

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

Title: MOZI: Threshold FHE at Scale

Subtitle: Efficient and Resilient Threshold FHE for Large-Scale MPC

Version: Preview Writeup 1.1 (2026-09-29)¹

Team name: MOZI

Team members: Man Ho Au, Yijia Chang, Rongmao Chen, Xinyi Huang, Xianhui Lu, Moti Yung

Abstract: MOZI is a planned submission package for threshold fully homomorphic encryption (ThFHE) in systems with hundreds or thousands of parties. Its goals are 1) to keep threshold decryption efficient as the number of parties grows and 2) to achieve active security against up to $t - 1$ corrupted share holders (t is the recovery threshold). At the first design level, MOZI separates threshold decryption from the underlying FHE through an approximate secret sharing (ApproxSS) interface. At the second design level, ApproxSS is separated from its two concrete realizations. ATASSES (Arbitrary-Threshold ApproxSS based on Encrypted Shares) uses lattice-based symmetric encryption and targets post-quantum security. The other realization uses Threshold Lattice-Exp-ElGamal (ThLEG), a threshold linearly homomorphic encryption based on elliptic-curve group operations. It offers lower complexity but is quantum-vulnerable.

The package will specify two complete Category S5 ThFHE systems, MOZI-Fang and MOZI-Yuan, based on ATASSES and ThLEG, respectively. It will also specify ApproxSS and ThLEG as standalone Category S7 gadgets. All these systems will use zero-knowledge proofs to achieve active security. In both cloud-aided and peer-to-peer deployments, a designated combiner will coordinate recovery over an asynchronous network. Under separate threshold profiles, MOZI will target guaranteed output delivery with a semi-honest combiner and security-with-abort against a malicious combiner. The public package will include technical specifications and reference code. It will also provide reproducible tests and benchmark scripts.

Proposed crypto-systems:

- (I) MOZI-Fang: ATASSES-Based Threshold FHE (Categories S5)
- (II) MOZI-Yuan: ThLEG-Based Threshold FHE (Categories S5)
- (III) Approximate Secret Sharing (ApproxSS) (Categories S7)
- (IV) Threshold Lattice-Exp-ElGamal (ThLEG) (Categories S7)

Keywords: Threshold Cryptography; Fully Homomorphic Encryption; Approximate Secret Sharing; NIST Threshold Call

Document License: [CC BY 4.0 International](https://creativecommons.org/licenses/by/4.0/)

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: Man Ho Au^{i1,a1}, Yijia Chang^{i2,a1}, Rongmao Chen^{i3,a2}, Xinyi Huang^{i4,a3}, Xianhui Lu^{i5,a4,a5}, Moti Yung^{i6,a6,a7}

Open Researcher and Contributor Identifiers (ORCID):

i1 (0000-0003-2068-9530); i2 (0009-0008-4999-7821); i3 (0000-0002-5113-387X); i4 (0000-0003-0070-1707); i5 (0000-0001-7091-5810); i6 (0000-0003-0848-0873)

Affiliations:

^{a1} Department of Computing, The Hong Kong Polytechnic University @ Hong Kong, China

^{a2} College of Computer Science and Technology, National University of Defense Technology @ Changsha, China

^{a3} College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics @ Nanjing, China

^{a4} Institute of Information Engineering, Chinese Academy of Sciences @ Beijing, China

^{a5} School of Cyber Security, University of Chinese Academy of Sciences @ Beijing, China

^{a6} Department of Computer Science, Columbia University @ New York, USA

^{a7} Google Inc. @ USA

Main contacts:

- **Team mailing list:** mozi-threshold-fhe@polyu.edu.hk
- **Primary technical contact person:** Yijia Chang, yijia.chang.crypto@gmail.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

MOZI is a planned submission package for threshold fully homomorphic encryption (ThFHE), with a particular focus on large-scale multi-party applications. Such applications arise in federated learning, cross-organizational analytics, and large-scale MPC preprocessing. ThFHE is particularly pertinent to these settings because homomorphic evaluation can be delegated to an untrusted party without revealing the inputs, while authority over the result remains distributed among the participants. Unlike general interactive MPC, the data owners need not jointly process every operation or remain online throughout the homomorphic evaluation. Such systems require efficient recovery and clear security guarantees as the number of parties grows.

MOZI follows a two-level modular design. At the first level, threshold decryption is decoupled from the underlying FHE scheme through an Approximate Secret Sharing (ApproxSS) interface [CL25]. This interface captures the approximate reconstruction functionality needed to protect partial decryptions using noise smudging. It will support various FHE schemes, with BFV [FV12], TFHE [CGGI16], and CKKS [CKKS17] as representative examples. At the second level, ApproxSS is decoupled from its concrete realization. The package will provide two implementations. One is Arbitrary-Threshold ApproxSS based on Encrypted Shares (ATASSES). It uses lattice-based symmetric encryption, such as symmetric BFV, and targets post-quantum security. The other uses threshold Lattice-Exp-ElGamal (ThLEG), a threshold linearly homomorphic encryption scheme that relies on elliptic-curve group operations [CCLH25]. It provides a quantum-vulnerable alternative aimed at reducing recovery costs in systems within “Enormous” class defined in the Threshold Call [NIST-IR8214C]. These interchangeable realizations allow the two design points to be compared under a common ThFHE interface and evaluation methodology.

The planned package will propose two complete cryptosystems under Category S5: MOZI-Fang and MOZI-Yuan. MOZI-Fang instantiates ApproxSS with ATASSES, while MOZI-Yuan uses the ThLEG-based realization. Both will provide complete threshold-decryption functionality for various FHE schemes and will be specified, implemented, and experimentally evaluated. The package will also propose two standalone gadgets under Category S7. ApproxSS will be submitted as a reusable abstraction for constructing efficient t -out-of- n threshold decryption, together with its concrete realizations. ThLEG will be submitted separately as a reusable threshold linearly homomorphic encryption gadget. Each S7 gadget will receive an independent specification, implementation, security analysis, and experimental evaluation in addition to its use within the complete S5 systems.

The resulting package is intended to provide both deployable ThFHE systems and reusable components for future threshold constructions. Its evaluation will cover cloud-aided and peer-to-peer deployments, asynchronous networks, and malicious adversaries. Authenticated Beaver-triple generation [Bea92; DPSZ12; KPR18] will serve as a large-scale MPC workload. The specifications, reference implementations, tests, benchmark scripts, and experimental results will provide evidence about scalability, resilience, security assumptions, and implementation tradeoffs. By presenting two complete S5 systems and two independently analyzable S7 gadgets within a common framework, MOZI directly supports the Threshold Call’s goal [NIST-IR8214C].

2. Specification

2.1. Organization

The Technical Specification will follow the two-level structure of MOZI. It will first describe the common building blocks. These include conventional FHE schemes such as BFV [FV12], TFHE [CGGI16], and CKKS [CKKS17]. It will also introduce linear secret sharing [Sha79] and noise smudging. The rest of the document will contain four crypto-system parts. Two parts will specify the proposed Category S7 gadgets. The other two will specify the complete Category S5 ThFHE systems.

The first S7 gadget is approximate secret sharing (ApproxSS). A t -out-of- n ApproxSS scheme consists of a sharing algorithm Share and an approximate-recovery protocol $\Pi_{\text{ApproxRec}}$. The sharing algorithm divides a message m into n shares. The approximate recovery protocol takes shares from any qualified set of at least t parties. It returns an approximate message $m' = m + e$, where e lies in a stated error region. The specification of ApproxSS will define two properties beyond vanilla secret sharing. Approximate correctness bounds the size of e , while approximate security limits what the recovery transcript reveals about the exact value of m . ApproxSS will have two concrete realizations. The first realization called ATASSES uses lattice-based symmetric encryption [CL25]. The second uses threshold linearly homomorphic encryption [CCLH25]. Both realizations implement the same ApproxSS interface.

The second S7 gadget is Threshold Lattice-Exp-ElGamal (ThLEG), a threshold linearly homomorphic encryption scheme [CCLH25]. Its specification will define key generation, encryption, linear evaluation, and threshold decryption. ThLEG is used as one realization of ApproxSS, but it is not tied to ThFHE. It may also serve as a building block for other threshold protocols.

The two S5 systems MOZI-Fang and MOZI-Yuan will share a generic construction from ApproxSS to ThFHE. Each party first computes a share of the FHE decryption value. The parties then invoke $\Pi_{\text{ApproxRec}}$ on these shares. The recovered value contains a controlled extra error. Decryption succeeds when this error stays within the decoding margin of the underlying FHE. MOZI-Fang and MOZI-Yuan will use the ATASSES and ThLEG realizations, respectively.

All four systems will specify zero-knowledge proofs for verifying recovery contributions and the combiner outputs. The specification will define the required consistency relations and error bounds, and bind each contribution to relevant input materials.

This organization separates three questions. The generic ThFHE construction explains how ApproxSS gives threshold decryption. The ApproxSS part explains the exact interface needed by that construction. The two concrete realizations explain how the interface can be implemented under different assumptions and performance tradeoffs.

2.2. System Model

We consider a system with n share holders. These parties communicate over authenticated, reliable point-to-point channels in an asynchronous network [CGHZ16]. Messages sent between honest parties are eventually delivered, but their delivery times and order are otherwise unconstrained. The same delivery guarantee applies to communication between these parties and a semi-honest combiner. This system model is associated with a threshold t ($1 \leq t \leq n$). Threshold decryption requires valid contributions from at least t distinct share holders. Note that only authenticated contributions that pass the verification specified in Section 2.1 count toward the threshold. Meanwhile, the set of parties whose contributions are used may vary between protocol rounds and executions. We assume that each ciphertext submitted to threshold decryption is honestly produced, i.e., either by honest encryption or by correct homomorphic evaluation on honestly generated ciphertexts, within the specified parameter and noise bounds. Each recovery request is assumed to be authorized by the application under a policy fixed at setup. The application also specifies the intended output recipients.

Each recovery execution has a *combiner* that collects and combines contributions. The combiner obtains the recovered result and sends it to the intended output recipients, which may include all parties. This system corresponds to two deployment models. In the cloud-aided model, a semi-honest cloud evaluates the FHE ciphertexts and coordinates threshold decryption. The cloud does not control the secret key and plays the role of pure combiner. In the peer-to-peer model, any party can play the role of combiner, implying that the combiner may be malicious.

The setup uses a trusted dealer for key generation, implying that distributed key generation is outside the planned scope. This dealer creates public parameters and distributes secret-key shares. The setup also provides a participant list and the public data needed to verify signatures and zero-knowledge proofs. MOZI may use an optional preprocessing phase to reduce online work. This phase is not necessary for correctness or security. When preprocessing is enabled, fresh materials are prepared for each recovery execution individually and cannot be reused.

2.3. Security

All four proposed cryptosystems will target simulation-based security against a static adversary corrupting up to f share holders, with a fixed corruption bound $f = t - 1$. We assume a malicious adversary, i.e., corrupted parties may deviate arbitrarily from the protocol. The combiner may share its view with the adversary. If the combiner also holds a share, its corruption counts toward this bound.

Security will be stated at each layer. For ApproxSS, the specification will define ordinary privacy of the shares. Fewer than t shares alone will reveal no information about m . In particular, they will not suffice to recover the noisy value $m + e$ produced by the recovery protocol. Beyond ordinary privacy, approximate security will show that an authorized recovery transcript reveals no more about m than the permitted noisy value $m + e$. Approximate correctness will ensure that the added error remains small enough within the predefined range.

For ThLEG, the specification will define confidentiality and correctness for threshold linearly homomorphic encryption. Its security will rely on the assumptions of its lattice component and its elliptic-curve group. ThLEG is therefore quantum-vulnerable. The ATASSES realization of ApproxSS will instead target post-quantum security based on lattice-based assumptions.

For the complete S5 systems, the main goals are the security of plaintexts, as well as the secret key. A coalition of at most $t - 1$ corrupted share holders must not recover the secret key. It must not learn more about the plaintext than the authorized output reveals. The security analysis will cover the underlying FHE scheme and the chosen ApproxSS realization.

We consider two types of combiner that lead to different threshold profiles and security targets. With a semi-honest combiner, we aim to achieve guaranteed output delivery to all intended output recipients, and the profile will require $t + f \leq n$, which becomes $t \leq \lfloor (n + 1)/2 \rfloor$ for $f = t - 1$. With a malicious combiner, the profile will use $1 \leq t \leq n$ and the target is security with abort. A malicious combiner may prevent completion or withhold the result from some recipients, but must not compromise confidentiality or the correctness of any output accepted by an honest recipient. For both combiner types, the core security claim will not include generic verification of FHE homomorphic evaluation. We assume that the ciphertext to be decrypted is honestly generated, as described in Section 2.2.

The parameter analysis will use worst-case bounds for the FHE decryption error and the error introduced by approximate recovery. It will state the required noise-smudging distribution and the allowed number of decryption queries [GNSJ24]. The underlying FHE instances will achieve 128-bit classical security. The security of the ThLEG system will also depend on the chosen group. With the current Ristretto-based implementation, this profile will target about 128-bit classical security [VGHLTV23].

3. Open-Source Implementation

3.1. Code Structure and Progress

The reference implementation will be written in Go and C++. It will contain an ApproxSS module with two realizations: ATASSES for MOZI-Fang and ThLEG for MOZI-Yuan. The ATASSES realization will use symmetric BFV. A separate ThLEG module will provide threshold linearly homomorphic encryption. The ThFHE module will connect the ApproxSS interface to the decryption interfaces of supported FHE schemes. BFV, TFHE, and CKKS will serve as representative examples. The implementation will include proof generation and verification for all four cryptosystems.

Conventional FHE operations will use Lattigo [EPF] and OpenFHE [Ope]. The current ThLEG implementation uses go-ristretto [Wes]. The main protocols have been implemented, but the repository is not yet public. The published starting points are ATASSES [CL25] and ThLEG [CCLLH25]. The current work focuses on reorganizing the code around the common interfaces. We are also completing the malicious-share checks, dependency review, and deployment scripts. The public repository will include parameter files and example configurations for

all four proposed crypto-systems. The repository will also contain the networking layer, tests, and benchmark programs.

3.2. Networking and Testing

Each party will run as an independent process. The cloud will run as a separate process in the cloud-aided model. Processes will communicate through authenticated TCP connections. Signed contributions and their proofs will be checked using the registered verification keys and public verification data. Each contribution will be bound to its sender, recovery execution, and other relevant materials. The receiver will accept messages in any order and proceed after collecting t contributions with valid proofs. In the malicious-combiner profile, each output recipient will verify the combiner’s zero-knowledge proof against the authorized recovery request before accepting the result. Tests will introduce delayed and reordered messages. They will also model temporary disconnection and timeout. Timeouts are used only to control test runs and are not part of the protocol’s termination rule. Malicious tests will include invalid shares, replayed messages, incorrect combiner outputs, and selective withholding of results. The test suite will provide a deterministic mode with fixed random seeds for testing and benchmarking.

4. Experimental Performance Evaluation

4.1. Evaluation Scope

The evaluation will cover both ApproxSS realizations, standalone ThLEG, and end-to-end execution of the two complete ThFHE systems. BFV, TFHE, and CKKS will be tested separately. All four proposed cryptosystems will be evaluated under both combiner profiles within their respective threshold ranges. The main configurations are summarized below.

Table 1: Planned evaluation configurations.

Dimension	Planned configurations
Participants	$n \in \{16, 64, 256, 1024, 4096\}$
Thresholds	$t = \lfloor (n+1)/2 \rfloor$ for both profiles; $t = n$ for the malicious-combiner profile
FHE schemes	BFV, TFHE, and CKKS with both ApproxSS realizations
Security	128-bit target for the underlying HE instances
Exact plaintexts	$p = 2$, $p = 65537$, $p = 786433$, and a batching-compatible prime close to 2^{60}
CKKS precision	20, 30, and 40 bits of final precision; 50 bits when practical
Circuit depth	Multiplicative depths 1, 3, 5, and 8
Faults	Delayed parties, temporary disconnection, timeout, and invalid outputs

For BFV and CKKS, the depth-8 experiment will include one bootstrapping operation. We will consider conventional FHE bootstrapping and interactive threshold bootstrapping where supported. The final submission will report the options used and their security and performance

tradeoffs. For TFHE, the evaluation will include programmable bootstrapping and report its cost and frequency in each workload.

The evaluation will report online and total latency, throughput, memory use, and per-party communication. Preprocessing cost will be separated from online cost. Total costs will include proof generation, verification, and all other protocol overhead. The report will also give the sizes of secret-key shares, public and evaluation keys, and recovery material. It will show how these costs change with n and t .

4.2. Large-Scale MPC Workload

The main application benchmark will generate authenticated Beaver triples for SPDZ-style MPC [Bea92; DPSZ12; KPR18]. The planned batch sizes include 2^{10} , 2^{16} , and 2^{20} triples. Larger streaming tests will be included when memory permits. The report will measure candidate triples and usable triples separately. It will include the cost of checking and sacrifice. The main throughput metric will be usable triples per second. A small MPC application will consume the triples to confirm that the exported shares and authentication data have the required format.

4.3. Baselines and Platforms

The evaluation will compare MOZI with known schemes such as ZZLP [ZZLP25], KLL [KLLPS26], and Ajax [HLWXYZ26] at the relevant abstraction layers, under the same threshold functionality and security model. Comparisons will use the same FHE scheme, workload, and values of n and t whenever possible, with comparable security and correctness parameters. Differences in corruption bounds, output guarantees, setup, combiner trust, and network assumptions will be stated explicitly. Results obtained by rerunning public code will be separated from values copied from published papers.

The NIST baseline platform has 16 cores and 64 GB of RAM [PubComs2PD]. We will run all configurations that fit on this machine. We will also conduct real multi-machine experiments over TCP that cover local-area and wide-area networks. Each result will state the hardware and measured network conditions. Each experiment will state its number of repetitions and warm-up policy. The report will provide the mean, median, and a tail-latency measure such as the 95th percentile.

5. Licensing, Patent Claims, and Funding

The planned license for the core code is Apache-2.0, subject to final rights review. Lattigo uses Apache-2.0, while the current go-ristretto dependency uses the MIT License [EPF; Wes]. The final package will list the exact version and license of every dependency. At present, the team is not aware of a patent claim within the scope of the Threshold Call that must be listed in this preview. No external funding acknowledgement is currently included.

6. Comparison with Other Submissions

MOZI focuses on making threshold recovery practical for hundreds or thousands of share holders. Its main contribution is a modular design that combines this scale with support for different FHE schemes. PANTHERIA [PAN26] focuses on configurations with at most 64 parties, while the Nexus variant for large $\binom{n}{f}$ currently supports TFHE [Zam26]. MOZI will explore dedicated ApproxSS constructions across representative schemes, including BFV, TFHE, and CKKS, with the aim of keeping recovery computation, communication, and storage manageable as the number of parties grows.

The ApproxSS interface supports two complementary approaches to this goal. MOZI-Fang uses lattice-based symmetric encryption through ATASSES and targets post-quantum security. MOZI-Yuan uses ThLEG to explore lower recovery costs under quantum-vulnerable assumptions. Their common interface allows these security and performance choices to be evaluated within the same ThFHE framework. The standalone ApproxSS and ThLEG gadgets will also make the recovery techniques reusable beyond the two complete systems.

MOZI concentrates its scope on recovery through trusted-dealer key generation and asynchronous network. This setup allows the package to focus on the recurring cost of recovery, with optional preprocessing to reduce online work. PANTHERIA considers both trusted-dealer and distributed key generation and leaves its networking choices open [PAN26]. Nexus includes distributed key generation and uses asynchronous online protocols, but key generation and its large- $\binom{n}{f}$ variant require synchronous offline phases [Zam26]. These setup and networking conditions will be accounted for when evaluating the cost of large-scale deployments.

MOZI also makes combiner trust an explicit deployment choice while targeting active security against static corruptions of share holders. With $f = t - 1$, its semi-honest-combiner profile targets guaranteed output delivery for $t + f \leq n$; its malicious-combiner profile targets security with abort for $1 \leq t \leq n$. We are also working on protocol extensions to achieve guaranteed output delivery in the malicious-combiner setting for $f < n/3$. PANTHERIA targets adaptive active security for threshold decryption, but its preview does not specify which threshold profiles provide guaranteed output delivery. Nexus describes guaranteed output delivery through robust MPC, with $f < n/3$ online, without stating an additional semi-honest-combiner assumption. MOZI's recovery guarantees assume honestly produced input ciphertexts, as specified in Section 2.3. Its planned evaluation will identify where dedicated recovery constructions and these deployment choices offer practical advantages at large n .

References

- [Bea92] Donald Beaver. “Efficient Multiparty Protocols Using Circuit Randomization”. In: *Advances in Cryptology – CRYPTO ’91*. Vol. 576. Lecture Notes in Computer Science. Springer, 1992, pp. 420–432. DOI: [10.1007/3-540-46766-1_34](https://doi.org/10.1007/3-540-46766-1_34).
- [CCLH25] Yijia Chang, Rongmao Chen, Chao Lin, Songze Li, and Xinyi Huang. “Resolving the Efficiency-Utility Dilemma of Threshold Linearly Homomorphic Encryption via Message-Space Adapter”. In: *Advances in Cryptology – CRYPTO 2025*. Vol. 16002. Lecture Notes in Computer Science. Springer, 2025, pp. 269–303. DOI: [10.1007/978-3-032-01881-6_9](https://doi.org/10.1007/978-3-032-01881-6_9). Also at ia.cr/2025/948.
- [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*. Vol. 10031. Lecture Notes in Computer Science. Springer, 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.
- [CGHZ16] Sandro Coretti, Juan Garay, Martin Hirt, and Vassilis Zikas. “Constant-Round Asynchronous Multi-Party Computation Based on One-Way Functions”. In: *Proceedings, Part II, of the 22nd International Conference on Advances in Cryptology — ASIACRYPT 2016 - Volume 10032*. Berlin, Heidelberg: Springer-Verlag, 2016, pp. 998–1021. DOI: [10.1007/978-3-662-53890-6_33](https://doi.org/10.1007/978-3-662-53890-6_33). Also at ia.cr/2016/208.
- [CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. “Homomorphic Encryption for Arithmetic of Approximate Numbers”. In: *Advances in Cryptology – ASIACRYPT 2017*. 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.
- [CL25] Yijia Chang and Songze Li. “Arbitrary-Threshold Fully Homomorphic Encryption with Lower Complexity”. In: *34th USENIX Security Symposium*. USENIX Association, 2025, pp. 8403–8422. URL: <https://www.usenix.org/conference/usenixsecurity25/presentation/chang-yijia-encryption>. Also at ia.cr/2025/084.
- [DPSZ12] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. “Multiparty Computation from Somewhat Homomorphic Encryption”. In: *Advances in Cryptology – CRYPTO 2012*. Vol. 7417. Lecture Notes in Computer Science. Springer, 2012, pp. 643–662. DOI: [10.1007/978-3-642-32009-5_38](https://doi.org/10.1007/978-3-642-32009-5_38). Also at ia.cr/2011/535.
- [EPF] EPFL-LDS and Tune Insight SA. *Lattigo*. URL: <https://github.com/tuneinsight/lattigo> (visited on 08/05/2026).
- [FV12] Junfeng Fan and Frederik Vercauteren. *Somewhat Practical Fully Homomorphic Encryption*. Cryptology ePrint Archive, Paper 2012/144. 2012. URL: <https://eprint.iacr.org/2012/144>.
- [GNSJ24] Qian Guo, Denis Nabokov, Elias Suvanto, and Thomas Johansson. “Key Recovery Attacks on Approximate Homomorphic Encryption with Non-Worst-Case Noise Flooding Countermeasures”. In:

33rd USENIX Security Symposium. USENIX Association, 2024, pp. 7447–7461. URL: <https://www.usenix.org/conference/usenixsecurity24/presentation/guo-qian>.

[HLWXYZZ26] Zhenkai Hu, Haofei Liang, Xiao Wang, Xiang Xie, Kang Yang, Yu Yu, and Wenhao Zhang. “Ajax: Fast Threshold Fully Homomorphic Encryption without Noise Flooding”. In: *35th USENIX Security Symposium*. USENIX Association, 2026, pp. 1–20. URL: <https://www.usenix.org/conference/usenixsecurity26/presentation/hu-zhenkai>. Also at ia.cr/2025/1834.

[KLLPS26] Won Kim, Changmin Lee, JeongHwan Lee, Alain Passelègue, and Damien Stehlé. “Asynchronous Lagrange-Based Threshold FHE with Smaller Modulus Overhead”. In: *Advances in Cryptology – CRYPTO 2026*. Vol. 16801. Lecture Notes in Computer Science. Springer, 2026, pp. 3–35. DOI: [10.1007/978-3-032-35374-0_1](https://doi.org/10.1007/978-3-032-35374-0_1). Also at ia.cr/2026/973.

[KPR18] Marcel Keller, Valerio Pastro, and Dragos Rotaru. “Overdrive: Making SPDZ Great Again”. In: *Advances in Cryptology – EUROCRYPT 2018*. Vol. 10822. Lecture Notes in Computer Science. Springer, 2018, pp. 158–189. DOI: [10.1007/978-3-319-78372-7_6](https://doi.org/10.1007/978-3-319-78372-7_6). Also at ia.cr/2017/1230.

[Ope] OpenFHE. *OpenFHE: Open-Source Fully Homomorphic Encryption Library*. URL: <https://github.com/openfheorg/openfhe-development>.

[PAN26] PANTHERIA. *PANTHERIA: Threshold FHE for RLWE-Based Cryptosystems*. NIST Multi-Party Threshold Cryptography Preview Writeup. January 21, 2026. URL: <https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/PANTHERIA-PW01.pdf>.

[Sha79] Adi Shamir. “How to Share a Secret”. In: *Communications of the ACM* 22.11 (1979), pp. 612–613. DOI: [10.1145/359168.359176](https://doi.org/10.1145/359168.359176).

[VGHLTV23] H. de Valence, J. Grigg, M. Hamburg, I. Lovecruft, G. Tankersley, and F. Valsorda. *RFC 9496: The ristretto255 and decaf448 Groups*. USA, 2023. DOI: [10.17487/RFC9496](https://doi.org/10.17487/RFC9496).

[Wes] Bram Westerbaan. *go-ristretto: Pure Go Implementation of the Ristretto Prime-Order Group*. URL: <https://github.com/bwesterb/go-ristretto> (visited on 08/05/2026).

[Zam26] Zama. *TFHE, ZHENith and Nexus: A Suite of Cryptosystems to Enable Fully Homomorphic Encryption Applications*. NIST Multi-Party Threshold Cryptography Preview Writeup. January 9, 2026. URL: <https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/Zama-PW01.pdf>.

[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*. ACM, 2025, pp. 2339–2353. DOI: [10.1145/3719027.3744884](https://doi.org/10.1145/3719027.3744884). Also at ia.cr/2025/1781.

[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).

[PubComs2PD] NIST-MPTC. *Compilation of Public Comments on NIST IR 8214C 2pd*. National Institute of Standards and Technology (NIST), Multi-Party Threshold Cryptography. June 2025. URL: <https://csrc.nist.gov/files/pubs/ir/8214/c/2pd/docs/nistir-8214c-2pd-public-feedback.pdf>.