer^{i23,a12}, Vinod Vaikuntanathan^{i24,a17,a1}, Vincent Zucca^{i25,a2}, Guy Zyskind^{i26,a11}

Open Researcher and Contributor Identifiers (ORCID):

i1 (0000-0001-5396-1241); i2 (0000-0001-7759-7368); i3 (0000-0002-5427-0423); i4 (0000-0002-2020-0224); i5 (0000-0002-1275-8367); i6 (0009-0007-1154-2277); i7 (0000-0001-8625-472X); i8 (0000-0003-3432-7899); i9 (0009-0007-7991-2406); ; i11 (0000-0002-0974-6787); i12 (0000-0001-9424-6498); i13 (0000-0001-7291-3106); i14 (0009-0005-3462-2988); i15 (0000-0001-5686-9459); i16 (0009-0009-6787-0813); i17 (0000-0003-0419-7501); i18 (0009-0006-5192-0626); i19 (0000-0002-5566-3763); i20 (0000-0002-2356-0241); i21 (0000-0003-3032-1916); i22 (0000-0002-0496-9789); i23 (0000-0002-8884-9564); i24 (0000-0002-2666-0045); i25 (0000-0001-7487-6986); i26 (0000-0001-6656-6312)

Affiliations:

- ^{a1} Duality Technologies @ Hoboken, NJ, USA
- ^{a2} OpenFHE team
- ^{a3} Anduril Industries @ Costa Mesa, CA, USA
- ^{a4} Poulpy team
- ^{a5} CISPA Helmholtz Center for Information Security @ Saarbrücken, Germany
- ^{a6} Yale University @ New Haven, CT, USA
- ^{a7} AWS @ New York, NY, USA
- ^{a8} Tune Insight @ Lausanne, Switzerland
- ^{a9} Lattigo team
- ^{a10} University of Michigan @ Ann Arbor, MI, USA
- ^{a11} Fhenix @ Tel Aviv, Israel
- ^{a12} Boston University @ Boston, MA, USA
- ^{a13} Inha University @ Incheon, South Korea
- ^{a14} DESILO Inc. @ Seoul, South Korea
- ^{a15} Hasso-Plattner-Institute, University of Potsdam @ Potsdam, Germany
- ^{a16} Seoul National University @ Seoul, South Korea

^{a17} EECS & CSAIL, Massachusetts Institute of Technology @ Cambridge, MA 02139, USA

Associateship clarifications: * This work was done while this author was at Duality Technologies.

Main contacts:

- **Team mailing list:** pantheria@homomorphicencryption.org
- **Primary technical contact person:** Yuriy Polyakov, ypolyakov@dualitytech.com
- **Secondary contact person 1:** Andreea Alexandru, aalexandru@dualitytech.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

Our submission will include two parts. The first part will focus on conventional passive-security FHE cryptosystems and passive-security thresholdization techniques that are already implemented in the current versions of OpenFHE [AABBC+22], Lattigo [EPF24], and/or Poulpy [Zon25]. The second part will deal with proposed cryptosystems that are being developed specifically for the NIST MPTC call. The proposed cryptosystems will be implemented and evaluated as part of preparing our submission package.

The first part will be described in the Preliminaries of the submission package. The following conventional, i.e., non-threshold, FHE crypto-systems will be included:

1. Brakerski/Fan-Vercauteren (BFV) [Bra12; FV12] and its recent “decomposed” variant dBV [PZZ25],
2. Brakerski-Gentry-Vaikuntanathan (BGV) [BGV14],
3. Chilotti-Gama-Georgieva-Izabachene (CGGI) [CGGI16], also known as TFHE,
4. Cheon-Kim-Kim-Song (CKKS) [CKKS17],
5. Ducas-Micciancio (DM) [DM15] and Lee-Micciancio-Kim-Choi-Deryabin-Eom-Yoo (LMK+) [LMKCDEY23] variants of FHEW.

BGV and (d)BFV are typically used for finite-field arithmetic over vectors, e.g., for Private Information Retrieval and Private Set Intersection. CGGI and DM/LMK+ are commonly used for low-latency operations over (few) small integers and support lookup table evaluation (best suited for blockchain applications). CKKS is often used for applications that deal with the polynomial evaluation over vectors of real/complex numbers, with applications in machine learning.

(d)BFV-based bootstrapping can also be used for high-throughput functional bootstrapping of FHEW/TFHE ciphertexts instead of RGSW, enabling lookup table evaluation of small-integer arbitrary functions [LW23]. Similarly, CKKS-based bootstrapping can be used for high-throughput functional bootstrapping of (d)BFV/RLWE and FHEW/TFHE ciphertexts [BKSS24; AKP25]. CKKS bootstrapping can also be used to bootstrap (d)BFV ciphertexts [KSS24].

Passive-security threshold variants of BGV, BFV, and CKKS are also already implemented in OpenFHE and Lattigo. These variants use homomorphic addition of secret key shares during distributed key generation and noise flooding for distributed decryption. For BFV and CKKS, two methods of interactive bootstrapping are available: (1) a general method for any n [MTBH21] (a slightly improved version is presented in Appendix E of [GGPLR+23]), and (2) a more efficient method for two parties based on distributed rounding (Appendix D of [GGPLR+23]). Interactive bootstrapping is implemented in both OpenFHE and Lattigo.

The second part of our submission will propose threshold FHE cryptosystems that achieve active security. For distributed threshold decryption, we will use the MPC-based approach proposed in [ZZLP25]. We will develop the thresholdized variants of (d)BFV, LMK+, and BGV, which we will label as Th-(d)BFV, Th-FHEW, and Th-BGV, respectively. For Th-FHEW, we will use improved distributed key generation techniques that minimize key generation noise.

Our solutions will target both n -out-of- n and t -out-of- n profiles (here, $t > f$, where f is the corruption threshold in the MPTC call).

All conventional FHE schemes and proposed threshold FHE schemes belong to the S5 category.

2. Specification

2.1. Organization

A Threshold FHE (Th-FHE) scheme consists of the following granular building blocks:

- A (distributed) key generation protocol KeyGen;
- A (non-interactive) public-key encryption algorithm Encrypt;
- A (non-interactive) homomorphic evaluation algorithm Evaluate; and
- A distributed decryption protocol Decrypt.

The Encrypt and Evaluate algorithms are the same as in the case of conventional FHE schemes, with the exception that larger parameters are sometimes needed in the threshold setting. The remaining building blocks are specific to the threshold instantiation, and we will describe them in more detail.

For KeyGen and Decrypt, all FHE schemes deal with generic Module-LWE (of which LWE and RLWE are special cases) schemes. BGV works with the least-significant-digit (LSD) encoding. BFV and LMK+ work with the most-significant-digit (MSD) message encoding, like Regev's (R)LWE scheme. dBFBV uses a more sophisticated "digit decomposition" lattice encoding, following [MP12], to obtain lesser noise growth and thus better parameters and efficiency. We will target (d)BFV, LMK+, and BGV in our proposed cryptosystems. Another building block, the RGSW scheme, is also used as part of LMK+. This scheme requires a thresholdized KeyGen in some scenarios.

We will describe various protocols as modules/gadgets, e.g., we will use an MPC-based module for distributed decryption in Th-(d)BFV, Th-FHEW, and Th-BGV.

2.1.1. Distributed key generation

We consider the following options: (i) trusted dealer, (ii) private channel for initial Shamir secret sharing and homomorphic addition of secret key shares, (iii) homomorphic addition of key secret shares, and (iv) MPC key generation.

Options (ii) and (iii) are currently implemented in Lattigo and OpenFHE.

2.1.2. Distributed decryption

The two main options in the literature to achieve passive security for distributed decryption are MPC and noise flooding. We will examine both options for Th-(d)BFV, Th-FHEW, and Th-BGV. The MPC module for *actively secure* distributed decryption will be developed by the Fhenix team using the method proposed in [ZZLP25].

2.2. System model

We will describe the details on participants, distributed systems, and communication at later stages of the NIST MPTC standardization process.

2.2.1. Threshold profiles

We will consider both n -out-of- n and t -out-of- n scenarios. We are planning to focus on threshold profiles (2), (3), (S), and (M) from the NIST MPTC call, i.e., the cases where n does not exceed 64.

In terms of corruption proportion, we will target dishonest majority (D) (n -out-of- n and $(n - 1)$ -out-of- n) and the two flavors of honest majority (h) ($t > N/2$) and (H) ($t > 2N/3$).

2.3. Security

In the proposed cryptosystems, we will achieve active (adaptive) security for threshold decryption using the MPC-based arithmetic black box model proposed in [ZZLP25]. For distributed key generation, we will consider the polynomial interactive oracle proof approach proposed in [HLSS25].

We will target public key encryption under passive security, and homomorphic evaluation under both passive and active (fully malicious) security. The computational security for (R)LWE will be estimated using the lattice estimator [APS15]. For active-security evaluation of the BGV and (d)BFV schemes, our approach is based on Laminare [PLFT25], which augments the homomorphic evaluation to efficiently produce a SNARG proof to verify that the encrypted outputs were correctly computed. More details will be added at later stages of the NIST MPTC program.

3. Open-Source Implementation

The current open-source implementations of threshold FHE concern passive security. We are planning to extend Th-(d)BFV, Th-FHEW, and Th-BGV to malicious security models as part of preparing the submission. For each proposed cryptosystem, we are planning to provide a reference implementation using at least one of the libraries listed below.

3.1. Open-source libraries

3.1.1. OpenFHE (C++)

OpenFHE implements single-key BGV, BFV, CGGI, CKKS, DM, and LMK+ FHE schemes.

Threshold FHE for BGV/BFV is implemented (2-round relinearization key generation is currently supported). Threshold FHE is implemented for CKKS in OpenFHE as an extension of the $IND-CPA^D$ mode. $IND-CPA^D$ is treated as the 1-party case (for decryption). For BGV, BFV, and CKKS, both n -out-of- n and t -out-of- n profiles are supported. The summary of what is currently implemented in OpenFHE is available at [Threshold FHE in OpenFHE](#).

An early prototype of low-noise threshold LMK+ is available in a feature branch of OpenFHE.

The OpenFHE source code is available at <https://github.com/openfheorg/openfhe-development>.

3.2. Lattigo (Go)

Lattigo implements BGV, BFV, and CKKS and their thresholdized variants. Both n -out-of- n and t -out-of- n profiles are supported. A summary of the threshold schemes implemented in Lattigo is available at [Multiparty Schemes in Lattigo](#).

The Lattigo source code is available at <https://github.com/tuneinsight/lattigo>.

3.3. Poulpy (Rust)

Poulpy provides an implementation in Rust of LMK+ in the conventional FHE setting. The Poulpy team will add a reference implementation of thresholdized LMK+ prior to the submission of the main package. The Poulpy library is accessible at <https://github.com/phantomzone-org/poulpy>.

3.4. Networking model

The published cryptographic literature considers many networking models; for example: asynchronous vs. synchronous, or broadcast channels, or private channels, or authenticated channels, or combinations of the above. All protocols need to be constructed in the same security model as the threshold system. (For example, in the same majority setting, the same passive vs. active threat modeling for the adversary, or point-to-point vs. broadcast, and so on.) In any instantiation, any particular assumptions regarding the existence of a public bulletin board, or a blockchain, etc., will be explicitly mentioned and justified.

However, this approach leads to a combinatorial explosion of possibilities. In the interest of building towards a useful standardization process, we believe identifying specific, important use-cases is critical. So then, which ones?

We take the approach of modern, early-deployed PQC protocols: specifically, Apple iMessage and Signal's encrypted text message chat. In particular, these Double and Triple Ratchet schemes are characterized by adopting NIST PQC KEM standards into complicated network flow protocols, and enhancing them for practice. These MLS-type *broadcast* protocols have uses extending far beyond point-to-point encrypted text messaging—serving as useful, immediate replacements for many heterogeneous systems in the air, space, and maritime environments.

In this context, we plan to examine multiple thresholdized uses of FHE.

3.5. Testing

A standard test-suite/benchmarking suite for functionality in this environment is to consider various experimental workloads: local vs. networked; small data vs. large data; and—particularly in the FHE case—a large variety of representative functions that parties may want to actually compute in the real world.

As a starting point, we will consider calculations over a single core on a local machine. Then, we will consider simulating experimental effects of network latency and bandwidth. The aim of these protocols is to work on actual networks of various types; e.g., 2G, 3G, 4G, 5G networks; 10GB network connections, 100-400GB connections, etc.

Moreover, a central issue in FHE benchmarking is which functions to compute, and on which hardware. We aim to align with community efforts on benchmarking, which consider 3 or 4 major function types, of various multiplicative depths, and running on various hardware architectures. For some such scenarios, we might observe only a 10X overhead on computation compared to the unencrypted calculation; on others, different slowdowns.

While it is too early to report comprehensive and authoritative results, we believe we understand (and are the implementers of) the key methodology and can provide rigorous testing data in the fullness of time.

4. Experimental Performance Evaluation

At this stage, we are focusing on defining the proposed cryptosystems and the gadgets used, e.g., an MPC-based arithmetic black box method for distributed decryption. Once all the components of the cryptosystems are fully specified, we will develop a performance evaluation plan and select a platform for executing the plan. For OpenFHE, we will extend the existing (micro)benchmarking capabilities based on the Google Benchmark library to the proposed threshold cryptosystems.

5. Licensing, Patent Claims, and Funding

5.1. Open-source licenses

The OpenFHE source code is available under the BSD-2 license. The Lattigo source code is available under the Apache 2.0 license. The Poulpy library is available under the Apache 2.0 license.

5.2. Relevant patents

Duality Technologies holds a patent on threshold FHE key generation (US Patent No. [11,962,679](#)). Fhenix has filed provisional patent applications for dBfV.

5.3. External research funding

- Duality Technologies' contribution to the submission will be partially funded by ARPA-H.

References

- [AABBC+22] Ahmad Al Badawi, Andreea Alexandru, Jack Bates, Flavio Bergamaschi, David Bruce Cousins, Saroja Erabelli, Nicholas Genise, Shai Halevi, Hamish Hunt, Andrey Kim, Yongwoo Lee, Zeyu Liu, Daniele Micciancio, Carlo Pascoe, Yuriy Polyakov, Ian Quah, Saraswathy R.V., Kurt Rohloff, Jonathan Saylor, Dmitriy Suponitsky, Matthew Triplett, Vinod Vaikuntanathan, and Vincent Zucca. *OpenFHE: Open-Source Fully Homomorphic Encryption Library*. 2022. DOI: [10.1145/3560827.3563379](https://doi.org/10.1145/3560827.3563379). Also at ia.cr/2022/915.
- [AKP25] Andreea Alexandru, Andrey Kim, and Yuriy Polyakov. “General Functional Bootstrapping Using CKKS”. In: *Advances in Cryptology – CRYPTO 2025*. Ed. by Yael Tauman Kalai and Seny F. Kamara. Cham: Springer Nature Switzerland, 2025, pp. 304–337. DOI: [10.1007/978-3-032-01881-6_10](https://doi.org/10.1007/978-3-032-01881-6_10). Also at ia.cr/2024/1623.
- [APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. In: *Journal of Mathematical Cryptology* 9.3 (2015), pp. 169–203. DOI: [doi:10.1515/jmc-2015-0016](https://doi.org/10.1515/jmc-2015-0016).
- [BGV14] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. “(Leveled) Fully Homomorphic Encryption without Bootstrapping”. In: *ACM Trans. Comput. Theory* 6.3 (2014), 13:1–13:36. DOI: [10.1145/2633600](https://doi.org/10.1145/2633600). Also at ia.cr/2011/277.
- [BKSS24] Youngjin Bae, Jaehyung Kim, Damien Stehlé, and Elias Suvanto. “Bootstrapping Small Integers With CKKS”. In: *Advances in Cryptology – ASIACRYPT 2024*. Ed. by Kai-Min Chung and Yu Sasaki. Singapore: Springer Nature Singapore, 2024, pp. 330–360. DOI: [10.1007/978-981-96-0875-1_11](https://doi.org/10.1007/978-981-96-0875-1_11). Also at ia.cr/2024/1637.
- [Bra12] Zvika Brakerski. “Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP”. In: *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*. Ed. by Reihaneh Safavi-Naini and Ran Canetti. Vol. 7417. Lecture Notes in Computer Science. Springer, 2012, pp. 868–886. DOI: [10.1007/978-3-642-32009-5_50](https://doi.org/10.1007/978-3-642-32009-5_50). Also at ia.cr/2012/078.
- [CGGI16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. “Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds”. In: *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10031. Lecture Notes in Computer Science. 2016, pp. 3–33. DOI: [10.1007/978-3-662-53887-6_1](https://doi.org/10.1007/978-3-662-53887-6_1). Also at ia.cr/2016/870.
- [CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. “Homomorphic Encryption for Arithmetic of Approximate Numbers”. In: *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications*

- of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10624. Lecture Notes in Computer Science. Springer, 2017, pp. 409–437. DOI: [10.1007/978-3-319-70694-8_15](https://doi.org/10.1007/978-3-319-70694-8_15). Also at ia.cr/2016/421.
- [DM15] Léo Ducas and Daniele Micciancio. “FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second”. In: *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9056. Lecture Notes in Computer Science. Springer, 2015, pp. 617–640. DOI: [10.1007/978-3-662-46800-5_24](https://doi.org/10.1007/978-3-662-46800-5_24). Also at ia.cr/2014/816.
- [EPF24] Tune Insight SA EPFL-LDS. *Lattigo v6*. Online: <https://github.com/tuneinsight/lattigo>. August 2024.
- [FV12] Junfeng Fan and Frederik Vercauteren. “Somewhat Practical Fully Homomorphic Encryption”. In: *IACR Cryptol. ePrint Arch.* (2012), p. 144. URL: <http://eprint.iacr.org/2012/144>.
- [GGPLR+23] Ravit Geva, Alexander Gusev, Yuriy Polyakov, Lior Liram, Oded Rosolio, Andreea Alexandru, Nicholas Genise, Marcelo Blatt, Zohar Duchin, Barliz Weissengrin, Dan Mirelman, Felix Bukstein, Deborah T. Blumenthal, Ido Wolf, Sharon Pelles-Avraham, Tali Schaffer, Lee A. Lavi, Daniele Micciancio, Vinod Vaikuntanathan, Ahmad Al Badawi, and Shafi Goldwasser. “Collaborative privacy-preserving analysis of oncological data using multiparty homomorphic encryption”. In: *Proceedings of the National Academy of Sciences* 120.33 (2023), e2304415120. DOI: [10.1073/pnas.2304415120](https://doi.org/10.1073/pnas.2304415120). Also at ia.cr/2023/1203.
- [HLSS25] Intak Hwang, Hyeonbum Lee, Jinyeong Seo, and Yongsoo Song. “Practical zero-knowledge PIOP for maliciously secure multiparty homomorphic encryption”. In: *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*. 2025, pp. 4049–4063. DOI: [10.1145/3719027.3765229](https://doi.org/10.1145/3719027.3765229). Also at ia.cr/2024/1879.
- [KSS24] Jaehyung Kim, Jinyeong Seo, and Yongsoo Song. “Simpler and Faster BFV Bootstrapping for Arbitrary Plaintext Modulus from CKKS”. In: *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. CCS '24. Salt Lake City, UT, USA: Association for Computing Machinery, 2024, pp. 2535–2546. DOI: [10.1145/3658644.3670302](https://doi.org/10.1145/3658644.3670302). Also at ia.cr/2024/109.
- [LMKCDEY23] Yongwoo Lee, Daniele Micciancio, Andrey Kim, Rakyong Choi, Maxim Deryabin, Jieun Eom, and Donghoon Yoo. “Efficient FHEW Bootstrapping with Small Evaluation Keys, and Applications to Threshold Homomorphic Encryption”. In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27,*

- 2023, *Proceedings, Part III*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14006. Lecture Notes in Computer Science. Springer, 2023, pp. 227–256. DOI: [10.1007/978-3-031-30620-4_8](https://doi.org/10.1007/978-3-031-30620-4_8). Also at ia.cr/2022/198.
- [LW23] Zeyu Liu and Yunhao Wang. “Amortized Functional Bootstrapping in Less than 7 ms, with $\tilde{O}(1)$ Polynomial Multiplications”. In: *Advances in Cryptology – ASIACRYPT 2023*. Ed. by Jian Guo and Ron Steinfeld. Singapore: Springer Nature Singapore, 2023, pp. 101–132. DOI: [10.1007/978-981-99-8736-8_4](https://doi.org/10.1007/978-981-99-8736-8_4). Also at ia.cr/2023/910.
- [MP12] Daniele Micciancio and Chris Peikert. “Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller”. In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 700–718. DOI: [10.1007/978-3-642-29011-4_41](https://doi.org/10.1007/978-3-642-29011-4_41). Also at ia.cr/2011/501.
- [MTBH21] Christian Mouchet, Juan Ramón Troncoso-Pastoriza, Jean-Philippe Bossuat, and Jean-Pierre Hubaux. “Multiparty Homomorphic Encryption from Ring-Learning-with-Errors”. In: *PoPETs 2021.4* (October 2021), pp. 291–311. DOI: [10.2478/popets-2021-0071](https://doi.org/10.2478/popets-2021-0071). Also at ia.cr/2020/304.
- [PLFT25] Kabir Peshawaria, Zeyu Liu, Ben Fisch, and Eran Tromer. *Laminate: Succinct SIMD-Friendly Verifiable FHE*. Cryptology ePrint Archive, Paper 2025/2285. 2025. URL: <https://eprint.iacr.org/2025/2285>.
- [PZZ25] Chris Peikert, Doron Zarchy, and Guy Zyskind. *High-Precision Exact FHE Made Simple, General, and Fast*. Cryptology ePrint Archive, Paper 2025/2321. 2025. URL: <https://eprint.iacr.org/2025/2321>.
- [Zon25] Phantom Zone. *Poulpy v0.1.0*. Online: <https://github.com/phantomzone-org/poulpy>. August 2025.
- [ZZLP25] Guy Zyskind, Doron Zarchy, Max Leibovich, and Chris Peikert. “High-Throughput Universally Composable Threshold FHE Decryption”. In: *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*. 2025, pp. 2339–2353. DOI: [10.1145/3719027.3744884](https://doi.org/10.1145/3719027.3744884).
- [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](https://doi.org/10.6028/NIST.IR.8214C).