

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

Title: Tulva: Asynchronous Threshold FHE Transform with Small Decryption Shares

Version: Preview Writeup 1.0 (2026-10-05)¹

Team name: Tulva

Team members: Valerio Cini, Antonio Guimarães, Aaron Kaiser, Russell W. F. Lai, Michał Osadnik, Ivy K. Y. Woo

Abstract: Tulva is a general compiler that turns typical lattice-based fully homomorphic encryption (FHE) schemes into threshold FHE schemes. It supports a wide range of threshold profiles (k, n) , where n is the number of shareholders and k the number of them needed to decrypt, including any choice of $k \leq n \leq 128$, and any choice of $n \leq 512$ and k such that $\min(k, n - k) \leq 16$. A compiled threshold FHE scheme preserves the ciphertext format and homomorphic computation procedures of the base scheme, has small decryption share size, and features fully asynchronous partial decryption. The message encrypted in any fresh or homomorphically evaluated ciphertext of the compiled scheme remains hidden against a (classical or quantum) adversary who statically corrupts $f' \leq f = k - 1$ shareholders and has access to fewer than $k - f'$ partial decryptions of this ciphertext generated by non-corrupted shareholders, assuming that the base FHE scheme and the hash functions are secure and that the module learning with errors (MLWE) assumption holds.

Proposed crypto-systems: (I) Tulva: threshold FHE (Categories S5).

Keywords: Threshold Cryptography; NIST Threshold Call; Threshold FHE

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: Valerio Cini^{i1,a1}, Antonio Guimarães^{i2,a2}, Aaron Kaiser^{i3,a3}, Russell W. F. Lai^{i4,a4}, Michał Osadnik^{i5,a4}, Ivy K. Y. Woo^{i6,a5}

Open Researcher and Contributor Identifiers (ORCID):

i1 (0009-0003-6876-0954); i2 (0000-0001-5110-6639); i3 (0009-0004-6141-4861); i4 (0000-0001-9126-1887); i5 (0000-0003-4000-3241); i6 (0000-0001-8905-1207)

Affiliations:

^{a1} Department of Computing Sciences, Bocconi University @ Milan, Italy

^{a2} IMDEA Software Institute @ Madrid, Spain

^{a3} Max Planck Institute for Security and Privacy @ Bochum, Germany

^{a4} Department of Computer Science, Aalto University @ Espoo, Finland

^{a5} Department of Mathematics and Systems Analysis, Aalto University @ Espoo, Finland

Main contacts:

- **Team mailing list:** tulva-threshold-fhe@googlegroups.com
- **Primary technical contact person:** Russell W. F. Lai, russell.lai@aalto.fi
- **Secondary contact person:** Antonio Guimarães, antonio.guimaraes@imdea.org

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

We plan to propose Tulva and its extension Tulva⁺, a general compiler that turns typical lattice-based fully homomorphic encryption (FHE) schemes into threshold FHE schemes. The secret key is shared among n shareholders and a partial decryption mechanism allows any set of at least k of them to jointly decrypt any ciphertext, which has potentially undergone homomorphic computation. Threshold FHE schemes obtained via Tulva offer the following features:

1. **Compatibility with typical lattice-based FHE:** Tulva can transform any FHE satisfying a structural “linear decryption” property common among lattice-based FHEs into its threshold variant, including, among others, the FHE families of BGV/BFV [BGV12; Bra12; FV12], FHEW/TFHE [DM15; CGGI16], and discrete CKKS [CKKS17] variants [BCKS24].
2. **Asynchronous threshold decryption:** Threshold decryption is fully non-interactive: any shareholder can locally compute a partial decryption of any ciphertext without communicating with, or even being aware of, other shareholders, so shareholders never need to be online simultaneously. Anyone holding honestly generated partial decryptions from any k shareholders can combine them to recover the plaintext.
3. **Security under partial decryptions of challenge ciphertexts from well-studied lattice assumptions:** Assuming that the base FHE scheme is IND-CPA secure, the compiled threshold FHE scheme is proven IND-CPA secure under the Module Learning with Errors (MLWE) assumption [LS15], in the threshold PKE security model of Cini, Lai and Woo [CLW25] (Section 2.3): the message encrypted in any ciphertext remains hidden from an adversary who statically corrupts $f' \leq f = k - 1$ shareholders and obtains fewer than $k - f'$ partial decryptions of this ciphertext from non-corrupted shareholders.
4. **Security with abort:** The extension Tulva⁺ ensures that an honest combiner outputs the correct message or aborts (security with abort), and possibly recovers it despite incorrect partial decryptions (robustness), see Tulva⁺ Extension.
5. **Performance preservation:** Tulva is split into an FHE layer, handled entirely by the base FHE scheme, and a threshold layer built on the threshold PKE scheme Pilvi [CLW25]. A ciphertext-only preprocessing key-switches a base FHE ciphertext into a ciphertext of the threshold PKE, so compiled schemes inherit the ciphertext format and the costs of encryption and homomorphic evaluation of the base scheme.
6. **Flexible threshold profiles:** Over the cyclotomic ring of power-of-two conductor $\mathfrak{f}$, Tulva supports any threshold profile (k, n) with n dividing $\mathfrak{f}$ and corruption threshold $f = k - 1$, at a cost that grows roughly with $\min(k, n - k) \cdot \log n$. Subject to the above divisibility condition, our example parameter sets cover any $k \leq n \leq 128$, and any $n \leq 512$ with $\min(k, n - k) \leq 16$ (Table 1).
7. **Small decryption shares:** A partial decryption is a single element of the ciphertext ring, of size $\varphi \log q$ bits for the ring degree $\varphi = \mathfrak{f}/2$, independent of the ciphertext dimension of the base FHE scheme. Its size ranges from 6.6 to 12 KiB for the example parameter sets with $\varphi = 256$ (Table 1).

2. Specification

2.1. Organisation

Building blocks. Tulva is instantiated over a cyclotomic ring $\mathcal{R} = \mathbb{Z}[\zeta]$ of power-of-two conductor $\mathfrak{f}$ and degree $\varphi = \mathfrak{f}/2$, with plaintext modulus p and ciphertext modulus q , $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$. The ring $\mathcal{R}$ admits a ξ -subtractive set $\Xi = \{\zeta^{i\mathfrak{f}/n} : i \in [n]\}$ [AL21] for any $n \mid \mathfrak{f}$, i.e. for any k -subset $T \subseteq \Xi$ the Vandermonde matrix $\mathbf{V}_T$ of T satisfies $\xi \cdot \mathbf{V}_T^{-1} \in \mathcal{R}^{k \times k}$, where $\xi \in \mathcal{R}$ depends on k . The elements of Ξ serve as evaluation points of Shamir secret sharing.

Tulva requires as building blocks two hash functions $H_{\text{tag}} : \{0, 1\}^* \rightarrow \mathcal{R}_q^h$ and $H_{\text{pk}} : \{0, 1\}^* \rightarrow \mathcal{R}_q^{d \times m}$, where the tag length h , the threshold-layer LWE dimension d and the gadget-decomposition length m are parameters of Tulva (Table 1), and a base FHE scheme Π . The only requirement on Π is a structural “linear decryption” property: (1) its decryption key is a vector $\hat{\mathbf{s}} \in \mathcal{R}_q^d$, and (2) except with probability $\varepsilon_\Pi(\lambda)$ over the randomness of key generation and encryption, its ciphertexts $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1)$ satisfy, for a slack factor $\xi \in \mathcal{R}$ and an error bound B ,

$$\hat{\mathbf{c}}_1 - \hat{\mathbf{s}}^T \cdot \hat{\mathbf{c}}_0 = \xi^{-2} \cdot \lfloor q/p \rfloor \cdot \mu + \hat{e} \bmod q \text{ with } \|\hat{e}\| \leq B. \quad (1)$$

Many lattice-based FHE schemes naturally have this property; for the others, the [Preprocessing Module](#) extends the support of Tulva to **any** FHE scheme.

Preprocessing Module. Let Π' be an arbitrary conventional FHE scheme and $\bar{\Pi}$ be a scheme with the linear decryption property. We create a new scheme Π that inherits all operations from Π' , except: Π .Setup outputs the public parameters of both schemes; Π .KGen runs both Π' .KGen and $\bar{\Pi}$.KGen, and the public key additionally contains an $\bar{\Pi}$ -encryption $\overline{\text{ctxt}}_{\text{sk}'}$ of the secret key of Π' ; for an Π' ciphertext ctxt encrypting some message μ , Π .Preproc(ctxt) homomorphically evaluates the decryption circuit Π' .Dec($\cdot, \text{ctxt}$) on $\overline{\text{ctxt}}_{\text{sk}'}$ using $\bar{\Pi}$.Eval, resulting in an $\bar{\Pi}$ ciphertext $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1)$ of the same message μ ; and Π .Dec runs Π .Preproc(ctxt) and decrypts the result under $\hat{\mathbf{s}}$, i.e. it recovers μ from Eq. (1). The only non-trivial part is Π .Preproc, which can be efficiently instantiated for all practical FHE schemes, as we will demonstrate for some of the most common ones (e.g. BGV/BFV, FHEW/TFHE and discrete CKKS). We require the message μ to be well-defined, which is a problem for approximate schemes [CKKS17] that mix noise and message, even for their conventional decryption [LM21]; the solutions [LMSS22; BCDKLT25] are orthogonal to the threshold setting.

Core Module. The core module of Tulva consists of the algorithms Setup, KGen, Enc, Eval, ParDec and Combine of an asynchronous threshold FHE scheme:

- Setup, Enc and Eval simply run the corresponding algorithms of Π .
- Key generation (KGen) runs Π .KGen to obtain $(\text{pk}_\Pi, \text{sk}_\Pi = \hat{\mathbf{s}})$, samples a fresh secret $\mathbf{r}$ and publishes an MLWE encryption $(\mathbf{A}, \mathbf{b})$ of $\hat{\mathbf{s}}$ under $\mathbf{r}$, with $\mathbf{A} := H_{\text{pk}}(\text{seed})$ for a random seed, $\mathbf{b}^T := \mathbf{r}^T \cdot \mathbf{A} + \mathbf{e}^T + \mathbf{z}^T \bmod q$, discrete Gaussian noise $\mathbf{e}$, and an encoding $\mathbf{z}$ of $\hat{\mathbf{s}}$ padded with h zeros. The public key is $\text{pk} = (\text{pk}_\Pi, \mathbf{A}, \mathbf{b})$, and the share $\text{sk}_i = \mathbf{s}_i \in \mathcal{R}_q^d$ of shareholder i is a Shamir share of $\mathbf{r}$ at evaluation point $\zeta^{i\mathfrak{f}/n} \in \Xi$, scaled by ξ .
- Partial decryption (ParDec) computes an inner product of the key share with a public re-encoding of the ciphertext, plus masking noise. It runs Π .Preproc(ctxt) to obtain $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1)$,

derives the ciphertext tag $\mathbf{t} := H_{\text{tag}}(\text{ctxt}) \in \mathcal{R}_q^h$, gadget-decomposes $(\hat{\mathbf{c}}_0, \mathbf{t})$ into a short vector $\mathbf{x}$, and forms the threshold-layer component $\mathbf{c}_0 := \mathbf{A} \cdot \mathbf{x} \bmod q$. It samples a discrete Gaussian masking noise e_i and outputs $\text{pd}_i := \mathbf{s}_i^T \cdot \mathbf{c}_0 + e_i \bmod q$.

- **Combination (Combine)** applies Shamir reconstruction to the k partial decryptions of a recovery set T and rounds. It recomputes $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1)$, $\mathbf{t}$ and $\mathbf{x}$ as in ParDec, sets $c_1 := \xi^2 \cdot \mathbf{b}^T \cdot \mathbf{x} \bmod q$, reconstructs $y = \xi^2 \cdot \mathbf{r}^T \cdot \mathbf{c}_0 + \text{noise from } (\text{pd}_i)_{i \in T}$ using the integral matrix $\xi \cdot \mathbf{V}_T^{-1}$, and recovers $\mu' := \lfloor p/q \cdot (\xi^2 \cdot \hat{c}_1 - (c_1 - y)) \rfloor \bmod p$.

Supporting Algorithms. We will include algorithms for uniform and discrete Gaussian sampling and for (de-)serialising public parameters, public keys, secret key shares, ciphertexts, partial decryptions and messages.

Reference Compilation. We plan to include the full specification and implementation of at least one compiled scheme, i.e. Tulva applied to a concrete FHE with its preprocessing module (Section 3), and the Tulva⁺ extension which achieves security with abort.

Use Case Demonstration. We plan to provide, as a concrete use case of Tulva, the homomorphic evaluation of the NIST lightweight cipher Ascon [DEMS21; NIST-SP800-232], together with an end-to-end benchmark.

2.2. System Model

Threshold profiles. Tulva requires n to be a divisor of f but places no further restriction on (k, n) . Applications requiring a non-power-of-two number of shareholders can round this number up to the smallest supported power of two and leave the additional secret key shares unassigned. Its parameters and costs depend on the threshold profile (k, n) but not on the corruption threshold, which is $f = k - 1$ for every profile; for correct decryption, $n - f \geq k$ is a trivial necessary condition. The example parameter sets of Table 1 roughly correspond to the threshold profiles (2), (3), (S), (M) of the NIST Threshold Call [NIST-IR8214C, Table 17] with any corruption proportion, and to profile (L) with restricted corruption proportion.

Trusted setup. We assume a trusted party runs KGen and distributes the secret key shares honestly; a distributed key generation (DKG) protocol is possible but out of scope of this package.

Network model. Tulva works in a completely asynchronous setting. The only messages of threshold decryption are the partial decryptions, each sent from a shareholder to whoever runs Combine over a confidential and authenticated channel (e.g. TLS); no broadcast channel is used. The adversary may delay, reorder or drop these messages, which affects only whether the combiner eventually receives k partial decryptions. For confidentiality, shareholders are assumed to run ParDec only on honestly generated or evaluated ciphertexts, as in the security model of Section 2.3; lifting this assumption requires ciphertexts to come with a proof of integrity, i.e. a verifiable FHE scheme, which is out of scope of this package. Correctness of Tulva assumes that the k partial decryptions used by the combiner are honestly generated; Tulva⁺ removes this assumption.

Ciphertext preprocessing. The preprocessing $\Pi.\text{Preproc}(\text{ctxt})$ run by ParDec and Combine is public and deterministic, so it can be delegated to a third party trusted to compute it correctly,

e.g. the server responsible for homomorphic computation; auditing such a delegation, e.g. by proofs of integrity of homomorphic computation, is out of scope of this package.

2.3. Security

Formulation of security. In our IND-CPA security model for threshold FHE, we consider an adversary who can 1) statically corrupt a set $\mathcal{C} \subseteq [n]$ of $f' := |\mathcal{C}| \leq f = k - 1$ shareholders, 2) request ciphertexts of specified messages and homomorphic evaluations of specified functions on honestly generated/evaluated ciphertexts, and 3) request partial decryptions of at most Q honestly generated/evaluated ciphertexts, subject to no trivial decryption of challenge ciphertexts. A threshold FHE is Q -bounded IND-CPA secure if, for every ciphertext for which strictly fewer than $k - f'$ partial decryptions generated by non-corrupted shareholders are given to the adversary, the encrypted message remains hidden. This is stronger [CLW25; BKW26] than models where the message is only guaranteed to remain hidden for ciphertexts of which no partial decryption is given to the adversary [BGGJKRS18; BS23; KLLPS26]. The model does not cover adaptively corrupted shareholders. We do not know how to prove security in their presence, beyond the generic but costly technique of complexity leveraging, but we are not aware of any attack that exploits it.

Assumptions and security strength estimation. We will provide a formal proof that any threshold FHE scheme obtained via Tulva is Q -bounded IND-CPA secure assuming that the underlying FHE scheme is IND-CPA secure, the MLWE assumption holds, and H_{tag} and H_{pk} are non-programmable random oracles, which are used only to derive the ciphertext tags and the public matrix $\mathbf{A}$. The MLWE parameters depend on the target security strength λ , the threshold profile (k, n) and the bound Q ; we will provide parameters for $\lambda \in \{128, 192, 256\}$ bits, $Q = 2^{64}$ and different choices of (k, n) , with security strength estimated using the Lattice Estimator [APS15].

Tulva⁺ Extension. In Tulva, Combine reconstructs the message linearly (modulo rounding) from the k partial decryptions of a recovery set without verifying them, so a single incorrect one may shift the output arbitrarily and undetectably. The package will also include Tulva⁺, which guarantees at least **security with abort** (i.e. an honest combiner outputs the correct message or aborts), and possibly also robustness (i.e. it recovers the correct message from any k correct partial decryptions, even if interleaved with incorrect ones). We will compare three approaches, and specify, prove secure and implement at least one of them:

1. *Proofs of integrity:* each partial decryption carries a zero-knowledge proof that it was computed correctly, and the combiner uses the first k with valid proofs. This achieves robustness at the cost of larger and slower partial decryptions.
2. *Homomorphic authentication:* Preproc also homomorphically evaluates a message authentication code or signature keyed by material encrypted in the public key (e.g. via a pseudorandom function), and the combiner releases the message only if the recovered tag is valid. This gives security with abort at the cost of a slower Preproc.
3. *Consistency check:* the combiner collects $k + f$ partial decryptions and aborts unless they are consistent, up to noise, with a single sharing. This gives security with abort at the cost of requiring $n \geq k + f$ (an honest majority; $k \leq (n + 1)/2$ for $f = k - 1$). Liveness (an honest combiner terminates even if corrupted shareholders withhold their partial decryptions) further requires $n \geq k + 2f$ (an honest supermajority; $k \leq (n + 2)/3$ for $f = k - 1$).

After an abort in Approach 2 or 3, the combiner can optionally request proofs of integrity, exclude the shareholders that fail to provide one, and retry.

3. Open-Source Implementation

Code structure. We will provide a complete implementation of Tulva in C and Python, in two versions: a **portable** version and an **optimised** version with architecture-specific optimisations for NIST’s reference platform. The submission package will include the threshold decryption protocols of Tulva and Tulva⁺; three versions of Π .Preproc (one specific to levelled-HE use cases, and two generic, one latency-optimised and one communication-optimised); and the Ascon use case.

Code progress and availability. This package is based on a paper [CGLW27] accepted to IEEE S&P 2027, which reports a prototype implementation and its benchmarks; the full version of the paper and the implementation will be made public in the near future.

Implementation of the networking model. Full network support is built upon Python’s socket library, with an optional latency-emulation parameter for local testing.

Testing. The implementation will support deterministic execution as a test mode: all randomness, in key generation, encryption and the masking noise e_i of ParDec, is then derived from a seed, so that the test vectors we will provide and debugging runs are reproducible; in regular operation these algorithms draw fresh randomness. Tests with misbehaving shareholders check that malformed partial decryptions do not crash the implementation, and that Tulva⁺ never outputs an incorrect message: it either aborts or recovers the correct one. For adverse networking conditions, tests check that honest partial decryptions may arrive in arbitrary order and that Combine recovers the message once k (or $k + f$ in Approach 3) of them have arrived.

4. Experimental Performance Evaluation

Performance Evaluation. Example parameters targeting 128-bit security for our preliminary implementation of `tulva-tfhe`, the threshold FHE scheme obtained by compiling TFHE [CGGI16], and its performance with these parameters on an AWS `c7a.32xlarge` instance are reported in Tables 1 and 2. Our formal submission will evaluate the **optimised** implementation on NIST’s reference machine [NIST-IR8214C], focusing on the overhead added by Tulva to the baseline FHE schemes and use cases.

5. Licensing, Patent Claims, and Funding

Licensing. Our code will be licensed under the Apache 2.0 License and will only rely on third-party libraries under OSI-approved licences.

Patent Claims. No team member claims any patent rights over any part of the proposal or software. To the best of our knowledge, no techniques or implementations included in this proposal are covered by existing patents.

Table 1: Parameter sets of tulva-tfhe (S: small ring, L: large ring).

Set	n	w	λ	q	d	h	m	χ	$\bar{\chi}$	$ \text{pk} $ [KiB]	$ \text{pd} $ [KiB]
S.1	8	8	131	2^{210}	26	1	45	2^{191}	2^{17}	295	6.6
S.2	32	16	133	2^{235}	34	3	55	2^{203}	2^4	404	7.3
S.3	128	16	129	2^{308}	45	9	119	2^{246}	2^7	1145	9.6
S.4	128	64	130	2^{368}	54	16	192	2^{290}	2^7	2208	11.5
S.5	512	16	128	2^{384}	56	17	200	2^{291}	2^7	2400	12.0
L.1	8	8	141	2^{215}	4	1	45	2^{194}	2^{12}	2419	53.8
L.2	32	16	141	2^{245}	5	1	45	2^{211}	2^3	2756	61.2
L.3	128	16	134	2^{315}	6	2	70	2^{252}	2^5	5513	78.8
L.4	128	64	131	2^{376}	7	2	80	2^{297}	2^7	7520	94.0
L.5	512	16	144	2^{392}	8	3	88	2^{297}	2^6	8624	98.0

All sets: $Q = 2^{64}$, $p = 2^4$; ring degree $\varphi = 256$ (S) and 2048 (L). n : number of shareholders; w : window size (the set covers $k \in [1, w] \cup [n - w, n]$, at the cost of its most expensive profile); λ : estimated security strength [bit]; q : ciphertext modulus; d : threshold-layer LWE dimension; h : tag length; m : gadget-decomposition length; $\chi, \bar{\chi}$: Gaussian widths of the ParDec masking noise and the KGen noise; $|\text{pk}|, |\text{pd}|$: sizes of the public key and of one partial decryption.

Table 2: Runtimes of tulva-tfhe for the parameter sets of Table 1.

Set	k/n	$ \mu $ [bit]	Preproc [s]	ParDec [ms]	Combine [μ s]
S.1	4/8	4 1024	0.40 20.80	1.00 1.13	40 40
S.2	16/32	4 1024	0.40 19.94	1.71 2.27	186 176
S.3	16/128	4 1024	0.60 21.20	10.52 11.15	319 326
S.4	64/128	4 1024	0.69 22.56	25.40 23.18	4062 3761
S.5	16/512	4 1024	0.69 23.19	25.07 25.35	989 1003
L.1	4/8	8192	181.00	2.96	323
L.2	16/32	8192	173.58	3.68	1650
L.3	16/128	8192	183.51	8.52	1176
L.4	64/128	8192	194.03	11.95	36744
L.5	16/512	8192	192.82	14.09	2393

k/n : threshold profile; $|\mu|$: plaintext size per decryption, where in the S rows an entry $a | b$ of a timing column gives the values for $|\mu| = 4 | 1024$ bits; Preproc, ParDec, Combine: median time of the ciphertext preprocessing, of one partial decryption, and of combining k partial decryptions, using TFHE bootstrapping [CGGI16].

Funding. Russell W. F. Lai and Michał Osadnik’s work was supported by the Research Council of Finland grant 358951 and the Future Makers Program project SPICED of the Technology Industries of Finland Centennial Foundation. Ivy K. Y. Woo’s work was supported by the Research Council of Finland grant 358950. Antonio Guimarães’ work was supported by grants JDC2024-055789-I and CEX2024-001471-M funded by MICIU/AEI/10.13039/501100011033 and ESF+. Valerio Cini’s work was supported by the European Union’s Horizon Europe research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101202597 (ACryL).

6. Comparison with Other Submissions

Zama [Zam26] and PANTHERIA [PAN26] both build their proposals around specific conventional FHE schemes, for which they provide threshold decryption based on MPC protocols. In contrast, Tulva is a **black-box compiler** for asynchronous (single-round) thresholdisation of *any* conventional scheme. In more detail:

Generality. Tulva only specifies a high-level interface [NIST-IR8214C, Requirement 11.4]. Our [Preprocessing Module](#) can homomorphically evaluate the decryption circuit of any base scheme, providing support to any FHE scheme. To our knowledge this procedure is efficient for all modern lattice-based schemes, and in particular for all schemes covered by Zama and PANTHERIA. The compiled scheme relies on the security of conventional decryption of the base scheme, so approximate schemes may inherit the leakage of [LM21], and one should use discretised [BCKS24] or noise-flooded [LMSS22; BCDKLT25] variants. The internal specifications of the base scheme (e.g., message encoding and HE evaluation procedures) are irrelevant.

Interaction. Tulva decryption, and that of Tulva⁺ with Approach 1 or 2 of [Tulva⁺ Extension](#), is a single-round protocol for every supported profile, with no offline phase, no predefined participating set, and corruption threshold $f = k - 1$. With Approach 3, decryption is still single-round, but the combiner needs $k + f$ partial decryptions, so correctness against $f = k - 1$ corruptions requires $n \geq 2k - 1$. PANTHERIA's noise-flooding decryption requires the set of participating parties to be known beforehand, and its active-security decryption is a multi-round MPC protocol [ZZLP25]. Nexus avoids set agreement only under an honest supermajority $f < n/3$ and by precomputing shared randomness for all possible sets, at a decryption cost growing with $\binom{n}{f}$, which is only practical for few parties; beyond that it relies on MPC. Single-round decryption is particularly important in real-world scenarios such as confidential smart-contract platforms [Zam23; Fhe24] and the generic thresholdisation of other primitives via threshold FHE [BGGJKRS18] (e.g. non-interactive threshold signatures).

Security model. Tulva is proven Q -bounded IND-CPA secure in a game-based model that allows partial decryptions of challenge ciphertexts; the other submissions use UC models each with different specifications, and we are not aware of results relating any of these models. For correctness against active corruptions, Tulva⁺ targets at least security with abort, possibly robustness ([Tulva⁺ Extension](#)).

Complementarity. With Approach 2 or 3, which give security with abort, Tulva⁺ could replace PANTHERIA's MPC-based threshold decryption when that is instantiated with abort. With Approach 1, which gives robustness, it could further replace Zama's Nexus. Either replacement brings single-round asynchronous decryption to the threshold profiles where they lack it (with Approach 3, only for $n \geq k + f$). It is up to the application, however, to define whether asynchronicity is a requirement.

References

- [AL21]** Martin R. Albrecht and Russell W. F. Lai. “Subtractive Sets over Cyclotomic Rings - Limits of Schnorr-Like Arguments over Lattices”. In: *CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Cham, August 2021, pp. 519–548. DOI: [10.1007/978-3-030-84245-1_18](https://doi.org/10.1007/978-3-030-84245-1_18). Also at ia.cr/2021/202.
- [APS15]** Martin R. Albrecht, Rachel Player, and Sam Scott. “On the concrete hardness of Learning with Errors”. In: *J. Math. Cryptol.* 9.3 (2015), pp. 169–203. DOI: [10.1515/jmc-2015-0016](https://doi.org/10.1515/jmc-2015-0016). Also at ia.cr/2015/046.
- [BCDKLT25]** Flávio Bergamaschi, Anamaria Costache, Dana Dachman-Soled, Hunter Kippen, Lucas LaBuff, and Rui Tang. “Revisiting the Security of Approximate FHE with Noise-Flooding Countermeasures”. In: *PKC 2025, Part V*. Ed. by Tibor Jager and Jiaxin Pan. Vol. 15678. LNCS. Springer, Cham, May 2025, pp. 102–134. DOI: [10.1007/978-3-031-91832-2_4](https://doi.org/10.1007/978-3-031-91832-2_4).
- [BCKS24]** Youngjin Bae, Jung Hee Cheon, Jaehyung Kim, and Damien Stehlé. “Bootstrapping Bits with CKKS”. In: *EUROCRYPT 2024, Part II*. Ed. by Marc Joye and Gregor Leander. Vol. 14652. LNCS. Springer, Cham, May 2024, pp. 94–123. DOI: [10.1007/978-3-031-58723-8_4](https://doi.org/10.1007/978-3-031-58723-8_4). Also at ia.cr/2024/767.
- [BGGJKRS18]** Dan Boneh, Rosario Gennaro, Steven Goldfeder, Aayush Jain, Sam Kim, Peter M. R. Rasmussen, and Amit Sahai. “Threshold Cryptosystems from Threshold Fully Homomorphic Encryption”. In: *CRYPTO 2018, Part I*. Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10991. LNCS. Springer, Cham, August 2018, pp. 565–596. DOI: [10.1007/978-3-319-96884-1_19](https://doi.org/10.1007/978-3-319-96884-1_19). Also at ia.cr/2017/956.
- [BGV12]** Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. “(Leveled) fully homomorphic encryption without bootstrapping”. In: *ITCS 2012*. Ed. by Shafi Goldwasser. ACM, January 2012, pp. 309–325. DOI: [10.1145/2090236.2090262](https://doi.org/10.1145/2090236.2090262). Also at ia.cr/2011/277.
- [BKW26]** Chris Brzuska, Michael Klooß, and Ivy K. Y. Woo. “Threshold Public-Key Encryption: Definitions, Relations, and CPA-to-CCA Transforms”. In: *PKC 2026, Part IV*. Ed. by Shi Bai and Edoardo Persichetti. Vol. 16554. LNCS. Springer, Cham, May 2026, pp. 162–197. DOI: [10.1007/978-3-032-26740-5_6](https://doi.org/10.1007/978-3-032-26740-5_6). Also at ia.cr/2025/1665.
- [Bra12]** Zvika Brakerski. “Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP”. In: *CRYPTO 2012*. Ed. by Reihaneh Safavi-Naini and Ran Canetti. Vol. 7417. LNCS. Springer, Berlin, Heidelberg, August 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.
- [BS23]** Katharina Boudgoust and Peter Scholl. “Simple Threshold (Fully Homomorphic) Encryption from LWE with Polynomial Modulus”. In: *ASIACRYPT 2023, Part I*. Ed. by Jian Guo and Ron Steinfeld. Vol. 14438. LNCS. Springer, Singapore, December 2023, pp. 371–404. DOI: [10.1007/978-981-99-8721-4_12](https://doi.org/10.1007/978-981-99-8721-4_12). Also at ia.cr/2023/016.

[CGGI16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. “Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds”. In: *ASIACRYPT 2016, Part I*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10031. LNCS. Springer, Berlin, Heidelberg, December 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.

[CGLW27] Valerio Cini, Antonio Guimarães, Russell W. F. Lai, Michał Osadnik, and Ivy K. Y. Woo. “Tulva: Practical Asynchronous Threshold FHE”. In: *2027 IEEE Symposium on Security and Privacy*. To appear. 2027.

[CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. “Homomorphic Encryption for Arithmetic of Approximate Numbers”. In: *ASIACRYPT 2017, Part I*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10624. LNCS. Springer, Cham, December 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.

[CLW25] Valerio Cini, Russell W. F. Lai, and Ivy K. Y. Woo. “Pilvi: Lattice Threshold PKE with Small Decryption Shares and Improved Security”. In: *ASIACRYPT 2025, Part VI*. Ed. by Goichiro Hanaoka and Bo-Yin Yang. Vol. 16250. LNCS. Springer, Singapore, December 2025, pp. 539–569. DOI: [10.1007/978-981-95-5119-4_17](https://doi.org/10.1007/978-981-95-5119-4_17). Also at ia.cr/2025/1691.

[DEMS21] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. “Ascon v1.2: Lightweight Authenticated Encryption and Hashing”. In: *Journal of Cryptology* 34.3 (July 2021), p. 33. DOI: [10.1007/s00145-021-09398-9](https://doi.org/10.1007/s00145-021-09398-9).

[DM15] Léo Ducas and Daniele Micciancio. “FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second”. In: *EUROCRYPT 2015, Part I*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9056. LNCS. Springer, Berlin, Heidelberg, April 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.

[Fhe24] Fhenix. *Fhenix: Private DeFi on Ethereum with Fully Homomorphic Encryption*. <https://www.fhenix.io/>. Accessed: 2026-06-09. 2024.

[FV12] Junfeng Fan and Frederik Vercauteren. *Somewhat Practical Fully Homomorphic Encryption*. Cryptology ePrint Archive, Report 2012/144. 2012. URL: <https://eprint.iacr.org/2012/144>.

[KLLPS26] Won Kim, Changmin Lee, JeongHwan Lee, Alain Passelègue, and Damien Stehlé. “Asynchronous Lagrange-Based Threshold FHE with Smaller Modulus Overhead”. In: *CRYPTO 2026*. 2026. 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.

[LM21] Baiyu Li and Daniele Micciancio. “On the Security of Homomorphic Encryption on Approximate Numbers”. In: *EUROCRYPT 2021, Part I*. Ed. by Anne Canteaut and François-Xavier Standaert. Vol. 12696. LNCS. Springer, Cham, October 2021, pp. 648–677. DOI: [10.1007/978-3-030-77870-5_23](https://doi.org/10.1007/978-3-030-77870-5_23).

[LMSS22] Baiyu Li, Daniele Micciancio, Mark Schultz, and Jessica Sorrell. “Securing Approximate Homomorphic Encryption Using Differential Privacy”. In: *CRYPTO 2022, Part I*. Ed. by Yevgeniy

Dodis and Thomas Shrimpton. Vol. 13507. LNCS. Springer, Cham, August 2022, pp. 560–589. DOI: [10.1007/978-3-031-15802-5_20](https://doi.org/10.1007/978-3-031-15802-5_20).

[LS15] Adeline Langlois and Damien Stehlé. “Worst-case to average-case reductions for module lattices”. In: *DCC* 75.3 (2015), pp. 565–599. DOI: [10.1007/s10623-014-9938-4](https://doi.org/10.1007/s10623-014-9938-4). Also at ia.cr/2012/090.

[PAN26] PANTHERIA Threshold FHE Team. *PANTHERIA: Threshold FHE for RLWE-Based Cryptosystems*. Preview Writeup (version 1.0) submitted to the NIST First Call for Multi-Party Threshold Schemes. January 2026. URL: <https://csrc.nist.gov/events/2026/mpts2026>.

[Zam23] Zama. *fhEVM: Confidential EVM Smart Contracts using Fully Homomorphic Encryption*. Whitepaper, <https://github.com/zama-ai/fhevm/blob/main/fhevm-whitepaper.pdf>. Accessed: 2026-06-09. 2023.

[Zam26] Zama. *TFHE, ZHENith and Nexus: A Suite of Cryptosystems to Enable Fully Homomorphic Encryption Applications*. Preview Writeup (version 0.1) submitted to the NIST First Call for Multi-Party Threshold Schemes. January 2026. URL: <https://csrc.nist.gov/events/2026/mpts2026>.

[ZZLP25] Guy Zyskind, Doron Zarchy, Max Leibovich, and Chris Peikert. “High-Throughput Universally Composable Threshold FHE Decryption”. In: *ACM CCS 2025*. Ed. by Chun-Ying Huang, Jyh-Cheng Chen, Shiuh-Pyng Shieh, David Lie, and Véronique Cortier. ACM Press, October 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).

[NIST-SP800-232] Meltem Sönmez Turan, Kerry McKay, Jinkeon Kang, John Kelsey, and Donghoon Chang. *Ascon-Based Lightweight Cryptography Standards for Constrained Devices: Authenticated Encryption, Hash, and Extendable Output Functions*. (National Institute of Standards and Technology) NIST Special Publication (SP) 800-232. August 2025. DOI: [10.6028/NIST.SP.800-232](https://doi.org/10.6028/NIST.SP.800-232).