(In)Security of Threshold Fully Homomorphic Encryption based on Shamir Secret Sharing

Wonhee Cho[1], Jiseung Kim[2], and Changmin Lee[3]

[1] Seoul National University
wony0404@snu.ac.kr
[2] Jeonbuk National University
jiseungkim@jbnu.ac.kr
[3] Korea Institute for Advanced Study
changminlee@kias.re.kr

Abstract. Boneh et al. (CRYPTO'18) proposed two t-out-of-N threshold fully homomorphic encryption (TFHE) schemes based on Shamir secret sharing scheme and $\{0,1\}$-linear secret sharing scheme. They demonstrated the simulation security, ensuring no information leakage during partial or final decryption. This breakthrough allows any scheme to be converted into a threshold scheme by using TFHE.
We propose two polynomial time algorithms to break the simulation security of t-out-of-N TFHE based on Shamir secret sharing scheme proposed by Boneh et al.. First, we show that an adversary can break the simulation security by recovering the secret key under some constraints on t and N, which does not violate the conditions for security proof. Next, we introduce a straightforward fix that theoretically satisfies the simulation security. However, we argue that this modification remains insecure insecure when implemented with any state-of-the-art fully homomorphic encryption libraries in practice. To ensure robustness against our subsequent attacks, we recommend using an error-refreshing algorithm, such as bootstrapping or modulus switching, for each addition operation.

Keywords: Threshold Fully Homomorphic Encryption, Shamir secret sharing scheme, Cryptanalysis

1 Introduction

Threshold cryptography [28,30,33] has been considered as one of the fundamental foundations in cryptography. It splits the secret key into N secret shares, with each share stored on a different server. A t-out-of-N threshold access structure ensures that at least t shares are required to decrypt a ciphertext, while any set of $t-1$ shares reveals no information about the secret.

Over the past few decades, many threshold signatures, and encryptions have been proposed. Most of them are built on the pre-quantum hardness such as factoring [26, 28, 33, 37, 46] and discrete logarithm [15, 16, 22, 24, 25, 31, 34–36, 40, 42, 47].

Recently, due to the growing threat of quantum computing, there have been several researches to construct post-quantum threshold cryptosystems, particularly lattice-based encryptions [7,10,11,18,23,43], and signatures [4,8,27,29,39]. Many of these lattice-based threshold primitives are built from lattice-friendly secret sharing schemes such as Shamir secret sharing [45], $\{0,1\}$-linear secret sharing ($\{0,1\}$-LSSS) [10], Tree secret sharing scheme (TreeSSS) [18] and pseudorandom secret sharing [11,21].

In their pioneering work on one-round threshold primitives, Boneh et al. [10] introduced the concept of a universal thresholdizer, demonstrating that the existence of threshold fully homomorphic encryption (TFHE) enables the transformation of any scheme into a threshold scheme. They also proposed two one-round TFHE schemes by combining secure fully homomorphic encryptions with Sharmir secret sharing and $\{0,1\}$-LSSS, respectively. Following this, Boudgoust and Scholl [11] proposed selectively secure TFHE constructions using $\{0,1\}$-LSSS and pseudorandom secret sharing, respectively. Cheon, Cho and Kim [18] later suggested a communication efficient one round TFHE construction based on a new secret sharing scheme, called TreeSSS.

Each TFHE construction should satisfy simulation security beyond the semantic security. Informally, the simulation security ensures that no information about the key shares or messages should leak during partial or final decryption, except for what is inherently revealed by the results of the homomorphic operations.

1.1 This work

This paper presents two types of polynomial time algorithms for breaking the simulation security of t-out-of-N TFHE based on Shamir secret sharing scheme in [10, Sec. 5.3]. We state the first result independent to underlying FHE schemes.

Theorem 1.1 (Informal) *Let FHE be a fully homomorhpic encryption that satisfies the following decryption: For a ciphertext ct of message m and a secret key sk, a decryption algorithm consists of two steps*[4]*:*

1. *Compute* $\langle \mathsf{ct}, \mathsf{sk} \rangle \bmod q = \lfloor \frac{q}{2} \rceil m + (N!)^2 \cdot e$.[5]
2. *Recover m from* $\langle \mathsf{ct}, \mathsf{sk} \rangle \bmod q$

where $|e| \leq B$ for some bound B.

Let TFHE be a t-out-of-N threshold fully homomorphic encryption based on FHE and Shamir secret sharing as in [10, Sec. 5.3]. Then, the following holds:

- *If $B \leq N^{\frac{7}{10}\frac{t}{\log t}}$, then one can (heuristically) break the simulation security of TFHE in polynomial time with probability at least 1/2.*

[4] Boneh et al. [10] argued that certain well-known FHE schemes [13, 14, 38] can be adapted to satisfy the properties.

[5] We mainly describe a scale-invariant style decryption, but our attack also works for BGV-style decryption that $\langle \mathsf{ct}, \mathsf{sk} \rangle \bmod q = m + p \cdot e$ for some prime $p \ll q$.

– *If* $\sqrt{t+1} \cdot B \leq N^{\frac{2t}{5}}$ *and* $2t \leq N$*, then one can (heuristically) break the simulation security of* TFHE *in polynomial time with probability at least 1/2.*

The first result is quite impressive, but there is a quick fix to make it robust against our attack. By utilizing FHE scheme such that

$$\langle \mathsf{ct}, \mathsf{sk} \rangle \bmod q = \lfloor \frac{q}{2} \rceil m + (N!)^4 \cdot e,$$

the attack in Theorem 1.1 is no longer applicable. Furthermore, following the original proof, the modified scheme achieves the simulation security.

The second result concerns the new TFHE scheme with an error $(N!)^4 \cdot e$. To distinguish it from the original TFHE in [10, Sec. 5.3], we call the modified scheme, TFHE′. We assert that TFHE′ remains insecure when instantiated using state-of-the-art FHE libraries such as OpenFHE, HElib, and SEAL [1–3, 6, 44]. More precisely, if FHE scheme implements homomorphic evaluation for the addition circuit as the ciphertext addition, then TFHE′ built from such a FHE cannot achieve the simulation security. These results reveal a security gap between the theoretical and practical use of current libraries. To mitigate this, we recommend incorporating an error-refreshing algorithm, such as bootstrapping or modulus switching, to ensure robustness against our second attack. However, it is crucial to note that implementing these algorithms could impact performance efficiency.

As a side contribution, the practical instantiation of the universal thresholdizer (UT) based on Shamir's TFHE in [10, Sec. 7] is also vulnerable in a similar manner.

Additionally, we identify a flaw in the proof of the simulation security of TFHE that affects the construction presented in the paper. Specifically, the proof of [10, Thm. 5.14] consists of a sequence of hybrid experiments:

– Game 0: The real game.
– Game 1: Identical to Game 0, except for the partial decryption phases.
– Game 2: The ideal game.

Boneh et al. [10] demonstrated that Game 0 and Game 1 are statistically indistinguishable using the noise smudging lemma [5]. The proof primarily focuses on simulating partial decryptions. However, the partial decryption process in Game 0 differs from that in Game 1 due to the algebraic structure of the error term during partial decryption. Further details are discussed in Section 3.2.

Attack Overview. To describe an idea of our attack, we briefly recap a threshold primitive from Shamir secret sharing. Consider FHE in Theorem 1.1, and let $\mathsf{sk} \in \mathbb{Z}_q^n$ be a secret key of FHE.

During the setup phase of the threshold variant of FHE, denoted by TFHE, a dealer applies Shamir t-out-of-N secret sharing to sk to generate N keys, called $\{\mathsf{sk}_i\}_{i\in[N]}$, where each sk_i is assigned to a user i. This sharing ensures that for any set $S \subset [N]$ of size t, there exist the (public) Lagrange coefficients λ_i^S such that

$$\sum_{i\in S} \lambda_i^S \cdot \mathsf{sk}_i = \mathsf{sk}.$$

The encryption algorithm remains the same as in the original fhe, but the threshold decryption differs significantly: To decrypt a ciphertext ct, each user first computes $\mathsf{p}_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + (N!)^2 \cdot e_i$ for some small e_i. Next, for any subset $S \subset \{1, \ldots, N\}$ of size t, the combiner compute[6]

$$\begin{aligned}\sum_{i \in S} \lambda_i^S \cdot \mathsf{p}_i &= \sum_{i \in S} \lambda_i^S \cdot (\langle \mathsf{ct}, \mathsf{sk}_i \rangle + (N!)^2 \cdot e_i) \bmod q \\ &= \langle \mathsf{ct}, \mathsf{sk} \rangle + \sum_i \lambda_i^S \cdot (N!)^2 \cdot e_i \bmod q \\ &= \lfloor \frac{q}{2} \rceil m + (N!)^2 \cdot e + \sum_i \lambda_i^S \cdot (N!)^2 \cdot e_i \bmod q\end{aligned}$$

where $(N!)^2 \cdot e$ is an error derived from ct. If $(N!)^2 \cdot e + \sum_i \lambda_i^S \cdot (N!)^2 \cdot e_i$ is smaller than $q/4$, then a combiner can decrypt a binary message m from ct by checking the size $[\sum_{i \in S} \lambda_i^S \cdot \mathsf{p}_i]_q$.

In a nutshell, the simulation security of the t-out-of-N TFHE construction of [10] asserts that the error e is not revealed even if the adversary $\mathcal{A}$ has access to $t-1$ secret shares $\{\mathsf{sk}_i\}_{i \in S^*}$, where S^* is a maximal invalid set of size $t-1$. We demonstrate that the adversary can exploit the algebraic structure of $\sum i \in S \lambda_i^S \cdot \mathsf{p}_i$ to recover information. For simplicity, we let $S = \{1, 2, \ldots, t\}$ and $S^* = \{2, \ldots, t\}$ be a maximal invalid set of size $t-1$.

We emphasize that $\mathcal{A}$ knows $\{e_i\}_{i \in S^*}$. Hence, the adversary $\mathcal{A}$ can have access to the following

$$E = (N!)^2 \cdot e + \lambda_1^S \cdot (N!)^2 \cdot e_1 \in \mathbb{Z}$$

by handling the decryption algorithm. In this equation, there are only two unknowns: e and e_1. Given size constraints for e and e_1, these values can be uniquely determined.

Once the attacker $\mathcal{A}$ recovers e from E, they can also compute $\mathsf{ct} - (N!)^2 \cdot e = \langle \mathbf{a}, \mathsf{sk} \rangle$ for some $\mathbf{a} \in \mathbb{Z}_q^n$. By repeating this process for sufficiently many ciphertexts ct_i, the attacker can obtain $\langle \mathbf{a}_i, \mathsf{sk} \rangle$ for some vectors $\mathbf{a}_i$. Using linear algebra, sk can then be recovered through Gaussian elimination, which runs in polynomial time with respect to n. Thus, the remaining challenge is to recover e from E. The detailed algorithm for recovering e will be provided in the main section.

Related Work. Recently, two papers [17, 19] have examined the insecurity of TFHE schemes under the IND-CPA^D model from [41]. Specifically, both papers propose efficient attacks on various FHE schemes, such as BGV [9], BFV [12,32], and Torus-FHE [20]. They further argue that the threshold variants of these FHE schemes would also be vulnerable if the underlying FHE schemes are not secure against their attacks.

In contrast, we present a direct attack on TFHE itself, independent of the security of the underlying FHE scheme.

[6] In the simulation security of TFHE, attackers can access any set of partial decryptions $\{\mathsf{p}_i\}_{i \in S}$. Thus, the combiner can perform the computation.

2 Preliminaries

Notations. Vectors and matrices are represented by bold letters. For any positive integer n, we denote the set $1, \ldots, n$ by $[n]$. For any set S, the notation $x \leftarrow S$ indicates that x is sampled from a uniform distribution over S.

2.1 Shamir Secret Sharing

For any subset $S \subset [N] \cup \{0\}$ of size t, we define the Lagrange coefficient $\lambda_{i,j}^S \in \mathbb{Z}_q$ as follows:

$$\lambda_{i,j}^S = \prod_{\substack{x \in S \\ x \neq i}} \frac{j-x}{i-x}.$$

Theorem 2.1 (Shamir Secret Sharing [45]) *Let $P = \{1, \ldots, N\}$ be a set of participants and t be a threshold. Then, t-out-of-N threshold Shamir Secret Sharing with secret space $\mathbb{Z}_q$ for some prime q satisfies the following:*

- $\mathsf{SS.Share}(w_0)$*: For any input $w_0 \in \mathbb{Z}_q$, a dealer distributes a single element $w_i \in \mathbb{Z}_q$ to each party $i \in P$.*
- $\mathsf{SS.Combine}$*: For every $i, j \in [N] \cup \{0\}$ and set $S \subset [N] \cup \{0\}$ of size $\geq t$, one can efficiently compute the Lagrange coefficients $\lambda_{i,j}^S \in \mathbb{Z}_q$ such that*

$$w_j = \sum_{i \in S} \lambda_{i,j}^S \cdot w_i.$$

For any subset S' of size less than t, the set of shares $\{w_i\}_{i \in S'}$ is indistinguishable from the set of $\{w_i'\}_{i \in S'}$ where $w_i' \leftarrow \mathbb{Z}_q$ for all $i \in S'$.

2.2 Fully Homomorphic Encryption

Here we describe a syntax of LWE-based fully homomorphic encryptions, denoted by FHE, consist of four algorithms $\mathsf{FHE.Setup}, \mathsf{FHE.Enc}, \mathsf{FHE.Eval}$ and $\mathsf{FHE.Dec}$.

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{FHE.Setup}(1^\lambda, 1^d)$: For the security parameter λ and pre-determined depth bound d, returns a pair of keys $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}, m)$: For the public key pk and a message $m \in \{0,1\}$, returns a ciphertext ct.
- $\hat{\mathsf{ct}} \leftarrow \mathsf{FHE.Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$: For the public key pk, a circuit C of depth at most d, and a family of ciphertexts $\{\mathsf{ct}_i\}_{i=1,\ldots,k}$, return an evaluated ciphertext $\hat{\mathsf{ct}}$.
- $\hat{m} \leftarrow \mathsf{FHE.Dec}(\mathsf{sk}, \mathsf{pk}, \hat{\mathsf{ct}})$: For the public key pk and the secret key sk, and a ciphertext $\hat{\mathsf{ct}}$, returns a message $\hat{m}$.

The FHE primitives that we consider should satisfy the following decryption phase, called a special decryption.

Definition 2.2 (Special Decryption) *For a ciphertext modulus q, it holds that*

1. *Compute* $\langle \mathsf{ct}, \mathsf{sk} \rangle \bmod q = \lfloor \frac{q}{2} \rceil m + c \cdot e$
2. *Recover* m *from* $\langle \mathsf{ct}, \mathsf{sk} \rangle \bmod q$.

where $e \in [-B, B]$ *for some noise parameter* B *and multiplicative constant* c*. If a fully homomorphic encryption scheme* FHE *satisfies a special decryption, then we say it a special* FHE*.*

We note that the multiplicative constant c depends on the secret sharing schemes. For example, TFHE from Shamir secret sharing exploits a numerous constant $c = (N!)^2$ and TFHE from $\{0,1\}$-LSSS uses $c = 1$.

Fortunately, Boneh et al. [10] argued that some well-known FHE schemes [13, 14, 38] can be adapted to satisfy the properties. In this case, we sometimes call it Special FHE.

2.3 Threshold Fully Homomorphic Encryption

This section presents a syntax of threshold fully homomorphic encryptions for t-out-of-N access structure, denoted by TFHE. Suppose that $P = \{P_1, \ldots, P_N\}$ be a set of parties.

- $(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_N) \leftarrow \mathsf{TFHE.Setup}(1^\lambda, 1^d)$: For the security parameter λ and pre-determined depth bound d, returns a public key pk and a set of secrets $\mathsf{sk}_1, \ldots, \mathsf{sk}_N$.
- $\mathsf{ct} \leftarrow \mathsf{TFHE.Enc}(\mathsf{pk}, m)$: For the public key pk and a message $m \in \{0,1\}$, returns a ciphertext ct.
- $\hat{\mathsf{ct}} \leftarrow \mathsf{TFHE.Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$: For the public key pk, a circuit C of depth at most d, and a family of ciphertexts $\{\mathsf{ct}_i\}_{i=1,\ldots,k}$, return an evaluated ciphertext $\hat{\mathsf{ct}}$.
- $\mathsf{p}_i \leftarrow \mathsf{TFHE.ParDec}(\mathsf{pk}, \mathsf{sk}_i, \mathsf{ct})$: For the public key pk, a secret key share sk_i and a ciphertext ct, returns a partial decryption p_i, related to a party P_i.
- $\hat{m} \leftarrow \mathsf{TFHE.FinDec}(\mathsf{pk}, J)$: For the public key pk and a set $J = \{\mathsf{p}_i\}_{i \in S}$ for some $S \subset P$, returns a message $\hat{m} \in \{0, 1, \perp\}$.

The detailed construction of TFHE will be deferred in Section 3.1.

3 Threshold Fully Homomorphic Encryption from Shamir Secret Sharing

In this section, we revisit the threshold fully homomorphic encryption scheme based on Shamir's secret sharing and FHE as presented in [10]. We begin by providing a description of the scheme and then point out a flaw in the proof of its simulation security.

3.1 TFHE Construction from FHE and Shamir secret sharing

Let $P = P_1, \ldots, P_N$ be a set of parties, and let FHE be a lattice-based fully homomorphic encryption scheme that satisfies special decryption (Definition 2.2) with a multiplicative constant $(N!)^2$ and a noise bound parameter B. Let SS denote a Shamir secret sharing scheme for a t-out-of-N threshold access structure as described in Theorem 2.1. Based on FHE and SS, a t-out-of-N threshold FHE, denoted by TFHE, can be constructed as follows:

- $\mathsf{TFHE.Setup}(1^\lambda, 1^d)$:
  - Sample $(\mathsf{fhepk}, \mathsf{fhesk}) \leftarrow \mathsf{FHE.Setup}(1^\lambda, 1^d)$
  - Divide fhesk into secret shares $(\mathsf{fhesk}_1, \ldots, \mathsf{fhesk}_N) \leftarrow \mathsf{SS.Share}(\mathsf{fhesk})$
  - Set $\mathsf{pk} = \mathsf{fhepk}$ and $\mathsf{sk}_i = \mathsf{fhesk}_i \in \mathbb{Z}_q^n$ for all i.
- $\mathsf{TFHE.Enc}(\mathsf{pk}, m)$:
  Sample $\mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}, m)$ and return ct.
- $\mathsf{TFHE.Eval}(\mathsf{pk}, C, \{\mathsf{ct}_i\})$:
  Compute $\hat{\mathsf{ct}} \leftarrow \mathsf{FHE.Eval}(C, \{\mathsf{ct}_i\})$ and return $\hat{\mathsf{ct}}$.
- $\mathsf{TFHE.ParDec}(\mathsf{pk}, \mathsf{ct}, \mathsf{sk}_i)$:
  - Sample a noise smudging error $e_i \leftarrow [-B_{\mathsf{sm}}, B_{\mathsf{sm}}]$
  - Compute $\mathsf{p}_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + (N!)^2 \cdot e_i \in \mathbb{Z}_q$ and return p_i.
- $\mathsf{TFHE.FinDec}(\mathsf{pk}, J)$:
  - Given $J = \{\mathsf{p}_i\}_{i \in S}$ for some $S \subset P$, compute $|S|$.
  - If $|S| < t$, then return $\perp$. Otherwise, compute the Lagrange coefficient $\lambda_{i,0}^S$ for all i.
  - Compute $\sum_i \lambda_{i,0}^S \cdot \mathsf{p}_i$ and reconstruct a message m from $\sum_i \lambda_{i,0}^S \cdot \mathsf{p}_i$.

In this context, the noise bound parameter B must satisfy $B \cdot N! + (N!)^3 \cdot N \cdot B_{\mathsf{sm}} \leq q/4$ for correctness, and $B/B_{\mathsf{sm}} = \mathsf{negl}(\lambda)$, where λ is the security parameter. Under these parameter constraints, and assuming the security of both FHE and SS, Boneh et al. proved that TFHE satisfies simulation security, provided that FHE and SS are secure [10, Thm. 5.14]. We define the notion of simulation security in Section 3.2.

Theorem 3.1 ([10, Thm. 5.14]) *Suppose FHE is a fully homomorphic encryption scheme that satisfies special decryption (Definition 2.2) and security, and SS is a Shamir secret sharing scheme (Theorem 2.1). Then, the TFHE scheme in Section 3.1 with parameter B_{sm} such that $B/B_{\mathsf{sm}} = \mathsf{negl}(\lambda)$ satisfies simulation security.*

3.2 Flaw in the security proof of TFHE Shamir secret sharing

This section describes a flaw in the security proof of TFHE from Shamir secret sharing. We first recall the definition of simulation security and the proof of Theorem 3.1.

Definition 3.2 (Simulation Security [10]) *For any adversary $\mathcal{A}$, if there exists a stateful PPT algorithm $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that the following experiments $\mathsf{Exp}_{\mathcal{A},\mathsf{Real}}(1^\lambda, 1^d)$ and $\mathsf{Exp}_{\mathcal{A},\mathsf{Ideal}}(1^\lambda, 1^d)$ are indistinguishable, a TFHE scheme satisfies simulation security with a security parameter λ, depth bound d.*
$\mathsf{Exp}_{\mathcal{A},\mathsf{Real}}(1^\lambda, 1^d)$:

1. The challenger $\mathcal{C}$ runs $(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_N) \leftarrow \mathsf{TFHE.Setup}(1^\lambda, 1^d)$ and sends pk to $\mathcal{A}$.
2. $\mathcal{A}$ outputs a maximal invalid party set[7] $S^* \subseteq \{P_1, \ldots, P_N\}$ and messages $m_1, \ldots, m_k \in \{0,1\}$.
3. $\mathcal{C}$ provides the key $\{\mathsf{sk}_i\}_{i \in S^*}$ and $\{\mathsf{TFHE.Encrypt}(\mathsf{pk}, m_i)\}_{i \in [k]}$ to $\mathcal{A}$.
4. $\mathcal{A}$ issues a polynomial number of adaptive queries of the form $(S \subseteq \{P_1, \ldots, P_N\}$, $C)$ for circuits $C : \{0,1\}^k \to \{0,1\}$ of depth at most d. For each query, $\mathcal{C}$ computes $\hat{\mathsf{ct}} \leftarrow \mathsf{TFHE.Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$ and provides partial decryptions $\{\mathsf{TFHE.PartDec}(\mathsf{pk}, \hat{\mathsf{ct}}, \mathsf{sk}_i)\}_{i \in S}$ to $\mathcal{A}$.
5. At the end of the experiment, $\mathcal{A}$ outputs a distinguishing bit b.

$\mathsf{Exp}_{\mathcal{A},\mathsf{Ideal}}(1^\lambda, 1^d)$:

1. The challenger $\mathcal{C}$ runs $(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_N) \leftarrow \mathcal{S}_1(1^\lambda, 1^d)$ and sends pk to $\mathcal{A}$.
2. $\mathcal{A}$ outputs a maximal invalid party set $S^* \subseteq \{P_1, \ldots, P_N\}$ and messages $m_1, \ldots, m_k \in \{0,1\}$.
3. $\mathcal{C}$ provides the key $\{\mathsf{sk}_i\}_{i \in S^*}$ and $\{\mathsf{TFHE.Encrypt}(\mathsf{pk}, m_i)\}_{i \in [k]}$ to $\mathcal{A}$.
4. $\mathcal{A}$ issues a polynomial number of adaptive queries of the form $(S \subseteq \{P_1, \ldots, P_N\}$, $C)$ for circuits $C : \{0,1\}^k \to \{0,1\}$ of depth at most d. For each query, the challenger $\mathcal{C}$ runs simulator $\{\mathsf{p}_i\}_{i \in S} \leftarrow \mathcal{S}_2(C, \{\mathsf{ct}_1, \ldots, \mathsf{ct}_k\}, C(m_1, \ldots, m_k), S, st)$ and sends $\{\mathsf{p}_i\}_{i \in S}$ to $\mathcal{A}$.
5. At the end of the experiment, $\mathcal{A}$ outputs a distinguishing bit b.

The proof of Theorem 3.1 proceeds a sequence of hybrid experiments, H_0, H_1, and H_2 between an adversary and a challenger.

- H_0: Real experiment
- H_1: Identical to H_0 except that for partial decryption phases.
  - If $i \in S^*$, then $\mathcal{C}$ returns $\mathsf{p}_i = \langle \hat{\mathsf{ct}}, \mathsf{sk}_i \rangle + (N!)^2 \cdot e_i \bmod q$ where $e_i \leftarrow [-B_{\mathsf{sm}}, B_{\mathsf{sm}}]$
  - If $i \notin S^*$, then $\mathcal{C}$ outputs p_i defined by

$$\lambda_{0,i}^{S^*} \cdot \lfloor \frac{q}{2} \rceil \cdot C(\boldsymbol{m}) + \sum_{j \in S^*} \lambda_{j,i}^{S^*} \cdot (\langle \hat{\mathsf{ct}}, \mathsf{sk}_j \rangle) + (N!)^2 \cdot e_i \bmod q$$

  where $e_i \leftarrow [-B_{\mathsf{sm}}, B_{\mathsf{sm}}]$ and $C(\boldsymbol{m}) = C(m_1, \ldots, m_k)$.

[7] For t-out-of-N access structure, a maximal invalid party is identical to a subset S^* of parties of size $t-1$.

– H_2: Ideal experiment

Flaw in the security proof of [10]. In [10], they (incorrectly) proved that H_1 and H_0 are indistinguishable. To be precise, on page 53, given a relation $\lfloor \frac{q}{2} \rceil \cdot C(\boldsymbol{m}) = \mathsf{FHE.Dec}_0(\mathsf{sk}, \hat{\mathsf{ct}}) + \widetilde{e}$, they showed that

$$\mathsf{p}_i = \lambda_{0,i}^{S^*} \cdot \lfloor \frac{q}{2} \rceil \cdot C(\boldsymbol{m}) + \sum_{j \in S^*} \lambda_{j,i}^{S^*} \cdot \mathsf{FHE.Dec}(\mathsf{sk}_j, \hat{\mathsf{ct}}) + (N!)^2 \cdot e,$$

which is identical to the following

$$\lambda_{0,i}^{S^*} \cdot \mathsf{FHE.Dec}_0(\mathsf{sk}, \hat{\mathsf{ct}}) + \widetilde{e} + \sum_{j \in S^*} \lambda_{j,i}^{S^*} \cdot \mathsf{FHE.Dec}(\mathsf{sk}_j, \hat{\mathsf{ct}}) + (N!)^2 \cdot e.$$

For correctness, the term $\widetilde{e}$ should be replaced by $\lambda_{0,i}^{S^*} \cdot \widetilde{e}$, not solely $\widetilde{e}$. The flaw messes up the proof. In fact, we can show that H_0 and H_1 can be distinguished.

3.3 Distinguishability between H_0 and H_1

We now demonstrate that H_0 and H_1 are distinguishable, even though **??** remains valid. For simplicity, we assume the following: Given a proper maximal invalid set S^*, a circuit C, and messages $m_i \in 0, 1$ such that $C(m_1, \ldots, m_k) = 0$, $\mathcal{C}$ can compute $\hat{\mathsf{ct}}$ as defined in Definition 3.2 Then, we first observe that

$$\langle \hat{\mathsf{ct}}, \mathsf{sk} \rangle \bmod q = \lfloor \frac{q}{2} \rceil \cdot C(\boldsymbol{m}) + (N!)^2 \cdot e = (N!)^2 \cdot e$$

for some $e \in [-B, B]$ by Definition 2.2. Moreover, $\mathcal{A}$ already possesses a set of secret shares $\{\mathsf{sk}_i\}_{i \in S^*}$. In other words, when $\mathcal{C}$ computes $\mathsf{p}_i = \langle \hat{\mathsf{ct}}, \mathsf{sk}_i \rangle \bmod q + (N!)^2 \cdot e_i$ for $i \in S^*$, $\mathcal{A}$ can recover each e_i by removing the term $\langle \hat{\mathsf{ct}}, \mathsf{sk}_i \rangle \bmod q$ for every $i \in S^*$. Thus, in the following, we consider e_i to be zero when $i \in S^*$.

Decryption in H_0). For each adaptive query $S \subset \{P_1 \ldots, P_N\}$, $\mathcal{C}$ provides set of partial decryptions

$$\mathsf{p}_i = \mathsf{TFHE.PartDec}(\mathsf{pk}, \hat{\mathsf{ct}}, \mathsf{sk}_i) = \langle \hat{\mathsf{ct}}, \mathsf{sk}_i \rangle + (N!)^2 \cdot e_i \bmod q,$$

when $j \notin S^*$. We here note $e_i \leftarrow [-B_{\mathsf{sm}}, B_{\mathsf{sm}}]$. Hence, $\mathcal{A}$ can run the final decryption phase as follows:

$$\lambda_{i,0}^{S^*} \cdot \mathsf{p}_i + \sum_{j \in S^*} \lambda_{j,0}^{S^*} \cdot (\langle \hat{\mathsf{ct}}, \mathsf{sk}_j \rangle) \bmod q$$

By definition of p_i, it is represented by

$$\lambda_{i,0}^{S^*} \cdot \big(\langle \hat{\mathsf{ct}}, \mathsf{sk}_i \rangle + (N!)^2 \cdot e_i \big) + \sum_{j \in S^*} \lambda_{j,0}^{S^*} \cdot (\langle \hat{\mathsf{ct}}, \mathsf{sk}_j \rangle) \bmod q.$$

Moreover, due to the linear property of inner products, we have

$$\langle \hat{\mathsf{ct}}, \lambda_{i,0}^{S^*} \cdot \mathsf{sk}_i + \sum_{j \in S^*} \lambda_{j,0}^{S^*} \cdot \mathsf{sk}_j \rangle + \lambda_{i,0}^{S^*} \cdot (N!)^2 \cdot e_i \bmod q,$$

which is identical to

$$\langle \hat{\mathsf{ct}}, \mathsf{sk} \rangle + \lambda_{i,0}^{S^*} \cdot (N!)^2 \cdot e_i \bmod q.$$

Hence, in the game H_0, an attacker finally gets

$$\lfloor \frac{q}{2} \rceil \cdot C(\boldsymbol{m}) + (N!)^2 \cdot e + \lambda_{i,0}^{S^*} \cdot (N!)^2 \cdot e_i = (N!)^2 \cdot e + \lambda_{i,0}^{S^*} \cdot (N!)^2 \cdot e_i \in \mathbb{Z}$$

if $B \cdot N! + (N!)^3 \cdot N \cdot B_{\mathsf{sm}} \leq q/4$.

Decryption in H_1). For each adaptive query $S \subset \{P_1 \ldots, P_N\}$, $\mathcal{C}$ provides set of partial decryptions

$$\widetilde{\mathsf{p}}_i = \lambda_{0,i}^{S^*} \cdot \lfloor \frac{q}{2} \rceil \cdot C(\boldsymbol{m}) + \sum_{j \in S^*} \lambda_{j,i}^{S^*} \cdot \langle \hat{\mathsf{ct}}, \mathsf{sk}_j \rangle + (N!)^2 \cdot \widetilde{e_i} \bmod q,$$

since $\mathcal{C}$ does not know sk_j when $j \notin S^*$. Here, it satisfies that $\widetilde{e_i} \leftarrow [-B_{\mathsf{sm}}, B_{\mathsf{sm}}]$.

Due to $\langle \hat{\mathsf{ct}}, \mathsf{sk} \rangle = \lfloor \frac{q}{2} \rceil \cdot C(\boldsymbol{m}) + (N!)^2 \cdot e$, $\widetilde{\mathsf{p}}_i$ can be represented by

$$\widetilde{\mathsf{p}}_i = \lambda_{0,i}^{S^*} \cdot \big(\langle \hat{\mathsf{ct}}, \mathsf{sk} \rangle - (N!)^2 \cdot e \big) + \sum_{j \in S^*} \lambda_{j,i}^{S^*} \cdot \langle \hat{\mathsf{ct}}, \mathsf{sk}_j \rangle + (N!)^2 \cdot \widetilde{e_i} \bmod q.$$

Moreover, the linearity of an inner product, it satisfies that

$$\widetilde{\mathsf{p}}_i = \langle \hat{\mathsf{ct}}, \lambda_{0,i}^{S^*} \cdot \mathsf{sk} + \sum_{j \in S^*} \lambda_{j,i}^{S^*} \mathsf{sk}_j \rangle - \lambda_{0,i}^{S^*} \cdot (N!)^2 \cdot e + (N!)^2 \cdot \widetilde{e_i} \bmod q.$$

On the other hand, the correctness Shamir secret sharing scheme implies $\lambda_{0,i}^{S^*} \cdot \mathsf{sk} + \sum_{j \in S^*} \lambda_{j,i}^{S^*} \mathsf{sk}_j = \mathsf{sk}_i$. Thus, $\widetilde{\mathsf{p}}_i$ can be expressed as

$$\langle \hat{\mathsf{ct}}, \mathsf{sk}_i \rangle - \lambda_{0,i}^{S^*} \cdot (N!)^2 \cdot e + (N!)^2 \cdot \widetilde{e_i} \bmod q.$$

On the other hand, $\mathcal{A}$ can compute the decryption of $\hat{\mathsf{ct}}$ as follows:

$$\lambda_{i,0}^{S^*} \cdot \widetilde{\mathsf{p}}_i + \sum_{j \in S^*} \lambda_{j,0}^{S^*} \cdot (\langle \hat{\mathsf{ct}}, \mathsf{sk}_j \rangle \bmod q).$$

By definition of $\widetilde{\mathsf{p}}_i$ and the linearity of inner products, the above equation is identical to

$$\langle \hat{\mathsf{ct}}, \underbrace{\lambda_{i,0}^{S^*} \cdot \mathsf{sk}_i + \sum_{j \in S^*} \lambda_{j,0}^{S^*} \cdot \mathsf{sk}_j}_{\mathsf{sk}} \rangle - \lambda_{i,0}^{S^*} \cdot \lambda_{0,i}^{S^*} \cdot (N!)^2 \cdot e + \lambda_{i,0}^{S^*} \cdot (N!)^2 \cdot e_i.$$

Moreover, by definition of the Lagrange's coefficients, it holds $\lambda_{i,0}^{S^*} \cdot \lambda_{0,i}^{S^*} = 1$. Therefore, $\mathcal{A}$ finally has

$$\langle \hat{\mathsf{ct}}, \mathsf{sk} \rangle - (N!)^2 \cdot e + \lambda_{i,0}^{S^*} \cdot (N!)^2 \cdot e_i.$$

By definition, it can be represented by

$$\lfloor \frac{q}{2} \rceil \cdot C(\boldsymbol{m}) + \lambda_{i,0}^{S^*} \cdot (N!)^2 \cdot e_i = \lambda_{i,0}^{S^*} \cdot (N!)^2 \cdot e_i,$$

which implies $\mathcal{A}$ gets $\lambda_{i,0}^{S^*} \cdot (N!)^2 \cdot e_i$ over $\mathbb{Z}$.

Distinguishing H_0 and H_1. Following the paragraphs, we can summarize that $\mathcal{A}$ can obtains the following information

- From H_0, $\mathcal{A}$ gets $E_0 = (N!)^2 \cdot e + \lambda_{i,0}^{S^*} \cdot (N!)^2 \cdot e_i$ over $\mathbb{Z}$
- From H_1, $\mathcal{A}$ gets $E_1 = \lambda_{i,0}^{S^*} \cdot (N!)^2 \cdot e_i$ over $\mathbb{Z}$.

Under the condition that $B/B_{\mathsf{sm}} = \mathsf{negl}(\lambda)$, the distributions are computationally indistinguishable. However, this setup introduces an algebraic relation. Specifically, $\mathcal{A}$ also knows that $\lambda_{i,0}^{S^*} = \frac{a}{b} \bmod q \in \mathbb{Z}_q$, where a and b are relatively prime integers. On the other hand, $\lambda_{i,0}^{S^*} \cdot (N!)^2$ must be an integer.

Assuming that $(N!)^2$ can be expressed as $b \cdot c$ for some integer $c \in \mathbb{Z}$, it follows that $\lambda_{i,0}^{S^*} \cdot (N!)^2 = a \cdot c$.

As a result, E_1 must be a multiple of $a \cdot c$, while E_0 only needs to be a multiple of c. For E_0 to also be a multiple of $a \cdot c$, the error term e must be divisible by a. The probability of this happening is less than $1/a \leq 1/2$.

Thus, by checking whether E_β is divisible by $a \cdot c$, $\mathcal{A}$ can easily determine whether $\beta = 0$ or $\beta = 1$. This distinction shows that H_0 and H_1 are distinguishable, contradicting the original proof.

4 Breaking TFHE from Shamir Secret Sharing

We now present a polynomial time algorithm for breaking TFHE in Section 3.1 for t-out-of-N threshold access structure under some constraints.

By definition of the simulation security (Definition 3.2), the adversary $\mathcal{A}$ outputs a maximal invalid party set S^*, and messages. Furthermore, the simulation security provides a set of partial decryptions $\{\mathsf{p}_i\}_{i \in S}$ to $\mathcal{A}$, where S is a set containing S^* of size t. We here denote the maximal invalid set S^* by $\{2, \ldots, t-1\}$, and $S = \{1\} \cup S^*$ for simplicity.

We now delve into the details of the final decryption phase. First, parties in S can compute the Lagrange coefficient $\lambda_{i,0}^{S}$ and

$$\begin{aligned} \sum_{i \in S} \lambda_{i,0}^{S} \cdot \mathsf{p}_i &= \sum_{i \in S} \lambda_{i,0}^{S} \cdot (\langle \mathsf{ct}, \mathsf{sk}_i \rangle + (N!)^2 \cdot e_i) \bmod q \\ &= \langle \mathsf{ct}, \mathsf{sk} \rangle + \sum_{i \in S} \lambda_{i,0}^{S} \cdot (N!)^2 \cdot e_i \bmod q \end{aligned}$$

$$= \lfloor \frac{q}{2} \rceil m + (N!)^2 \cdot e + \sum_{i \in S} \lambda_{i,0}^S \cdot (N!)^2 \cdot e_i \bmod q.$$

where $(N!)^2 \cdot e$ is an error in ciphertext ct of size less than $B \cdot (N!)^2$. Here, we note that $(N!)^2 \cdot e + \sum_{i \in S} \lambda_{i,0}^S \cdot (N!)^2 \cdot e_i$ is smaller than $q/4$ because of $B \cdot (N!)^2 + (N!)^3 \cdot N \cdot B_{\mathsf{sm}} \leq q/4$. Then, one can recover the message $m \in \{0, 1\}$ by checking the size of $\left[\sum_{i \in S} \lambda_{i,0}^S \cdot \mathsf{p}_i\right]_q$.

We now intend to describe an algorithm for recovering the e from the intermediate term $\left[\sum_{i \in S} \lambda_{i,0}^S \cdot \mathsf{p}_i\right]_q$. Since the decryption algorithm outputs m, $\mathcal{A}$ can remove $\lfloor \frac{q}{2} \rceil \cdot m$.

Here, we also remark that the adversary $\mathcal{A}$ effortlessly compute $\lambda_{i,0}^S$ for all i. Moreover, $\mathcal{A}$ has access to e_i with $i \in S$. This collective knowledge, shared among the colluding parties, gives a relation

$$\sum_{i \in S} \lambda_{i,0}^S \cdot \mathsf{p}_i - \sum_{i=2}^{t-1} \lambda_{i,0}^S \cdot (N!)^2 \cdot e_i \bmod q = (N!)^2 \cdot e + \lambda_{1,0}^S \cdot (N!)^2 \cdot e_1.$$

From the above size constraint $B \cdot (N!)^2 + (N!)^3 \cdot N \cdot B_{\mathsf{sm}} \leq q/4$, the last equation is defined over $\mathbb{Z}$, not $\mathbb{Z}_q$. For a simple description, we let E denote the term:

$$E = (N!)^2 \cdot e + \lambda_{1,0}^S \cdot (N!)^2 \cdot e_1 \in \mathbb{Z}. \tag{1}$$

In conclusion, the adversary $\mathcal{A}$ obtains E leveraging the final decryption phase. As the next step, we describe an algorithm for recovering e, noise of ciphertext of TFHE from E.

Recover e from E. Let $g = \gcd((N!)^2, \lambda_{1,0}^S \cdot (N!)^2)$ and consider the following relation:

$$\begin{aligned} E/g &= (N!)^2/g \cdot e + (\lambda_{1,0}^S \cdot (N!)^2)/g \cdot e_1 \\ &= (N!)^2/g \cdot e \bmod (\lambda_{1,0}^S \cdot (N!)^2)/g. \end{aligned}$$

Therefore, if $|e| \leq \frac{\lambda_{1,0}^S \cdot (N!)^2}{2g}$, then e can be exactly determined, as $C, (N!)^2, g, \lambda_{1,0}^S$ are already known to $\mathcal{A}$.

Once the attacker gets e from ct, then the attacker obtains $\mathsf{ct} - (N!)^2 \cdot e = \langle \mathbf{a}, \mathsf{sk} \rangle$ for some known $\mathbf{a} \in \mathbb{Z}_q^n$. Hence, for sufficiently many ciphertexts ct_i, by repeating the same process, we can obtain a family of inner products $\langle \mathbf{a}_i, \mathsf{sk} \rangle$ for some $\mathbf{a}_i$. After that, by the linear algebra, we can recover sk using the Gaussian elimination, which takes polynomial time in n.

4.1 Attack amplification

In the following, we state how to mitigate the constraint for breaking the TFHE by leveraging several partial decryptions.

Let $\mathcal{H}$ be a set of honest participants of size h and $S_k = S^* \bigsqcup \{k\}$ where $k \in \mathcal{H}$. For simplicity, we assume that $\mathcal{H} = \{1, 2, \ldots, h\}$ and S^* is a subset of $\{h+1, \ldots, N\}$ of size $t-1$. Let ct be a ciphertext of a message $m = 0$.

Then, since $|S_k| = t$, by definition of TFHE, S_k can conduct the final decryption algorithm. As the previous section, each S_k can compute the following:

$$\begin{aligned}\sum_{i\in S_k} \lambda_{i,0}^{S_k} \cdot \mathsf{p}_i &= \sum_{i\in S_k} \lambda_{i,0}^{S_k} \cdot (\langle \mathsf{ct}, \mathsf{sk}_i\rangle + (N!)^2 \cdot e_i) \bmod q \\ &= \langle \mathsf{ct}, \mathsf{sk}\rangle + \sum_{i\in S_k} \lambda_{i,0}^{S_k} \cdot (N!)^2 \cdot e_i \bmod q \\ &= \lfloor \frac{q}{2} \rceil m + (N!)^2 \cdot e + \sum_{i\in S_k} \lambda_{i,0}^{S_k} \cdot (N!)^2 \cdot e_i \bmod q \\ &= (N!)^2 \cdot e + \sum_{i\in S_k} \lambda_{i,0}^{S_k} \cdot (N!)^2 \cdot e_i\end{aligned}$$

where $(N!)^2 \cdot e$ is an error in FHE ciphertext ct and $\lambda_{i,0}^{S_k}$ is the Lagrange coefficient corresponding to the set S_k. By reducing the term $\{e_i\}_{i\in S^*}$ shared noise terms from the adversary, $\mathcal{A}$ can obtain relations

$$E_k = (N!)^2 \cdot e + \lambda_{k,0}^{S_k} \cdot (N!)^2 \cdot e_k \quad \text{for} \quad k \in \mathcal{H}.$$

From the same argument as the above, each E_k is lying over $\mathbb{Z}$. We then consider a set of equations of E_k/g_k, where $g_k = \gcd((N!)^2, \lambda_{k,0}^{S_k} \cdot (N!)^2)$ for $k \in \mathcal{H}$:

$$\left\{ E_k/g_k = (N!)^2/g_k \cdot e + (\lambda_{k,0}^{S_k} \cdot (N!)^2)/g_k \cdot e_k \right\}_{k\in\{1,\ldots,h\}}. \tag{2}$$

To represent these relations as a matrix multiplication, we let $\mathbf{e} = (e, e_1, \ldots, e_h)$, $\mathbf{c} = (E_1/g_1, \ldots, E_h/g_h)$, and define $\mathbf{A}$ as following:

$$\mathbf{A} = \begin{pmatrix} \frac{(N!)^2}{g_1} & -\lambda_{1,0}^{S_1} \cdot \frac{(N!)^2}{g_1} & & \\ \frac{(N!)^2}{g_2} & & -\lambda_{2,0}^{S_2} \cdot \frac{(N!)^2}{g_2} & \\ \vdots & & & \ddots \\ \frac{(N!)^2}{g_h} & & & -\lambda_{h,0}^{S_h} \cdot \frac{(N!)^2}{g_h} \end{pmatrix}.$$

Under the notation, $\mathbf{A} \cdot \mathbf{e} = \mathbf{c}$ over $\mathbb{Z}$.

Once $\mathbf{e}$ is obtained from the linear system, as in the previous attack, one can recover the secret key in polynomial time in n.

Recover e from $(\mathbf{A}, \mathbf{c})$. The remaining part is to find $\mathbf{e}$ from a pair $(\mathbf{A}, \mathbf{c})$ satisfying $\mathbf{A} \cdot \mathbf{e} = \mathbf{c}$. Since the rank of $\mathbf{A}$ is obviously h, the orthogonal complement $\mathbf{A}^\perp$ can be generated by a single vector, denoted by $\mathbf{a}$. We also let $\mathbf{e}_0$ be a particular solution such that $\mathbf{A} \cdot \mathbf{e}_0 = \mathbf{c}$. Then, $\mathbf{e}$ can be expressed as $\mathbf{e}_0 + \mathbf{a} \cdot y$ for some integer y.

We first define a diagonal matrix $\mathbf{H} \in \mathbb{Z}^{(h+1)\times(h+1)}$ as $\mathsf{diag}\left(\frac{B_{\mathsf{sm}}}{B}, 1, \ldots, 1\right)$ to balance the entries of $\mathbf{e}$. We then intend to find a vector $\mathbf{H} \cdot \mathbf{e}$ from the set $\mathbf{H} \cdot \mathbf{e}_0 + \mathbf{H} \cdot \mathbf{a} \cdot \mathbb{Z}$.

To this end, we will show that $\mathbf{H} \cdot \mathbf{e}$ is the shortest vector of $\mathbf{H} \cdot \mathbf{e}_0 + \mathbf{H} \cdot \mathbf{a} \cdot \mathbb{Z}$ when $\sqrt{h+1} \cdot B_{\mathsf{sm}} \leq \|\mathbf{H} \cdot \mathbf{a}\|$. This is because $\|\mathbf{H} \cdot \mathbf{e}\| \leq \sqrt{h+1} \cdot B_{\mathsf{sm}}$. With

an obvious computation, one can show that the size of a vector of the form $\mathbf{H} \cdot \mathbf{e}_0 + \mathbf{H} \cdot \mathbf{a} \cdot y$ becomes smallest when $y = -\left\lfloor \frac{\langle \mathbf{H}\cdot\mathbf{e}_0, \mathbf{H}\cdot\mathbf{a} \rangle}{\langle \mathbf{H}\cdot\mathbf{a}, \mathbf{H}\cdot\mathbf{a} \rangle} \right\rceil$. Thus, one can recover the target vector $\mathbf{H} \cdot \mathbf{e}$ (Resp. $\mathbf{e}$) in polynomial time under the constraint

$$\sqrt{h+1} \cdot B_{\mathsf{sm}} \leq \|\mathbf{H} \cdot \mathbf{a}\|. \tag{3}$$

Therefore, it suffices to describe the attack constraint of Eq. (3) with respect to $\mathcal{M}$. To estimate a lower bound of $\|\mathbf{H} \cdot \mathbf{a}\|$, we initially consider a vector

$$\mathbf{t} = \left(L', \frac{L'}{\lambda_{1,0}^{S_1}}, \frac{L'}{\lambda_{2,0}^{S_2}}, 0 \ldots, 0 \right),$$

given the least common multiple L' of $(\lambda_{1,0}^{S_1} \cdot (N!)^2)/g_1$ and $(\lambda_{2,0}^{S_2} \cdot (N!)^2)/g_2$. We note that the vector $((\lambda_{1,0}^{S_1} \cdot (N!)^2)/g_1, (N!)^2/g_1, \ldots, 0)$ is orthogonal to the first row of $\mathbf{A}$, and the vector $((\lambda_{2,0}^{S_2} \cdot (N!)^2)/g_2, 0, (N!)^2/g_2, \ldots, 0)$ is orthogonal to the second row of $\mathbf{A}$, respectively. It implies that $\mathbf{t}$ is the smallest vector orthogonal to the first and second row of $\mathbf{A}$ at the same time.

With an obvious extension, one can verify that the vector $\mathbf{a}$ is of the form:

$$\mathbf{a} = \left(L, \frac{L}{\lambda_{1,0}^{S_1}}, \frac{L}{\lambda_{2,0}^{S_2}}, \ldots, \frac{L}{\lambda_{h,0}^{S_h}} \right),$$

where L is the least common multiple of $\{(\lambda_{i,0}^{S_i} \cdot (N!)^2)/g_i\}_{1 \leq i \leq h}$. This ensures that $\|\mathbf{H} \cdot \mathbf{a}\| \geq \frac{B_{\mathsf{sm}}}{B} \cdot L$. Combining it with the above attack constraint, one can recover the noise vector $\mathbf{e}$ when $L \geq \sqrt{h+1} \cdot B$.

4.2 Heuristic Analysis

While both attacks that we propose in the previous section are theoretically feasible, these constraints heavily depend on which maximal invalid set is selected. In this section, we approximate the average case of constraints through several experimental results for various t and N to provide a general result that does not depend on the maximal invalid set. From our implementation, we give approximation results for breaking TFHE construction [10, Sec. 5.3].

Theorem 4.1 (Heuristic) *Suppose FHE is a fully homomorhpic scheme that satisfies a special decryption (Definition 2.2) with a multiplicative constant $(N!)^2$ and a noise bound parameter B, and SS is a Shamir secret sharing scheme.*

Then the t-out-of-N TFHE scheme based on $(\mathsf{FHE}, \mathsf{SS})$ from construction [10, Sec. 5.3] can be broken with at least $1/2$ probability in polynomial time t and $\log N$ for $t-1$ adversary parties if one of the following holds:

- $B \leq N^{\frac{7}{10}\frac{t}{\log t}}$
- $\sqrt{t+1} \cdot B \leq N^{\frac{2}{5}t}$ *and* $2t \leq N$.

Proof. The proof of this theorem is in the form of applying the algorithms in Section 4 to several parameters and averaging the obtained results.

To be precise, we first identify two attacks to classify the attacks:

- Basic attack: a constraint from a single equation: $B \leq (\lambda_{i,0}^S \cdot (N!)^2)/g$ where $g = \gcd((N!)^2, \lambda_{i,0}^S \cdot (N!)^2)$ with a maximal invalid party S^* and $S = S^* \bigsqcup \{i\}$.[8]
- Improved attack: a constraint from h equations: $\sqrt{h+1} \cdot B \leq L$ where $L = \operatorname*{lcm}_{1 \leq k \leq h} \left(\frac{\lambda_{k,0}^{S_k} \cdot (N!)^2}{g_k} \right)$ with $g_k = \gcd(((N!)^2, \lambda_{k,0}^{S_k} \cdot (N!)^2))$, a maximal invalid party S^* and $S_k = S^* \bigsqcup \{k\}$.

Empirically, for various parameters t and N, we aim to show that

$$\mathbb{E}\left((\lambda_{i,0}^S \cdot (N!)^2)\right)/g \geq N^{\frac{7}{10}\frac{t}{\log t}} \text{ and } \mathbb{E}(L) \geq N^{\frac{2}{5}t},$$

where $\mathbb{E}(\cdot)$ is an average function. Combining them together, completes the proof. We then justify each case in the following.

Basic attack. Initially, we claim the size of $(\lambda_{i,0}^S \cdot (N!)^2)/g$ where $S = S^* \bigsqcup \{i\}$. By the structure of $\lambda_{i,0}^S \cdot (N!)^2$, we observe that

$$\lambda_{i,0}^S \cdot (N!)^2 = \prod_{x \in S \setminus \{i\}} \frac{-x}{(i-x)} \cdot (N!)^2 = \left(\prod_{x \in S^*} -x \right) \cdot \prod_{x \in S^*} \frac{(N!)^2}{(i-x)}$$
$$(N!)^2 = \left(\prod_{x \in S^*} i - x \right) \cdot \prod_{x \in S^*} \frac{(N!)^2}{(i-x)}.$$

From this relation, it follows that g is divisible by $\prod_{x \in S^*} \frac{(N!)^2}{(1-x)}$ and that $(\lambda_{i,0}^S \cdot (N!)^2)/g$ is divisible by $\prod_{x \in S^*} -x$. To specify further,

$$(\lambda_{i,0}^S \cdot (N!)^2)/g = \frac{\prod_{x \in S^*} -x}{\gcd(\prod_{x \in S^*} i - x, \prod_{x \in S^*} -x)}.$$

Consequently, the expected size of $(\lambda_{i,0}^S \cdot (N!)^2)/g$ should be less than N^t. Through this observation, we initially compared the actual values with $N^{\frac{t}{\log t}}$ and $N^{\frac{t}{\log t \log\log t}}$.

In Table 1, the 'Real' column gives the mean value of $\log_N((\lambda_{i,0}^S \cdot (N!)^2)/g)$. The 'ratio with $\frac{t}{\log t}$' column gives a ratio $\log_N((\lambda_{i,0}^S \cdot (N!)^2)/g)/\frac{t}{\log t}$. This result shows that the size of $\log_N((\lambda_{i,0}^S \cdot (N!)^2)/g)$ is always larger than $\frac{t}{\log t \log\log t}$. At the same time, $\frac{7}{10}\frac{t}{\log t}$ gives the tight bound for all parameters. Especially, we have observed that the ratio with $\frac{t}{\log t}$ converges to (approximately) 0.7 regardless of the size of N. Hence, we empirically argue that $\log_N((\lambda_{i,0}^S \cdot (N!)^2)/g) \geq \frac{7}{10} \cdot \frac{t}{\log t}$.

[8] In the previous section, we simply employ $S = S^* \bigsqcup \{1\}$.

Improved attack. In this case, we focus on determining the size of L. L is defined as the least common multiple (lcm) of the set $(\lambda_{i,0}^{S_k} \cdot (N!)^2)/g_k$ for $1 \leq k \leq h$. Drawing parallels with the basic case, we note that:

$$\lambda_{j,0}^{S_j} \cdot (N!)^2 = \prod_{x \in S_j \setminus \{j\}} \frac{-x}{(j-x)} \cdot (N!)^2 = \prod_{x \in S^*} \frac{-x}{(j-x)} \cdot (N!)^2$$
$$= \left(\prod_{x \in S^*} \frac{1}{(j-x)} \cdot (N!)^2 \right) \cdot \left(\prod_{x \in S^*} -x \right).$$

Given that g_i incorporates the factor $\left(\prod_{x \in S^*} \frac{1}{(j-x)} \cdot (N!)^2\right)$, it follows that $L = \operatorname{lcm}_{1 \leq k \leq h}((\lambda_{k,0}^{S_k} \cdot (N!)^2)/g_k)$ invariably divides $\left(\prod_{x \in S^*} -x\right)$.

This scenario mirrors the basic case, where each term $(\lambda_{k,0}^{S_k} \cdot (N!)^2)/g_k$ in L is analogous. Based on this framework, we logically infer that the size of L would closely approximate $\left(\prod_{x \in S^*} -x\right)$, especially for a significant value of h. Following this logic, we compared the actual value against N^t.

By taking $h = t+1$, the results, as displayed in the Table 2, consistently show that the expected size of $\log_N(L)$ tends to converge to $0.86 \cdot t/2 \geq \frac{2}{5}t$ independent to the choice of N. Hence, we heuristically argue $\log_N(L) \geq \frac{2}{5} \cdot t$. □

5 Correcting Proofs and Considerations in Practical Implementation of TFHE

In this section, we provide a modification of the TFHE scheme. However, we also demonstrate that while these corrections are theoretically sound, they still pose problems when instantiated through current homomorphic encryption libraries. Specifically, we illustrate that careful use of homomorphic encryption libraries is essential for the practical implementation of TFHE.

Correction of TFHE. The attacks in Section 4 can be easily fixed by blowing up the parameters from $(N!)^2$ to $(N!)^4$. To be precise, a ciphertxt $\mathsf{ct} \leftarrow \mathsf{TFHE.Enc}(\mathsf{pk}, m)$ holds the following: $\langle \mathsf{ct}, \mathsf{sk} \rangle = \lfloor \frac{q}{2} \rceil \cdot m + e \cdot (N!)^4$. This modification directly implies that $\mathcal{A}$ obtains the following structure

$$\lfloor \frac{q}{2} \rceil \cdot C(m_1, \ldots, m_k) + e \cdot (N!)^4 + \lambda_{1,0} \cdot (N!)^2 \cdot e_1$$

during the decryption phase when $\mathcal{A}$ submits a query $S = S^* \bigsqcup \{1\}$. This modification is obtained by exploiting a multiplicative constant $(N!)^4$. More precisely, it satisfies that $(N!)^4 \cdot e + \lambda_{1,0} \cdot (N!)^2 \cdot e_1$ can be expressed as $\lambda_{1,0} N!^2 \cdot (e \cdot \kappa + e_1)$ for some κ since $(N!)^4$ is a multiple of $\lambda_{1,0} \cdot (N!)^2$. If e_1 is super-polynomially bigger than $e \cdot \kappa$, then $e \cdot \kappa + e_1$ is statistically close to e_1 and hence this can be simulated by the simulator, which prevents our attacks and ensures the Theorem 3.1 holds.

Parameters		Basic attack			
N	t	$\frac{t}{\log t}$	$\frac{t}{\log t \log\log t}$	Real	ratio with $\frac{t}{\log t}$
50	10	3.01	1.74	3.33	1.11
	20	4.63	2.19	5.18	1.12
	30	6.11	2.66	6.30	1.03
	40	7.52	3.12	6.62	0.88
	49	8.73	3.51	6.07	0.70
100	20	4.63	2.19	6.33	1.37
	40	7.52	3.12	9.57	1.27
	50	8.86	3.55	10.62	1.20
	60	10.16	3.96	11.13	1.10
	80	12.65	4.76	11.68	0.92
	99	14.93	5.47	10.51	0.70
150	30	6.11	2.66	9.29	1.52
	60	10.16	3.96	13.99	1.38
	75	12.04	4.56	15.36	1.28
	90	13.86	5.14	16.38	1.18
	120	17.37	6.23	16.78	0.97
	149	20.64	7.24	14.85	0.72

Table 1: Comparison basic attack results of Expected values and Real value. Basic attack consists of $t-1$ adversaries and 1 honest party. For each parameter (N, t), 1000 experiments are conducted.

Parameters			Improved attack			
N	t	h	$\frac{t}{2}$	$\frac{t}{\log t}$	Real	ratio with $\frac{t}{2}$
50	10	11	5	3.01	5.31	1.06
	20	21	10	4.63	9.25	0.93
	25	26	12.5	5.38	10.77	0.86
100	20	21	10	4.63	10.92	1.09
	40	41	20	7.52	18.71	0.94
	50	51	25	8.86	21.53	0.86
150	30	31	15	6.11	16.30	1.09
	60	61	30	10.16	28	0.93
	75	76	37.5	12.04	32.22	0.86

Table 2: Comparison improved attack results of Expected values and Real value. Basic attack consists of $t-1$ adversaries and h honest parties. For each parameter (N, t, h), 100 experiments are conducted.

This modification can be generalized as follows: Suppose that during the decryption phase, $\mathcal{A}$ can obtain the following

$$\lfloor \frac{q}{2} \rceil \cdot C(m_1, \ldots, m_k) + e \cdot (N!)^{\alpha} + \lambda_{1,0} \cdot (N!)^{\beta} \cdot e_1$$

where $\alpha \geq \beta + 2$. Then, one can prove the simulation security using the technique in the original paper [10].

Practical aspects of TFHE instantiation. However, such a fix of TFHE cannot achieve the simulation security when TFHE is instantiated by the state-of-the-art FHE libraries[1–3, 6, 44] satisfying special decryption (Definition 2.2).

To support our claim, we propose a new attack on the modified TFHE scheme under the assumption that $\mathsf{TFHE.Eval}(\mathsf{pk}, +, \mathsf{ct}_1, \mathsf{ct}_2)$ is conducted by $\mathsf{ct}_1 + \mathsf{ct}_2$, where $+$ is the addition circuit of two inputs. Remark that every $\mathsf{Special\ FHE}$ (Definition 2.2) obtained from state-of-the-art FHE scheme and its libraries implement a homomorphic evaluation for the addition circuit as the ciphertext addition, as in our assumption. We now consider the circuit $C : \{0,1\}^k \to \{0,1\}$ of depth at most d computed as follows:

1. Given the input $(m_1, \ldots, m_k)$, compute $C'(m_1, \ldots, m_k) = m_1^2 - m_1$ and
2. K times summation of C'. i.e., $C(m_1, \ldots, m_k) = \sum_{i=1}^{K}(m_1^2 - m_1)$, where $K = 1/(N!)^{\alpha} \bmod q$.

We first note that the circuit C consumes only 1 depth since it only adds messages K times after only one multiplication to get a ciphertext of message $m_1^2 - m_1$.

Suppose that the adversary $\mathcal{A}$ queries to the challenger $\mathcal{C}$ to obtain a ciphertext $\hat{\mathsf{ct}} \leftarrow \mathsf{TFHE.Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$ given a family ciphertexts $\{\mathsf{ct}_i\}$. By definition of the circuit C, $\hat{\mathsf{ct}}$ is represented by

$$\hat{\mathsf{ct}} = \sum_{i=1}^{K} \overline{\mathsf{ct}}$$

for some ciphertext $\overline{\mathsf{ct}} \leftarrow \mathsf{TFHE.Eval}(\mathsf{pk}, C', \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$ of which message is $m_1^2 - m_1 = 0$. On the other hand, the special decryption (Definition 2.2) implies that for the secret key sk, it satisfies that

$$\langle \overline{\mathsf{ct}}, \mathsf{sk} \rangle = (N!)^{\alpha} \cdot \overline{e}$$

for some $\overline{e}$. Thus, it gives a relation that

$$\langle \hat{\mathsf{ct}}, \mathsf{sk} \rangle = \sum_{i=1}^{K} \langle \overline{\mathsf{ct}}, \mathsf{sk} \rangle = \overline{e} \bmod q \tag{4}$$

We here note that $\mathcal{A}$ can query a circuit C to $\mathcal{C}$ since Definition 3.2 indicates that $\mathcal{A}$ can issue a circuit C of depth at most d. According to the basic attack with $\hat{\mathsf{ct}}$, $\mathcal{A}$ obtains X defined by

$$X = \overline{e} + \lambda_{1,0} \cdot (N!)^{\beta} \cdot e_1 \bmod q$$

$$= \overline{e} + \lambda_{1,0} \cdot (N!)^{\beta} \cdot e_1$$

If $\overline{e}$ is smaller than $\frac{\lambda_{1,0} \cdot (N!)^{\beta}}{2}$, then $\mathcal{A}$ gets $\overline{e} \bmod \lambda_{1,0} \cdot (N!)^{\beta} = \overline{e}$. In fact, the current implementations of FHE libraries, $\overline{e}$ is sufficiently small. After having $\overline{e}$, then one can easily recover sk as in Section 4.

The attack argues that the state-of-the-art FHE schemes cannot satisfy the special decryption (Definition 2.2) for every circuit C.

Remark. To the best of our knowledge, the simplest countermeasure to this attack is to implement bootstrapping/modulus switching algorithms for each addition. However, it significantly reduces the performance of FHE schemes and TFHE schemes for almost all applications.

References

1. HEAAN. `https://github.com/snucrypto/HEAAN`. Accessed: 2022-10-07.
2. HElib. `https://github.com/homenc/HElib`. Accessed: 2024-09-03.
3. Lattigo. `https://github.com/tuneinsight/lattigo`. Accessed: 2022-10-07.
4. S. Agrawal, D. Stehlé, and A. Yadav. Round-optimal lattice-based threshold signatures, revisited. In M. Bojanczyk, E. Merelli, and D. P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 8:1–8:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
5. G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold fhe. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 483–501. Springer, 2012.
6. A. A. Badawi, A. Alexandru, J. Bates, F. Bergamaschi, D. B. Cousins, S. Erabelli, N. Genise, S. Halevi, H. Hunt, A. Kim, Y. Lee, Z. Liu, D. Micciancio, C. Pascoe, Y. Polyakov, I. Quah, S. R.V., K. Rohloff, J. Saylor, D. Suponitsky, M. Triplett, V. Vaikuntanathan, and V. Zucca. OpenFHE: Open-source fully homomorphic encryption library. Cryptology ePrint Archive, Paper 2022/915, 2022. `https://eprint.iacr.org/2022/915`.
7. R. Bendlin and I. Damgård. Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In *Theory of Cryptography Conference*, pages 201–218. Springer, 2010.
8. R. Bendlin, S. Krehbiel, and C. Peikert. How to share a lattice trapdoor: threshold protocols for signatures and (h) ibe. In *International Conference on Applied Cryptography and Network Security*, pages 218–236. Springer, 2013.
9. D. Boneh, R. Gennaro, and S. Goldfeder. Using level-1 homomorphic encryption to improve threshold dsa signatures for bitcoin wallet security. In *International Conference on Cryptology and Information Security in Latin America*, pages 352–377. Springer, 2017.
10. D. Boneh, R. Gennaro, S. Goldfeder, A. Jain, S. Kim, P. M. Rasmussen, and A. Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. In *Annual International Cryptology Conference*, pages 565–596. Springer, 2018.
11. K. Boudgoust and P. Scholl. Simple threshold (fully homomorphic) encryption from lwe with polynomial modulus. *Cryptology ePrint Archive*, 2023.

12. Z. Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *Advances in Cryptology – CRYPTO 2012*, pages 868–886, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
13. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.
14. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on computing*, 43(2):831–871, 2014.
15. R. Canetti, R. Gennaro, S. Goldfeder, N. Makriyannis, and U. Peled. Uc non-interactive, proactive, threshold ecdsa with identifiable aborts. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1769–1787, 2020.
16. G. Castagnos, D. Catalano, F. Laguillaumie, F. Savasta, and I. Tucker. Bandwidth-efficient threshold ec-dsa. In *IACR International Conference on Public-Key Cryptography*, pages 266–296. Springer, 2020.
17. M. Checri, R. Sirdey, A. Boudguiga, J.-P. Bultel, and A. Choffrut. On the practical cpad security of "exact" and threshold fhe schemes and libraries. Cryptology ePrint Archive, Paper 2024/116, 2024. `https://eprint.iacr.org/2024/116`.
18. J. H. Cheon, W. Cho, and J. Kim. Improved universal thresholdizer from iterative shamir secret sharing. Cryptology ePrint Archive, Paper 2023/545, 2023. `https://eprint.iacr.org/2023/545`.
19. J. H. Cheon, H. Choe, A. Passelègue, D. Stehlé, and E. Suvanto. Attacks against the indcpa-d security of exact fhe schemes. Cryptology ePrint Archive, Paper 2024/127, 2024. `https://eprint.iacr.org/2024/127`.
20. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachene. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *international conference on the theory and application of cryptology and information security*, pages 3–33. Springer, 2016.
21. R. Cramer, I. Damgård, and Y. Ishai. Share conversion, pseudorandom secret-sharing and applications to secure computation. In *Theory of Cryptography: Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005. Proceedings 2*, pages 342–362. Springer, 2005.
22. E. C. Crites, C. Komlo, and M. Maller. Fully adaptive schnorr threshold signatures. In H. Handschuh and A. Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part I*, volume 14081 of *Lecture Notes in Computer Science*, pages 678–709. Springer, 2023.
23. M. Dahl, D. Demmler, S. El Kazdadi, A. Meyre, J.-B. Orfila, D. Rotaru, N. P. Smart, S. Tap, and M. Walter. Noah's ark: Efficient threshold-fhe using noise flooding. In *Proceedings of the 11th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, WAHC '23, page 35–46, New York, NY, USA, 2023. Association for Computing Machinery.
24. A. Dalskov, C. Orlandi, M. Keller, K. Shrishak, and H. Shulman. Securing dnssec keys via threshold ecdsa from generic mpc. In *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part II 25*, pages 654–673. Springer, 2020.
25. I. Damgård, T. P. Jakobsen, J. B. Nielsen, J. I. Pagter, and M. B. Østergaard. Fast threshold ecdsa with honest majority. *Journal of Computer Security*, 30(1):167–196, 2022.

26. I. Damgård and M. Koprowski. Practical threshold rsa signatures without a trusted dealer. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 152–165. Springer, 2001.
27. I. Damgård, C. Orlandi, A. Takahashi, and M. Tibouchi. Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices. *Journal of Cryptology*, 35(2):14, 2022.
28. A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung. How to share a function securely. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 522–533. ACM, 1994.
29. R. del Pino, S. Katsumata, M. Maller, F. Mouhartem, T. Prest, and M.-J. Saarinen. Threshold raccoon: Practical threshold signatures from standard lattice assumptions. To appear at Eurocrypt 2024.
30. Y. Desmedt and Y. Frankel. Threshold cryptosystesns. *Advance in Cryptology*, pages 305–315, 1989.
31. J. Doerner, Y. Kondi, E. Lee, and A. Shelat. Threshold ecdsa from ecdsa assumptions: The multiparty case. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1051–1066. IEEE, 2019.
32. J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Paper 2012/144, 2012.
33. Y. Frankel. A practical protocol for large group oriented networks. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 56–61. Springer, 1989.
34. R. Gennaro and S. Goldfeder. Fast multiparty threshold ecdsa with fast trustless setup. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1179–1194, 2018.
35. R. Gennaro, S. Goldfeder, and A. Narayanan. Threshold-optimal dsa/ecdsa signatures and an application to bitcoin wallet security. In *International Conference on Applied Cryptography and Network Security*, pages 156–174. Springer, 2016.
36. R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust threshold dss signatures. *Information and Computation*, 164(1):54–84, 2001.
37. R. Gennaro, T. Rabin, S. Jarecki, and H. Krawczyk. Robust and efficient sharing of rsa functions. *Journal of Cryptology*, 13(2):273–300, 2007.
38. C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 75–92. Springer, 2013.
39. K. D. Gur, J. Katz, and T. Silde. Two-round threshold lattice signatures from threshold homomorphic encryption. *Cryptology ePrint Archive*, 2023.
40. C. Komlo and I. Goldberg. Frost: flexible round-optimized schnorr threshold signatures. In *Selected Areas in Cryptography: 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers 27*, pages 34–65. Springer, 2021.
41. B. Li and D. Micciancio. On the security of homomorphic encryption on approximate numbers. In *Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I 40*, pages 648–677. Springer, 2021.
42. Y. Lindell and A. Nof. Fast secure multiparty ecdsa with practical distributed key generation and applications to cryptocurrency custody. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1837–1854, 2018.

43. D. Micciancio and A. Suhl. Simulation-secure threshold pke from lwe with polynomial modulus. Cryptology ePrint Archive, Paper 2023/1728, 2023. `https://eprint.iacr.org/2023/1728`.
44. Microsoft SEAL (release 4.1). `https://github.com/Microsoft/SEAL`, Jan. 2023. Microsoft Research, Redmond, WA.
45. A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.
46. V. Shoup. Practical threshold signatures. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 207–220. Springer, 2000.
47. D. R. Stinson and R. Strobl. Provably secure distributed schnorr signatures and a (t, n) threshold scheme for implicit certificates. In *Australasian Conference on Information Security and Privacy*, pages 417–434. Springer, 2001.