Scilight

Pragmatic Cybersecurity
https://www.sciltp.com/journals/pc

Article

Check for updates

Cryptanalysis of an Unbounded Fully Homomorphic Encryption Scheme †

Hyewon Sung [1,*,‡] and Mehdi Tibouchi [2,*,‡]

[1] Department of Mathematics, Ewha Womans University, Seoul 03760 , Korea
[2] NTT Social Informatics Laboratories, Tokyo 180-8585, Japan
* Correspondence: hyewonsung@ewha.ac.kr (H.S.); mehdi.tibouchi@ntt.com (M.T.)
† Extended from a non-archival presentation titled: Cryptanalysis of an Unbounded Fully Homomorphic Encryption Scheme. Presented at the 31st Australasian Conference on Information Security and Privacy, Perth, WA, Australia, 6–9 July 2026.
‡ These authors contributed equally to this work.

How To Cite: Sung, H.; Tibouchi, M. Cryptanalysis of an Unbounded Fully Homomorphic Encryption Scheme. *Pragmatic Cybersecurity* **2026**, *1*(2), 17. https://doi.org/10.53941/pc.2026.100017

Received: 9 July 2026
Revised: 13 August 2026
Accepted: 14 August 2026
Published: 9 September 2026

Abstract: Gentry's original blueprint for the construction of fully homomorphic encryption (FHE) starts from a *somewhat* homomorphic encryption scheme, that supports the homomorphic evaluation of arithmetic circuits of moderate multiplicative depth up to L, and whose decryption circuit itself can furthermore be "squashed" to a multiplicative depth less than L. Then, the homomorphic evaluation of this squashed decryption circuit, called bootstrapping, transforms any ciphertext into a ciphertext supporting at least one extra level of multiplication, enabling the homomorphic evaluation of arbitrary arithmetic circuits. *Bootstrapping* is the cornerstone of Gentry's breakthrough. Unfortunately, it introduces substantial overhead that limits the efficiency of most FHE schemes so far. Consequently, numerous attempts have been made to circumvent the bottleneck of bootstrapping by directly constructing encryption schemes that support the evaluation of circuits of unbounded depth, e.g., by avoiding *ciphertext noise*, the randomness present in Gentry-like somewhat homomorphic schemes that grows after each homomorphic multiplication. Noise effectively limits the depths of supported circuits, as ciphertexts fail to correctly decrypt when noise size exceeds a certain threshold. Most attempts to achieve noiseless FHE schemes have proved insecure, however. In this paper, we take a closer look at a recent construction along those lines due to Zheng et al.'s unbounded FHE scheme, and show that it is no exception. We show that it is susceptible to an efficient full key recovery attack, and moreover that ciphertexts can be decrypted faster with the public parameters alone than the time it takes to generate the public key.

Keywords: fully homomorphic encryption; ideal lattices; Chinese remainder theorem; cryptanalysis

1. Introduction

Fully Homomorphic Encryption (FHE) is a fundamental cryptographic primitive that realizes the notion of privacy homomorphism first proposed in [1], by enabling arbitrary computations to be carried out directly on ciphertexts without access to the decryption key. Although various forms of somewhat homomorphic encryption were proposed previously [2–5] , the development of practical modern FHE schemes was paved by Gentry's groundbreaking paper in 2009 [6].

Most of the FHE schemes used in practice today [7–12] ensure security based on the hardness of standard lattice problems by injecting noise during encryption. Since these schemes inherently accumulate noise as homomorphic operations are repeated, a technique known as bootstrapping is required to refresh this noise level. This technique enables FHE to achieve unbounded evaluation and establishes it as a core technology in fields such as secure outsourced

Copyright: © 2026 by the authors. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).
Publisher's Note: Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

computation or privacy-preserving protocols. While bootstrapping allows FHE to support unbounded evaluations, it remains a primary bottleneck hindering practical deployment. This process of managing and refreshing noise is computationally expensive, making it the principal cause of performance degradation.

To circumvent the computational constraints imposed by bootstrapping, the concept of *Noiseless FHE* has emerged as an alternative paradigm. Unlike modern FHE schemes that rely on injected noise for security, Noiseless FHE aims to achieve unbounded evaluation by leveraging specific algebraic structures where noise is either never introduced or naturally vanishes during decryption.

Early research attempted to design such schemes by masking the ciphertexts instead of generating noisy ciphertexts. However, while noise incurs computational costs, it simultaneously serves as a crucial security buffer that conceals the underlying algebraic structure of the ciphertexts. Consequently, noise-free ciphertexts often exhibit an algebraically exact structure, providing adversaries with clues to inversely extract the decryption map. One of the earliest proposals exploited the non-commutative property of matrix multiplication, exemplified by the scheme called Matrix Operation for Randomization or Encryption (MORE) proposed in [13]. This approach attempted to implement noiseless operations via conjugation, multiplying the plaintext matrix by invertible matrices. However, it was shown that plaintexts could be recovered without the private key by using the fact that linear algebraic invariants, such as eigenvalues and traces, are preserved under conjugation [14]. Later variants of MORE were also broken by ciphertext-only attacks based on linear algebraic structure leakage [15,16].

To evade vulnerabilities arising from matrix invariants, subsequent designs incorporated non-commutative rings [17] where matrix representations are difficult to derive. However, these too were broken by comparison-based attacks because the exact nature of noiseless ciphertexts failed to conceal the underlying algebraic structure [18]. Furthermore, schemes based on non-associative octonion algebras were proposed to thwart algebraic attacks [19,20] but linear algebraic attacks remained feasible as invariants are exposed during the evaluation of operations. Additionally, frameworks for noiseless FHE based on group theory were suggested [21], yet these approaches also faced limitations in completely excluding linear constraints [21].

Despite these numerous instances of cryptanalysis, the demand for efficient noiseless FHE continues to drive new research. Recently, schemes have been proposed where noise exists during encryption but is eliminated during decryption. In particular, Zheng et al. [22] propose a new *unbounded FHE* scheme based on ideal lattices and the Chinese Remainder Theorem (CRT).

Another recent series of work that deserves a mention is the one of Chun et al. [23,24], also uses the term "noiseless" FHE, based on the approximate greatest common divisor (AGCD) problem. However, those schemes are basically identical to the original AGCD-based construction of van Dijk et al. [25] (with the only possible difference that they consider larger moduli than 2, which was obviously considered and shown to be bootstrappable long ago [26], and also in some sense always worse than the binary case [27]). In particular, these schemes do not do away with bootstrapping or ciphertext noise at all. What the authors mean by "noiseless" is exact decryption in the standard sense, in contrast to approximate or stochastic FHE schemes such as [9,28], which intentionally allow small decryption inaccuracies.

Against this background, in we present a concrete cryptanalysis of the scheme proposed in [22]. By exploiting the underlying CRT structure and properties of ideal lattices, we introduce two attack strategies capable of recovering the plaintext using only public information. Our results reaffirm that FHE designs based on noiseless algebraic structures remain inherently vulnerable to classical algebraic attacks.

Cryptanalysis of Zheng et al.'s Unbounded FHE Scheme [22]

Unlike earlier noiseless FHE schemes, Zheng et al.'s unbounded FHE scheme [22] takes a different approach to avoiding the bootstrapping bottleneck. Their ciphertexts contain noise like modern FHE schemes. However, they adopted a specific algebraic structure to completely eliminate this noise during decryption.

The feasibility of this structure arises from the underlying algebraic setting and the key properties. The entire scheme is constructed over the ring of integers of a cyclotomic field. Secret keys are defined as ideal lattices within this ring and each secret key has an associated one-dimensional modulus. This modulus defines the plaintext space for its respective channel. The CRT bridges these individual plaintext spaces directly to a single global ciphertext space.

During encryption, the scheme maps values from the smaller spaces into the global space. The sender introduces random polynomials into the public keys to perturb the ciphertext. The decryption process applies a modulo operation with the secret ideal. The injected noise terms are designed specifically as multiples of this secret ideal, so the noise inherently vanishes to zero during the modulo operation to cleanly recover the plaintext.

Prior attempts to construct unbounded FHE schemes without a bootstrap procedure have proved insecure. In this paper, we demonstrate that the unbounded FHE scheme proposed by Zheng et al. [22] is no exception to this

trend. Their CRT structure constitutes a critical security vulnerability, so an attacker can exploit the mathematical properties of the CRT to extract the plaintexts from purely public information.

We propose two primary attack strategies. First, we present a **public-key-only attack** for full key recovery, which combines the CRT structures shared across the public keys to deduce the secret keys. Second, we introduce a **public-parameter-only attack** for direct message decryption, which extracts secret information from the public parameters enabling the decryption of ciphertexts without the secret key through a simple polynomial evaluation. The first attack applies to the general framework introduced by Zheng et al. [22], and achieves key recovery in polynomial time. In particular, it also violates unbreakability under chosen-plaintext attack (UBK-CPA), a basic key-recovery notion requiring that no efficient adversary can recover the secret key, or equivalent decryption information, from public information. The second attack applies to the specific instantiation of the framework that Zheng et al. [22] call their "practical instantiation", and while it does not technically recover the secret key, it efficiently recovers enough information to decrypt any plaintext, breaking the one-wayness of the scheme in seconds or less even for very large parameters. Both attacks are validated experimentally, and the attack code is publicly available on GitHub.

2. Preliminaries

We review some key definitions and a theorem essential for describing the target scheme in [22] and additional definitions are provided in Appendix A.

The ambient space of the target scheme in [22] admits two complementary viewpoints. Algebraically, it is defined over the ring of integers of a number field and its quotient rings by ideals. Geometrically, via a coefficient embedding, ideals correspond to full-rank $\mathbb{Z}$-lattices in an ambient vector space. Below, we recall the required algebraic notions and their lattice counterparts.

2.1. Basic Algebraic Number Theory Definitions and Ideal Lattices

Let p be a positive integer and let $\phi_p(x)$ be the p-th cyclotomic polynomial of degree $n = \varphi(p)$. Throughout this paper, we fix the cyclotomic field $K = \mathbb{Q}(\zeta_p) \cong \mathbb{Q}[x]/\langle\phi_p(x)\rangle$, and denote its ring of integers by $\mathcal{O}_K = \mathbb{Z}[\zeta_p] \cong \mathbb{Z}[x]/\langle\phi_p(x)\rangle$.

Then, the ring of integers $\mathcal{O}_K$ is a Dedekind domain. In particular, every non-zero proper ideal $I \subset \mathcal{O}_K$ factors uniquely as a product of prime ideals $I = \prod_{i=1}^{k} \mathfrak{p}_i^{e_i}$, where the $\mathfrak{p}_i$ are distinct prime ideals and $e_i \geq 1$.

Definition 1 (Principal Ideal)**.** *An ideal $I \subseteq \mathcal{O}_K$ is called a principal ideal if it is generated by a single element $\alpha \in \mathcal{O}_K$, and we write*

$$I = \langle\alpha\rangle = \{\alpha r \mid r \in \mathcal{O}_K\}.$$

In our attack algorithms, we utilize algebraic and ideal norms.

Definition 2 (Algebraic and Ideal Norm)**.** *For $\alpha \in K$, the algebraic norm $N_{K/\mathbb{Q}}(\alpha)$ is the product of all $\mathbb{Q}$-embeddings of α into $\mathbb{C}$. For a non-zero integral ideal $I \subseteq \mathcal{O}_K$, the ideal norm is*

$$N(I) := |\mathcal{O}_K/I|.$$

In particular, for a non-zero principal ideal $I = \langle\alpha\rangle$ with $\alpha \in \mathcal{O}_K$, we have $N(I) = |N_{K/\mathbb{Q}}(\alpha)|$.

To relate ideals to lattices, we use the fact that $\mathcal{O}_K$ is a free $\mathbb{Z}$-module of rank $n = [K : \mathbb{Q}]$. After fixing a $\mathbb{Z}$-basis of $\mathcal{O}_K$, we may identify $\mathcal{O}_K$ with $\mathbb{Z}^n$ as $\mathbb{Z}$-modules. Under this identification, any non-zero integral ideal $I \subseteq \mathcal{O}_K$ corresponds to a full-rank $\mathbb{Z}$-lattice in $\mathbb{Z}^n$.

Definition 3 (Ideal Lattice)**.** *Let $I \subseteq \mathcal{O}_K$ be a non-zero integral ideal. Via the above identification $\mathcal{O}_K \cong \mathbb{Z}^n$, the image of I forms a full-rank lattice in $\mathbb{Z}^n$. We call this lattice an ideal lattice associated with I.*

2.2. Chinese Remainder Theorem for Ideals

The target scheme connects the structurally decomposed spaces with the combined space using the Chinese Remainder Theorem for pairwise coprime ideals.

Definition 4 (Coprime Ideals)**.** *Two ideals $I, J \subseteq \mathcal{O}_K$ are called coprime if $I + J = \mathcal{O}_K$. A family $\{I_i\}_{i=1}^{k}$ is pairwise coprime if $I_i + I_j = \mathcal{O}_K$ for all $i \neq j$.*

Theorem 1 (Chinese Remainder Theorem)**.** *Let $I_1, \ldots, I_k$ be pairwise coprime ideals in $\mathcal{O}_K$, and let $I = \prod_{i=1}^{k} I_i$. Then there is a natural ring isomorphism*

$$\psi : \mathcal{O}_K/I \xrightarrow{\sim} \prod_{i=1}^{k} (\mathcal{O}_K/I_i),$$

given by

$$\psi(a \bmod I) \;=\; (a \bmod I_1, \ldots, a \bmod I_k).$$

In particular, ring operations in $\mathcal{O}_K/I$ correspond to component-wise operations in $\prod_{i=1}^{k}(\mathcal{O}_K/I_i)$.

Lattice viewpoint. The above Theorem 1 can also be interpreted from the lattice viewpoint.

Fix a $\mathbb{Z}$-basis identification $\mathcal{O}_K \cong \mathbb{Z}^n$, and let $L_i \subseteq \mathbb{Z}^n$ denote the full-rank lattice corresponding to I_i. The coprimality condition implies $L_i + L_j = \mathbb{Z}^n$, and the CRT induces an isomorphism of additive groups

$$\mathbb{Z}^n/L \;\cong\; \prod_{i=1}^{k} \big(\mathbb{Z}^n/L_i\big),$$

where L corresponds to I.

3. Zheng et al.'s Unbounded FHE Scheme

In this section, we review the unbounded FHE scheme targeted by our cryptanalysis, originally proposed in [22]. The core of the scheme is utilizing the CRT over rings to bridge plaintext and ciphertext spaces. In [22], the authors first present a general unbounded FHE framework and then instantiate it using principal ideal lattices and cyclotomic field properties. Our attack analysis specifically targets this practical instantiation. In the following, we summarize the general framework and then detail the practical cyclotomic instantiation in Construction 1.

3.1. General Framework

Let $\phi(x) \in \mathbb{Z}[x]$ be a monic polynomial of degree n and set $R \coloneqq \mathbb{Z}[x]/\langle \phi(x) \rangle$. Each element of R admits a unique representative polynomial of degree $< n$, and the coefficient embedding $\tau : R \to \mathbb{Z}^n$ maps such a representative to its coefficient vector. In Section 2 [22], $\mathbb{Z}^n$ is equipped with component-wise addition and a convolution product $\cdot$ (denoted $\otimes$ in [22]) so that τ becomes a ring isomorphism. Equivalently, the operation $\cdot$ corresponds to polynomial multiplication in R under the identification induced by τ, and hence $(\mathbb{Z}^n, +, \cdot)$ forms a commutative ring isomorphic to R.

Under the above identification, a subset $I \subset \mathbb{Z}^n$ that is an ideal of the ring $(\mathbb{Z}^n, +, \cdot)$ is simultaneously a full-rank ideal lattice, and $\mathbb{Z}^n/I$ can be viewed as a quotient ring. Note that an ideal lattice I has finite index, and we write this index as

$$t(I) \;:=\; |\mathbb{Z}^n/I|.$$

In the setting of [22], the relevant ideal lattices behave one-dimensionally in the sense that the diagonal elements of their Hermite Normal Form (HNF) matrices are of the form $(t(I), 1, \ldots, 1)$, so the index is captured by the first diagonal entry of the HNF matrix. Equivalently, the quotient $\mathbb{Z}^n/I$ has a single nontrivial cyclic component of size $t(I)$. Algebraically, this happens if and only if the prime ideals occurring in the decomposition of I all have distinct prime absolute norms. In particular, it holds for ideals with square-free norm, and a randomly chosen ideal typically has this property. Under this structure, congruence modulo I can therefore be represented by reduction modulo the scalar $t(I)$. Accordingly, this scalar $t(I_i)$ is used in [22] as the i-th plaintext modulus, namely $t_i \coloneqq t(I_i)$, and the plaintext space is defined as $\mathcal{P} \coloneqq \bigoplus_{i=1}^{m} \mathbb{Z}_{t_i}$.

Let $I_1, \ldots, I_m \subset R$ be pairwise coprime ideals and write $I \coloneqq \prod_{i=1}^{m} I_i$. Then the CRT yields a canonical ring isomorphism

$$R/I \;\cong\; \prod_{i=1}^{m} R/I_i.$$

In the construction of [22], each component R/I_i induces the one-dimensional plaintext modulus t_i described above. Thus a single ciphertext ring R/I corresponds to m plaintext slots in $\bigoplus_{i=1}^{m} \mathbb{Z}_{t_i}$.

The public key contains CRT recombination elements $A_i \in R$ satisfying

$$A_i \equiv 1 \pmod{I_i} \quad \text{and} \quad A_i \equiv 0 \pmod{I_j} \;\; (j \neq i).$$

A public-key randomization step produces $A'_1, \ldots, A'_m$. Given slot values $u_i \in \mathbb{Z}_{t_i}$, one chooses representatives $\bar{u}_i \in \mathbb{Z}$ and views them as constant elements of R. Encryption packs the slots into a single ciphertext $c = \sum_{i=1}^{m} \bar{u}_i A'_i \in R$.

The secret key consists of the ideals $(I_1, \ldots, I_m)$. To recover the i-th slot, the decryptor reduces the ciphertext modulo I_i and then decodes the resulting embedded integer modulo t_i.

Construction 1: Practical unbounded FHE scheme in [22]

- **Setup:** Fix an odd prime p and set $n \coloneqq p-1$. Let ζ_p be a primitive p-th root of unity and $\phi_p(x) \coloneqq x^{p-1} + \cdots + x + 1$. Set $K \coloneqq \mathbb{Q}(\zeta_p)$, $\mathcal{O}_K \coloneqq \mathbb{Z}[\zeta_p]$, and $R \coloneqq \mathbb{Z}[x]/\langle \phi_p(x) \rangle$. Let $\tau : R \to \mathbb{Z}^n$ be the coefficient embedding and let $\cdot$ be the induced product on $\mathbb{Z}^n$. Then we can obtain the following isomorphism,
$$\mathcal{O}_K \cong R \cong (\mathbb{Z}^n, +, \cdot).$$

- **Secret Key Generation:** Choose distinct primes $q_1, \ldots, q_m$ with $q_i \neq p$. For each $i \in [m]$, set
$$\alpha_i \coloneqq \zeta_p^{n-1} + q_i, \qquad I_i \coloneqq \langle \alpha_i \rangle \subseteq \mathcal{O}_K \cong R.$$
Output $\mathsf{SK} \coloneqq (I_1, \ldots, I_m)$ and define
$$t_i \coloneqq t(I_i) = q_i^n - q_i^{n-1} + \cdots - q_i + 1.$$

- **Public Key Generation:** For each $i \in [m]$, set $d_i \coloneqq \prod_{j \neq i} \alpha_j \in R$ and find $\mathcal{D}_i \in R$ such that $d_i \mathcal{D}_i \equiv 1 \pmod{I_i}$. Set $A_i \coloneqq d_i \mathcal{D}_i \in R$, so that
$$A_i \equiv 1 \pmod{I_i} \quad \text{and} \quad A_i \equiv 0 \pmod{I_j} \quad \text{for} \quad j \neq i.$$
Sample $a = (a_1, \ldots, a_m) \leftarrow \chi$ over $\prod_{j=1}^{m} \mathbb{Z}_{t_j}$. For each j, let $\bar{a}_j \in R$ denote the constant element corresponding to a chosen representative of a_j in $\mathbb{Z}$. For each $i \in [m]$, compute an element $E_i \in R$ such that
$$E_i \equiv 1 \pmod{I_i} \quad \text{and} \quad E_i \equiv \bar{a}_j \pmod{I_j} \text{ for all } j \neq i.$$
Set $A'_i \coloneqq A_i E_i$ and output $\mathsf{PK} \coloneqq (A'_1, \ldots, A'_m)$.

- **Encryption:** Let the plaintext space $\mathcal{P} \coloneqq \bigoplus_{i=1}^{m} \mathbb{Z}_{t_i}$. Given the plaintext $\mathbf{u} = (u_1, \ldots, u_m) \in \mathcal{P}$, choose representatives $\bar{u}_i \in \mathbb{Z}$ and view them as constant elements of R. Output the ciphertext
$$c \coloneqq \sum_{i=1}^{m} \bar{u}_i A'_i \in R.$$

- **Decryption:** Given $c \in R$ and $\mathsf{SK} = (I_1, \ldots, I_m)$, for each i, compute $c \bmod I_i$ and recover $u_i \in \mathbb{Z}_{t_i}$ from this residue. Output $\mathbf{u} = (u_1, \ldots, u_m)$.

- **Homomorphic evaluation:** Addition and multiplication are performed in the ciphertext ring R.

3.2. Practical Instantiation

For a practical implementation, the Construction 1 is instantiated over the p-th cyclotomic field in [22]. Let p be an odd prime and set $n \coloneqq p-1$, and let ζ_p be a primitive p-th root of unity with minimal polynomial $\phi_p(x) = 1 + x + \cdots + x^{p-1} \in \mathbb{Z}[x]$.

We work in $K \coloneqq \mathbb{Q}(\zeta_p)$ with ring of integers $\mathcal{O}_K \coloneqq \mathbb{Z}[\zeta_p] \cong \mathbb{Z}[x]/\langle \phi_p(x) \rangle$. Combining this isomorphism with the coefficient embedding τ, elements of $\mathcal{O}_K$ are represented in $(\mathbb{Z}^n, +, \cdot)$. Then the following isomorphisms can be obtained.
$$\mathbb{Z}[\zeta_p] \cong \mathbb{Z}[x]/\langle \phi_p(x) \rangle \cong (\mathbb{Z}^n, +, \cdot).$$

The secret key consists of principal ideals $I_1, \ldots, I_m \subseteq \mathcal{O}_K$. Choose pairwise distinct primes $q_1, \ldots, q_m$ with $q_i \neq p$ and define $\alpha_i \coloneqq \zeta_p^{n-1} + q_i \in \mathcal{O}_K$ and $I_i \coloneqq \langle \alpha_i \rangle$. For each i, the associated one-dimensional modulus is

$$t_i := t(I_i) = q_i^n - q_i^{n-1} + \cdots - q_i + 1.$$

Public key randomization and the encryption/decryption procedures follow the general framework in 3.1. Construction 1 summarizes the resulting scheme.

Remark 1. *Correctness follows from the CRT structure of the pairwise coprime ideals $I_1, \ldots, I_m$ and the congruences satisfied by $(A'_1, \ldots, A'_m)$. Let $\mathbf{u} = (u_1, \ldots, u_m) \in \mathcal{P}$ and let $c = \sum_{j=1}^{m} \bar{u}_j A'_j \in R$, where each $\bar{u}_j$ is a representative of $u_j \in \mathbb{Z}_{t_j}$ viewed as a constant element of R. Fix $i \in [m]$. Using $A'_i \equiv 1 \pmod{I_i}$ and $A'_j \equiv 0 \pmod{I_i}$ for $j \neq i$, it holds that $c \equiv \bar{u}_i \cdot 1 + \sum_{j \neq i} \bar{u}_j \cdot 0 \equiv \bar{u}_i \pmod{I_i}$. Therefore, reducing c modulo I_i eliminates all terms except the i-th component and yields a residue congruent to $\bar{u}_i$ modulo I_i. Applying the decoding rule modulo t_i recovers $u_i \in \mathbb{Z}_{t_i}$. Repeating this procedure for all i recovers $\mathbf{u}$.*

4. Cryptanalysis and Two Main Attack Strategies

This section presents two main attack strategies against Construction 1. The first strategy recovers the full secret keys from the public keys, thereby directly breaking the UBK-CPA security. The recovered keys can then be used to decrypt any ciphertext. The second strategy exploits the publicly given parameters to recover the hidden parameters, which ultimately enables plaintext recovery. The two strategies are detailed in Sections 4.1 and 4.2, and are summarized in Algorithms 1–3, respectively.

Before describing the attacks, we will give two simple observations indicating that Construction 1 cannot even satisfy IND-CPA security, which requires that no efficient adversary can distinguish encryptions of two chosen plaintexts with non-negligible advantage. In [22], the authors claim that the scheme can achieve IND-CPA security through a public-key randomization procedure. Since the randomized public key $\mathsf{PK} = (A'_1, \ldots, A'_m)$ is generated using randomness sampled from a distribution χ, the encryptions of the same plaintext under independently generated public keys are different. However, this claim is incorrect.

First, randomness is only involved in the public-key generation procedure. Although the public key contains some ”noise” so that encryptions appear randomized, this noise is completely eliminated during decryption due to the CRT-based structure of the public key. As a result, the decrypted plaintext contains no residual noise, so the scheme looks like an unbounded FHE. However, for a fixed PK, the encryption algorithm is deterministic. Consequently, repeated encryptions of the same plaintext under a fixed PK always yield the same ciphertext, which directly violates IND-CPA security.

Second, the decryption procedure is inherently linear. In the decryption procedure, it reduces the ciphertext modulo each secret ideal I_i and then reduces the resulting value modulo the fixed integer t_i to recover the i-th plaintext slot. Both steps are linear operations, and thus the overall decryption map can be viewed as a fixed linear transformation from the ciphertext space to the plaintext space. Therefore, if there is an adversary $\mathcal{A}$ who obtains a sufficient number of ciphertext–plaintext pairs, $\mathcal{A}$ can interpolate this linear map and then recover the plaintext of any ciphertext, again contradicting IND-CPA security.

These observations already suggest a generic vulnerability. Our main attack strategies are strictly stronger than the above considerations. We will show that Construction 1 leaks enough structure in the publicly given information to recover the plaintext without collecting any ciphertext–plaintext pairs.

Construction 1 is further claimed in [22] to be secure based on the worst-case hardness of lattice problems. However, this claim is also flawed. We defer a detailed discussion to Appendix C.

4.1. Public-Key-Only Attack: Recovering Full Secret Keys and Plaintexts

The first attack strategy uses only the public keys to reconstruct the secret keys and then uses the recovered keys to decrypt the plaintexts. The main idea of this attack is to exploit the CRT structure embedded in the public keys and it proceeds in two phases. Phase 1 (Secret key recovery) reconstructs the secret ideals, or equivalently, recovers other ideals that are sufficient to perform decryption correctly. Phase 2 (Plaintext recovery) uses the recovered keys from Phase 1 to extract the plaintexts from ciphertexts.

High-level idea. Fix a target ideal index $i \in [m]$. By construction, the public keys $\mathsf{PK} = \{A'_j\}_{j \in [m]}$ satisfy the CRT congruences $A'_j \equiv 0 \pmod{I_i}$ for all $j \neq i$, and $A'_i \equiv 1 \pmod{I_i}$. Using these public keys, consider linear combinations of the form

$$x = \sum_{j \neq i} r_j A'_j,$$

where $r_j \in \mathbb{Z}$ are chosen by $\mathcal{A}$ and are embedded into R via $\mathbb{Z} \hookrightarrow R$. Since each $A'_j \equiv 0 \pmod{I_i}$ for $j \neq i$, it

follows that

$$x \equiv 0 \pmod{I_i},$$

and hence $x \in I_i$ for any choice of the coefficients $\{r_j\}_{j \neq i}$. For any $k \neq i$, the same CRT structure yields

$$x \equiv r_k \pmod{I_k},$$

because $A'_k \equiv 1 \pmod{I_k}$ and $A'_j \equiv 0 \pmod{I_k}$ for $j \neq k$. Consequently, $x \in I_k$ holds if and only if $x \equiv 0 \pmod{I_k}$. Since $x \equiv r_k \pmod{I_k}$ and r_k is an embedded integer, the definition of the one-dimensional modulus $t_k = t(I_k)$ implies that $x \in I_k$ is equivalent to

$$r_k \equiv 0 \pmod{t_k}.$$

Therefore, choosing coefficients such that $r_k \not\equiv 0 \pmod{t_k}$ for every $k \neq i$ produces samples x that lie in I_i but in none of the other ideals I_k.

By generating sufficiently many such samples $x^{(1)}, \ldots, x^{(s)} \in I_i$ and forming the ideal lattice they generate, one can reconstruct the target ideal I_i with high probability. Once such an ideal I_i is recovered, the i-th plaintext component can be extracted from any ciphertext.

4.1.1. Phase I: Secret Key Recovery

We describe how to recover the i-th secret decryption key information I_i using only the public keys $\mathsf{PK} = (A'_1, \ldots, A'_m)$ with the plaintext moduli $(t_1, \ldots, t_m)$.

Generating samples. For $\ell = 1, \ldots, s$, sample $r^{(\ell)} \in \mathbb{Z}^m$ such that

$$r_i^{(\ell)} = 0 \qquad \text{and} \qquad r_j^{(\ell)} \not\equiv 0 \pmod{t_j} \text{ for all } j \neq i,$$

and set

$$x^{(\ell)} := \sum_{j \neq i} r_j^{(\ell)} A'_j \in R.$$

By construction, each $x^{(\ell)}$ lies in the target ideal I_i, due to the condition $r_k^{(\ell)} \not\equiv 0 \pmod{t_k}$, while avoiding every other ideal I_k for $k \neq i$.

Recovering an ideal from samples. Define the ideal generated by the samples $(x^{(1)}, \ldots, x^{(s)})$ by

$$J_i := \langle x^{(1)}, \ldots, x^{(s)} \rangle = \sum_{\ell=1}^{s} \langle x^{(\ell)} \rangle \subseteq I_i.$$

Since I_i is a principal ideal, write $I_i = \langle g_i \rangle$. Then every sample can be written as $x^{(\ell)} = g_i \cdot y^{(\ell)}$ for some $y^{(\ell)} \in \mathcal{O}_K$, and hence

$$J_i = \langle g_i y^{(1)}, \ldots, g_i y^{(s)} \rangle = \langle g_i \rangle \langle y^{(1)}, \ldots, y^{(s)} \rangle.$$

Therefore $J_i = \langle g_i \rangle$ holds exactly when the cofactors generate the unit ideal. Heuristically, for generic samples generated as above, the cofactors share no nontrivial common ideal factor after a small number of samples, and thus J_i stabilizes to I_i with high probability.

When $J_i = I_i$, the i-th plaintext component can be recovered by the direct reduction procedure shown in Phase II of Algorithm 1. More generally, even when $J_i \subsetneq I_i$, the recovered ideal may still suffice to extract the i-th plaintext component under the conditions described in Remark 2.

4.1.2. Phase II: Plaintext Recovery

Given the recovered ideals $J_1, \ldots, J_m$ from Phase I, plaintext recovery follows the decryption procedure.

Let $c \in R$ be a ciphertext. For each $i \in [m]$, compute the residue class $c \bmod J_i$. When $J_i = I_i$, reduction modulo J_i cancels all components other than the i-th one and yields a value congruent to u_i under the canonical embedding $\mathbb{Z} \hookrightarrow R$. The i-th slot is then recovered by determining the unique $u_i \in \mathbb{Z}_{t_i}$ such that

$$c - u_i \in J_i,$$

where u_i is viewed as an element of R via the embedding $\mathbb{Z} \hookrightarrow R$. Repeating this for all $i \in [m]$ yields the plaintext

$$\mathbf{u} = (u_1, \ldots, u_m) \in \bigoplus_{i=1}^{m} \mathbb{Z}_{t_i}.$$

Remark 2. *Exact recovery of I_i is not strictly necessary for extracting the i-th plaintext component. It suffices to obtain an ideal $J \subset I_i$ such that*

$$A'_j \in J \text{ for all } j \neq i, \quad \text{and} \quad A'_i \notin J.$$

Let $c = \sum_{j=1}^{m} \bar{u}_j A'_j \in R$ be a ciphertext, where each $u_j \in \mathbb{Z}_{t_j}$ is represented by a chosen lift $\bar{u}_j \in \mathbb{Z}$ and viewed in R as a constant polynomial. Reducing modulo J yields

$$c \equiv \bar{u}_i A'_i \pmod{J}.$$

Define

$$S := \{\alpha \in \mathbb{Z}_{t_i} : c - \bar{\alpha} A'_i \in J\},$$

where $\bar{\alpha} \in \mathbb{Z}$ denotes a fixed representative of α. Then S is nonempty since $c - \bar{u}_i A'_i = \sum_{j \neq i} \bar{u}_j A'_j \in J$. Moreover, S contains at most one element. If $\alpha, \beta \in S$, then

$$(\bar{\alpha} - \bar{\beta}) A'_i \in J \subset I_i.$$

Reducing modulo I_i and using $A'_i \equiv 1 \pmod{I_i}$ gives

$$\bar{\alpha} - \bar{\beta} \equiv 0 \pmod{I_i},$$

which implies $\alpha = \beta$ in $\mathbb{Z}_{t_i}$. Therefore $S = \{u_i\}$, and the unique element of S equals the i-th plaintext component.

4.1.3. Public-Key-Only Attack Algorithm and Optimizations

The public-key-only attack is summarized in Algorithm 1. The algorithm follows the two-phase structure described above. However, directly executing this algorithm has some practical computational challenges. The primary bottleneck is the ideal generation step (line 6). As the number of generators s increases, $\mathcal{A}$ must perform exceedingly expensive internal lattice operations to reconstruct a single ideal. Furthermore, $\mathcal{A}$ does not know a priori exactly how many samples are needed to fully recover I_i, which can lead to costly trial-and-error.

Algorithm 1 Public-key-only attack

Require: $\mathsf{PK} = (A'_1, \ldots, A'_m)$, plaintext moduli $t = (t_1, \ldots, t_m)$, and a ciphertext c
Ensure: Plaintext $\mathbf{u} = (u_1, \ldots, u_m)$
Phase I: Secret ideal recovery
1: **for** $i = 1$ to m **do**
2: **for** $\ell = 1$ to s **do**
3: Sample $r^{(\ell)} = (r_1^{(\ell)}, \ldots, r_m^{(\ell)}) \in \mathbb{Z}^m$ s.t. $r_i^{(\ell)} = 0$ and $r_j^{(\ell)} \not\equiv 0 \pmod{t_j}$ for all $j \neq i$
4: Compute $x^{(\ell)} \leftarrow \sum_{j \neq i} r_j^{(\ell)} A'_j \in R$ ▷ Sample in I_i
5: **end for**
6: $J_i \leftarrow \langle x^{(1)}, \ldots, x^{(s)} \rangle \subset R$ ▷ Ideal generated by samples
7: **end for**
Phase II: Plaintext recovery
8: **for** $i = 1$ to m **do**
9: Compute $c_i \leftarrow c \bmod J_i$
10: Recover $u_i \leftarrow c_i \bmod t_i \in \mathbb{Z}_{t_i}$
11: **end for**
12: **return** $\mathbf{u} = (u_1, \ldots, u_m)$

To mitigate this computational burden, the attack can be optimized by minimizing the number of generators. Note that our ambient working space $\mathcal{O}_K$ is a Dedekind domain, and this mathematical structure guarantees that any non-zero ideal can be generated by at most two elements. Thus, if $\mathcal{A}$ can efficiently identify two appropriate samples $x, y \in I_i$, the secret ideal can be reconstructed as $I_i = \langle x, y \rangle$ without invoking heavy ideal arithmetic during the search phase.

This selection can be done by a fast filtering technique based on algebraic norms. For any $x \in I_i$, the principal ideal $\langle x \rangle$ is a subideal of I_i, implying $N(I_i) \mid N(\langle x \rangle)$. By Lemma A2, there is a natural ring isomorphism $\mathcal{O}_K/I_i \cong \mathbb{Z}_{t_i}$. Since the absolute norm of an ideal is the cardinality of its quotient ring, it follows that

$$N(I_i) = |\mathcal{O}_K/I_i| = |\mathbb{Z}_{t_i}| = t_i.$$

Therefore, the absolute norm $|N_{K/\mathbb{Q}}(x)| = N(\langle x \rangle)$ is always a multiple of t_i.

To find a valid pair efficiently, $\mathcal{A}$ can precompute a small set of base samples in I_i. By taking random $\mathbb{Z}$-linear combinations of these base samples using small coefficients, $\mathcal{A}$ generates numerous candidate pairs (x, y) requiring only basic additions and scalar multiplications. For each candidate pair, $\mathcal{A}$ calculates their algebraic norms and applies the following condition:

$$\gcd\big(|N(x)|, |N(y)|\big) = t_i.$$

This GCD condition mathematically guarantees the correct recovery of the secret ideal. Letting $J = \langle x, y \rangle$, the inclusion $J \subseteq I_i$ implies $N(I_i) \mid N(J)$, meaning $t_i \mid N(J)$. Conversely, since $x, y \in J$, the properties of the norm dictate that $N(J) \mid \gcd(|N(x)|, |N(y)|) = t_i$. Therefore, $N(J) = t_i$. Since $J \subseteq I_i$ and their norms are identical, it strictly follows that $J = I_i$.

By employing this filtering technique, the repetitive workload of $\mathcal{A}$ is shifted from expensive lattice operations to highly efficient integer arithmetic. The heavy ideal generation is executed exactly once per slot using exactly two generators. This refined procedure, referred to as the optimized public-key-only attack, is summarized in Algorithm 2.

Algorithm 2 Optimized public-key-only attack

Require: $\mathsf{PK} = (A'_1, \ldots, A'_m)$, plaintext moduli $t = (t_1, \ldots, t_m)$, base size L, bound B, ciphertext c
Ensure: Plaintext $\mathbf{u} = (u_1, \ldots, u_m)$

Phase I: Ideal recovery
for $i = 1$ to m **do**
Step 1: Base sample generation
  for $k = 1$ to L **do**
    Sample $r \in \mathbb{Z}^m$ with $r_i = 0$ and $r_j \not\equiv 0 \pmod{t_j}$ for $j \neq i$
    Compute $g_k \leftarrow \sum_{j \neq i} r_j A'_j \in R$
  end for
Step 2: Candidate generation and GCD filtering
  loop
    Sample small random coefficient vectors $\mathbf{v}, \mathbf{w} \in [-B, B]^L$
    Compute candidate pair $x \leftarrow \sum_{k=1}^{L} v_k g_k$ and $y \leftarrow \sum_{k=1}^{L} w_k g_k$
    Compute $N_x \leftarrow |N_{K/\mathbb{Q}}(x)|$, and $N_y \leftarrow |N_{K/\mathbb{Q}}(y)|$
    if $t_i \mid \gcd(N_x, N_y)$ **then** ▷ Fast integer prefilter
      $J_i \leftarrow \langle x, y \rangle$ ▷ Ideal generated by 2 samples
      if $N(J_i) = t_i$ **then**
        break ▷ Successfully recovered I_i; proceed to next slot
      end if
    end if
  end loop
end for
Phase II: Plaintext recovery
for $i = 1$ to m **do**
  Compute $c_i \leftarrow c \bmod J_i$
  Recover $u_i \leftarrow c_i \bmod t_i \in \mathbb{Z}_{t_i}$
end for
return $\mathbf{u} = (u_1, \ldots, u_m)$

4.2. Public-Parameter-Only Attack: Recovering Secret Parameters and Plaintext

Our second attack strategy uses the public parameters $\{t_i\}_{i \in [m]}$. We show that it is possible to recover the hidden secret parameters $\{q_i\}_{i \in [m]}$ directly from these moduli, which immediately leads to the decryption of any arbitrary ciphertext. In [22], it is claimed that recovering q_i from the public t_i is computationally intractable. Recall

that in Construction 1, the modulus is given by

$$t_i \;=\; q_i^n - q_i^{n-1} + \cdots - q_i + 1,$$

and the discussion interprets recovering q_i from t_i as solving the high-degree equation

$$x^n - x^{n-1} + \cdots - x + 1 \;=\; t_i.$$

Relying on Galois theory, they assert that no general closed-form solution method is known for algebraic equations of degree $n \geq 5$. While this classical result holds for finding the generic roots of arbitrary high-degree polynomials, this reasoning is not applicable in the present setting. In Construction 1, $n = p - 1$ for an odd prime p, and q_i is known a priori to be a positive integer. Under these constraints, the polynomial defines a strictly monotonically increasing function over the positive integers. Consequently, q_i can be recovered via a monotone binary search or numerical root extraction method.

In Construction 1, each t_i is defined as a polynomial in q_i and this polynomial is given as a public description of the scheme. Even if one assumes this formula is not explicitly given, it can be derived from the algebraic definition of the secret ideal and the cyclotomic relation. Lemma A1 in Appendix B formalizes this derivation.

High-level idea. This attack shows that the secret ideal is not actually needed for decryption. The public moduli already determine enough information to construct a decryption evaluation map directly. After recovering q_i from t_i, $\mathcal{A}$ defines

$$a_i \equiv -q_i^{-1} \pmod{t_i}$$

and constructs the evaluation map

$$\psi_i : R = \mathbb{Z}[x]/\langle \phi_p(x) \rangle \longrightarrow \mathbb{Z}_{t_i}, \qquad \psi_i(f) = f(a_i) \bmod t_i.$$

Under the polynomial representation of R, both a ciphertext c and each public-key element A'_j can be viewed as polynomials of degree less than $p - 1$. Thus, for the i-th plaintext slot, $\mathcal{A}$ can evaluate the ciphertext polynomial and the corresponding public-key polynomial at a_i modulo t_i. Then, this evaluation map acts as the decryption map for the i-th component, giving

$$\psi_i(c) \equiv u_i \cdot \psi_i(A'_i) \pmod{t_i}.$$

Since A'_i is public, $\mathcal{A}$ can also compute $\psi_i(A'_i) = A'_i(a_i) \bmod t_i$ and invert it modulo t_i. Therefore, the plaintext slot is recovered as

$$u_i \equiv c(a_i) \cdot \big(A'_i(a_i)\big)^{-1} \pmod{t_i}.$$

Hence, plaintext recovery reduces to simple polynomial evaluations and one modular inversion for each plaintext slot.

4.2.1. Step 1: Recovering Each q_i from the Public Parameter t_i

By Lemma A1, each public one-dimensional modulus t_i is explicitly expressed in terms of the secret prime q_i as:

$$t_i = \frac{q_i^p + 1}{q_i + 1}.$$

For integers $q_i \geq 2$, the algebraic function $f(q) = \frac{(q^p+1)}{(q+1)} = q^{p-1} - q^{p-2} + \cdots + 1$ is strictly monotonically increasing. Consequently, the value of t_i uniquely determines the secret parameter q_i. Therefore, $\mathcal{A}$ can efficiently recover q_i in polynomial time by performing a discrete monotone binary search over the integers or by employing standard numerical root finding algorithms.

4.2.2. Step 2: Plaintext Recovery via Polynomial Evaluation

Once the secret parameters $\{q_i\}_{i\in[m]}$ are recovered, $\mathcal{A}$ can reconstruct the secret ideal I_i for the decryption. But $\mathcal{A}$ can bypass this complex ideal lattice generation computation and instead, $\mathcal{A}$ can decrypt the ciphertext simply by evaluating it as a polynomial by simply constructing a decryption function.

Recall from Construction 1 that under the isomorphism

$$\mathbb{Z}[\zeta_p] \cong \mathbb{Z}[x]/\langle \phi_p(x) \rangle,$$

any ring element $c \in \mathbb{Z}^n$ can be naturally viewed as a polynomial $c(x) \in \mathbb{Z}[x]$ of degree strictly less than $p-1$. To decrypt the i-th plaintext slot, $\mathcal{A}$ can construct a ring homomorphism ψ_i that simulates the reduction modulo I_i.

To achieve this simulation, ψ_i must annihilate the secret ideal I_i. In the polynomial representation, I_i is generated by $1 + q_i x$. Therefore, forcing this generator to vanish at an evaluation point $x = a_i$ natively requires the condition:

$$1 + q_i a_i \equiv 0 \pmod{t_i} \implies a_i \equiv -q_i^{-1} \pmod{t_i}.$$

We formally show that evaluating polynomials at $a_i \pmod{t_i}$ induces a well-defined decryption ring homomorphism map ψ_i such that $\psi_i(I_i) = 0$ in Lemma A2 in Appendix B.

This elimination of I_i translates the CRT properties of the public keys into simple modular arithmetic. By the design of the scheme, the public keys satisfy $A'_j \equiv 0 \pmod{I_i}$ for all $j \neq i$. This means $A'_j \in I_i$, and consequently, any element in I_i is mapped to zero under ψ_i. Thus, $\psi_i(A'_j) \equiv 0 \pmod{t_i}$. Similarly, because A'_i is coprime to I_i, its evaluation yields an invertible element $\psi_i(A'_i) \in \mathbb{Z}_{t_i}^{\times}$.

Therefore, to extract the i-th plaintext u_i from the ciphertext $c(x) = \sum_{j=1}^{m} \bar{u}_j A'_j(x)$, $\mathcal{A}$ simply evaluates $c(x)$ at the decryption point a_i:

$$\psi_i(c(x)) \equiv u_i \cdot \psi_i(A'_i(x)) \pmod{t_i}.$$

Since A'_i is public, $\mathcal{A}$ evaluates the public key polynomial $A'_i(x)$ at a_i once to compute its modular inverse. The plaintext is then immediately extracted by evaluating the following polynomials:

$$u_i \equiv c(a_i) \cdot \left(A'_i(a_i)\right)^{-1} \pmod{t_i}.$$

Repeating this polynomial evaluation for each index $i \in [m]$ completely recovers all the plaintext slots $(u_1, \ldots, u_m)$ using only basic integer arithmetic.

4.2.3. Public-Parameter-Only Attack Algorithm

The entire procedure of the public-parameter-only attack is summarized in Algorithm 3. We emphasize that this attack imposes minimal computational overhead on $\mathcal{A}$, as it completely bypasses expensive operations such as lattice reduction or ideal generation. The attack algorithm relies on elementary polynomial evaluations and basic modular arithmetic, making it highly efficient.

Algorithm 3 Public-parameter-only attack

Require: Public key $\mathsf{pk} = (A'_1, \ldots, A'_m)$, plaintext moduli $(t_1, \ldots, t_m)$, and a ciphertext $c \in \mathbb{Z}[x]/\langle \phi_p(x) \rangle$.
Ensure: Plaintext $u = (u_1, \ldots, u_m) \in \bigoplus_{i=1}^{m} \mathbb{Z}_{t_i}$

1: **for** $i = 1$ to m **do**
2: Recover q_i by solving $t_i = \frac{q^p+1}{q+1}$
3: $a_i \leftarrow -q_i^{-1} \bmod t_i$ ▷ Compute the decryption evaluation point
4: $s_i \leftarrow c(a_i) \bmod t_i$ ▷ Evaluate the ciphertext polynomial, i.e., $\psi_i(c)$
5: $e_i \leftarrow A'_i(a_i) \bmod t_i$ ▷ Evaluate the public key polynomial, i.e., $\psi_i(A'_i)$
6: $u_i \leftarrow s_i \cdot e_i^{-1} \bmod t_i$ ▷ Recover the i-th plaintext slot
7: **end for**
8: **return** $\mathbf{u} = (u_1, \ldots, u_m)$

5. Performance Evaluation

In this section, we present experimental results for the attacks against Construction 1 described in Section 4. We implemented the public-key-only attack using Algorithm 1 and its optimized variant Algorithm 2, and we implemented the public-parameter-only attack following Algorithm 3. For the public-key-only attack scenario, we additionally report how much the optimized algorithm improves over the basic attack algorithm.

- **Experiment environment.** All experiments were run on Ubuntu 20.04.6 LTS under WSL2 on a Windows 10 Pro host. The host machine was equipped with an Intel Core i9-12900K and 32 GB RAM. The WSL2 guest was configured with 14 GiB RAM and 8 GiB swap. All experiments were conducted in SageMath 9.0. Our implementation is publicly available at https://github.com/HyewonSung/ACISP2026_submission (accessed on 26 August 2026).
- **Implementation details.** Construction 1 has two natural representations, the ring of integers $\mathcal{O}_K = \mathbb{Z}[\zeta_p]$ and the quotient ring $\mathbb{Z}[x]/\langle \phi_p(x) \rangle$. We implemented the public-key-only attack in the ring-of-integers setting,

since Phase I requires constructing explicit ideal objects from some samples and performing modulo ideal operations. On the other hand, we implemented the public-parameter-only attack in the quotient-ring setting. This attack uses only polynomial arithmetic modulo $\phi_p(x)$ and does not require explicit reconstruction of ideal objects, so the quotient representation avoids the overhead of number-field ideal computations.
For experiments, we selected an odd prime p and distinct primes $q_1, \ldots, q_m$, computed the corresponding moduli $(t_i)_i$, generated keys, and sampled plaintexts uniformly from $\bigoplus_{i=1}^{m} \mathbb{Z}_{t_i}$. Correctness was verified by checking that the recovered plaintext equals the original one. We report end-to-end running time for each attack excluding one-time environment initialization, but we also provide setup timings for completeness.

- **Parameters.** To reflect standard lattice-based cryptographic parameters, we scaled the ring dimension $n = p - 1$ up to 2^{11} and 2^{12} in our experiments. For simplicity, we fixed the set of secret moduli to five distinct primes $q_i \in \{11, 71, 263, 547, 1039\}$. We note that the number of moduli can be arbitrarily extended without loss of generality. In Algorithm 2, we use base size $L = 12$ and bound $B = 2^{10}$.

5.1. Efficiency of the Public-Key-Only Attack

In this section, we evaluate the performance of the public-key-only attacks described in Algorithms 1 and 2. All experimental results are summarized in Table 1.

Table 1. Comparison of attack execution times (in seconds) between the public-key-only attack (Algorithm 1) and its optimized variant (Algorithm 2). 'OOM' denotes an Out of Memory error where the process exceeded the 6 GB memory limit allocated in the SageMath environment. The reported total attack time measures the end-to-end attack latency and excludes the one-time setup cost. While both algorithms eventually encounter memory exhaustion as p grows into the low hundreds, Algorithm 2 demonstrates better scalability, successfully recovering keys and plaintexts for larger parameters than Algorithm 1 before reaching the memory limit.

		Public-Key-Only Attack					**Optimized Public-Key-Only Attack**						
	Prime p	**31**	**83**	**109**	**127**	**131**	**31**	**83**	**109**	**127**	**163**	**181**	**211**
	Setup	0.3	3.7	8	12	14	0.3	3.7	9	14	27	43	66
Attack	Phase I	14	2831	5441	14,490	OOM	10.6	1456	3441	9888	38,821	41,618	OOM
	Phase II	0.06	15	72	185	–	0.07	15	76	176	745	1349	–
Total attack time		14.06	2846	5513	14,675	–	10.7	1471	3518	10,065	39,567	42,967	–

5.1.1. Performance Evaluation

Table 1 summarizes the execution times (in seconds) for both the public-key-only attack (Algorithm 1) and its optimized variant (Algorithm 2). The setup time includes the construction of the ambient space and key generation. As shown in the table, the Setup and Phase II times are comparable between the two methods.

The main performance gap appears in Phase I. Generally, Algorithm 2 is faster for most parameter sets. This advantage typically arises when Algorithm 1 requires more than two generators to recover the ideal. When Algorithm 1 happens to recover the ideal from two generators, it can be faster than Algorithm 2, since it avoids the extra norm and GCD computations used in the optimized procedure. However, this scenario is not guaranteed, since the performance of Algorithm 1 depends on the random selection of samples, regardless of the parameter size.

As parameters grow, memory becomes a dominant constraint. Algorithm 2 offers greater stability than Algorithm 1 by deterministically fixing the number of generators. Despite this stability, Algorithm 2 is also eventually bounded by memory limitations. We provide a detailed analysis of these bottlenecks in the discussion below.

5.1.2. Limitation: Memory Exhaustion

The bottleneck of the public-key-only attack is the prohibitive memory overhead incurred during the ideal reconstruction process (Phase I). Specifically, recovering the ideal requires computing the HNF of a dense integer matrix constructed from k samples. Each sample corresponds to a vector of length n, resulting in an input matrix of dimension $kn \times n$, where the memory complexity is directly proportional to the number of samples. In Algorithm 1, the number of samples k required to generate the ideal is probabilistic. If $\mathcal{A}$ is unlucky, additional generators must be appended, increasing k beyond the minimum. This expansion of the input matrix significantly increases the risk of memory failure even for smaller parameters. Experimental results indicate that for $p = 131$ ($n = 130$), the memory required for HNF computation with Algorithm 1 already exceeds 6 GB, causing overflows on standard workstations. In contrast, Algorithm 2 deterministically constructs the ideal using exactly two generators, thereby

fixing the matrix dimensions at $2n \times n$. This fixed size offers better stability compared to Algorithm 1. As shown in Table 1, Algorithm 2 successfully recovers the ideal for parameters up to $p = 181$ within the same memory limit. However, for parameters larger than $p = 181$, the attack fails to execute.

It is important to note that this failure is mainly due to hardware limitations rather than algorithmic complexity. Since the attack algorithms operate in polynomial time, they would theoretically succeed for larger parameters given sufficient memory resources. However, the prohibitive memory requirement on standard hardware renders the ideal-recovery-based approach impractical for large n. This practical limitation motivates the memory-efficient, ideal-free approach presented in the subsequent section.

5.2. Efficiency of the Public-Parameter-Only Attack

In this section, we report the performance of the public-parameter-only attack implemented according to Algorithm 3. Unlike the public-key-only approach, this attack does not require any ideal computations entirely and relies solely on integer arithmetic and modular evaluation, resulting in improved scalability in our experiments.

Performance Evaluation

Table 2 summarizes the execution times (in seconds) of the proposed public-parameter-only attack (Algorithm 3) for various primes p. The setup time includes the construction of the ambient space and key generation. The experimental results demonstrate that the attack is highly efficient and scalable.

Table 2. Attack execution times (in seconds) for public-parameter-only attack (Algorithm 3). The reported total attack time corresponds to the end-to-end latency, excluding the setup time. All timings are rounded to three decimal places, so the q_i-recovery cost may not be visible in the reported total when it is below 0.001 seconds. The results demonstrate that the attack succeeds within 1 second for $p \lesssim 2^{10}$ and remains highly efficient even for larger parameters $p \geq 2^{12}$.

		Public-Parameter-Only Attack							
	Prime p	**83**	**163**	**181**	**211**	**563**	**1033**	**2267**	**4127**
	Setup	0.01	0.02	0.03	0.04	0.4	2.0	11.4	48
Attack	Recover q_i	0.0001	0.0001	0.0001	0.0001	0.0002	0.0002	0.001	0.001
	Recover plaintexts	0.003	0.007	0.006	0.008	0.067	0.289	2.04	9.4
Total attack time		0.003	0.007	0.006	0.008	0.067	0.289	2.04	9.4

The first step, recovering q_i from t_i, is computationally negligible, taking less than 0.001 seconds across all tested parameters due to the logarithmic complexity of the binary search. Consequently, the total attack time is dominated by the polynomial evaluation step for plaintext recovery. Even for large parameters where $p \geq 2^{12}$, which would be computationally prohibitive for ideal-lattice-based approaches, our attack successfully recovers the plaintext in just 9.4 seconds. This confirms that the proposed attack strategy effectively breaks the scheme in practical time for any realistic choice of p without the bottlenecks associated with ideal operations.

6. Conclusions

Unbounded FHE aims to support arbitrarily many homomorphic operations without requiring bootstrapping to refresh ciphertexts. In this paper, we cryptanalyze the recently proposed unbounded FHE scheme of Zheng et al. [22]. This scheme achieves its unbounded property by using a CRT decomposition over ideal lattices. They first present a general construction of the scheme and further provide a practical instantiation over a cyclotomic field using specially structured principal ideals.

We present two attacks exploiting the algebraic structures of the general construction and its practical instantiation, respectively, and also show that the scheme does not even satisfy standard IND-CPA security. Our first attack, the public-key-only attack, recovers each secret ideal by generating samples from the target ideal using only the public key. We further develop an optimized public-key-only attack, which identifies two suitable generators through algebraic-norm filtering and thereby reduces the computational cost of ideal reconstruction. Our second attack, the public-parameter-only attack, targets the practical instantiation and directly recovers the plaintext by deriving the hidden parameters from the public plaintext moduli and constructing an evaluation map that simulates the decryption map.

We implemented all three attack algorithms in SageMath and experimentally confirmed that they successfully break the scheme. The optimized public-key-only attack scales to larger parameters than the basic variant, while the public-parameter-only attack remains highly efficient, recovering plaintexts within seconds.

Author Contributions

This work was done during H.S.'s research internship at NTT. M.T. supervised the project and provided the initial idea that led to the first attack. Hyewon Sung developed the second attack and carried out the verification of the attack ideas, experimental evaluation, analysis, and overall preparation of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding

There is no funding information to report.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The experimental code is publicly available at https://github.com/HyewonSung/ACISP2026_submission.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used Claude Code to assist with the generation of experimental code. The generated code was subsequently reviewed and verified by the authors. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Appendix A. Security Notion

We recall the standard security notion for public key encryption schemes. For a scheme to be considered secure against passive adversaries, it must satisfy Indistinguishability under Chosen Plaintext Attack (IND-CPA).

Definition A1 (IND-CPA Security)**.** *An encryption scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is IND-CPA secure if for any probabilistic polynomial-time adversary* $\mathcal{A}$*, the advantage defined below is negligible in the security parameter* λ*:*

$$\mathbf{Adv}^{\textit{IND-CPA}}_{\Pi,\mathcal{A}}(\lambda) = \left| \Pr\left[b' = b \;\middle|\; \begin{array}{l} (pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda), \\ (m_0, m_1) \leftarrow \mathcal{A}(pk), \\ b \leftarrow \{0,1\}, \quad c_b \leftarrow \mathsf{Enc}(pk, m_b), \\ b' \leftarrow \mathcal{A}(pk, c_b) \end{array} \right] - \frac{1}{2} \right|.$$

Remark A1. *FHE schemes include an additional algorithm* Eval *to support computations on ciphertexts, but the underlying encryption layer must satisfy IND-CPA. This property is the minimal security requirement for plaintext privacy.*

Remark A2. *While IND-CPA is the standard security requirement for guaranteeing plaintext privacy, there is also a strictly weaker, yet fundamental security notion known as Unbreakability under Chosen Plaintext Attack (UBK-CPA). UBK-CPA merely requires that it is computationally infeasible for* $\mathcal{A}$ *to recover the secret key. If a scheme fails to satisfy UBK-CPA,* $\mathcal{A}$ *can completely recover the secret key and trivially decrypt any ciphertext, thereby immediately violating IND-CPA as well.*

Appendix B. Supplementary Lemmas and Proofs

Lemma A1. *Let* p *be an odd prime and set* $n = p - 1$*. Let* $\phi_p(x) = x^{p-1} + \cdots + x + 1 \in \mathbb{Z}[x]$*, and consider the ring of integers* $\mathbb{Z}[\zeta_p] \cong \mathbb{Z}[x]/\langle \phi_p(x) \rangle$*. Fix a prime* $q \neq p$ *and define the principal ideal*

$$I_q = \langle \zeta_p^{-1} + q \rangle \subset \mathbb{Z}[\zeta_p].$$

Let $t(I_q)$ denote the one-dimensional modulus of I_q. Then,

$$t(I_q) = q^n - q^{n-1} + \cdots - q + 1.$$

Equivalently, since $n = p - 1$, we have

$$t(I_q) = \frac{q^p + 1}{q + 1}.$$

Proof. Note that the one-dimensional modulus $t(I_q)$ is equivalent to the ideal norm of I_q. Since $I_q = \langle \zeta_p^{-1} + q \rangle$ is a principal ideal, its norm is simply the absolute algebraic norm of its generator. Therefore, it suffices to evaluate $t(I_q) = |N(\zeta_p^{-1} + q)|$.

Note that $\zeta_p^{n-1} = \zeta_p^{p-2} = \zeta_p^{-1}$. Since ζ_p is a unit in $\mathbb{Z}[\zeta_p]$, multiplying the generator by ζ_p does not change the ideal. Thus, we can rewrite the ideal as:

$$I_q = \langle \zeta_p^{-1} + q \rangle = \langle \zeta_p(\zeta_p^{-1} + q) \rangle = \langle 1 + q\zeta_p \rangle.$$

The algebraic norm of $1 + q\zeta_p$ is the product of its Galois conjugates:

$$N(1 + q\zeta_p) = \prod_{k=1}^{p-1} (1 + q\zeta_p^k).$$

To evaluate this product, we use the standard factorization $x^p - 1 = \prod_{k=0}^{p-1}(x - \zeta_p^k)$. Substituting $x = -q$ yields:

$$(-q)^p - 1 = \prod_{k=0}^{p-1}(-q - \zeta_p^k).$$

Since p is an odd prime, $(-q)^p - 1 = -(q^p + 1)$. Factoring out -1 from the right side gives:

$$-(q^p + 1) = \prod_{k=0}^{p-1} -(q + \zeta_p^k) = -(q + 1) \prod_{k=1}^{p-1} (q + \zeta_p^k).$$

Dividing both sides by $-(q + 1)$ provides:

$$\prod_{k=1}^{p-1} (q + \zeta_p^k) = \frac{q^p + 1}{q + 1}.$$

We can rewrite the terms in the product as $q + \zeta_p^k = \zeta_p^k(1 + q\zeta_p^{-k})$. Taking the product over $1 \le k \le p - 1$, the coefficient is $\prod_{k=1}^{p-1} \zeta_p^k = \zeta_p^{p(p-1)/2}$, which equals 1 because p is an odd prime. Furthermore, as k ranges from 1 to $p - 1$, the set of inverses $\{\zeta_p^{-k}\}$ is simply a permutation of $\{\zeta_p^k\}$. Therefore:

$$\prod_{k=1}^{p-1} (q + \zeta_p^k) = \prod_{k=1}^{p-1} (1 + q\zeta_p^k) = N(1 + q\zeta_p).$$

Finally, expanding the rational expression $\frac{q^p+1}{q+1}$ as a geometric series for odd p yields:

$$t(I_q) = |N(1 + q\zeta_p)| = \frac{q^p + 1}{q + 1} = q^{p-1} - q^{p-2} + \cdots - q + 1.$$

Substituting $n = p - 1$ concludes the proof. □

Lemma A2. *Let p be an odd prime and q_i be a prime such that $q_i \neq p$. Let $t_i = (q_i^p + 1)/(q_i + 1)$ be the corresponding public one-dimensional modulus. Define $a_i \equiv -q_i^{-1} \pmod{t_i}$. Then, the evaluation map*

$$\psi_i : \mathbb{Z}[x]/\langle \phi_p(x) \rangle \to \mathbb{Z}_{t_i}, \qquad \psi_i(f(x)) = f(a_i) \bmod t_i$$

is a well-defined ring homomorphism. Furthermore, the secret ideal $I_i = \langle 1 + q_i x \rangle$ *is contained in the kernel of* ψ_i.

Proof. First, we prove that the map ψ_i is well-defined over the quotient ring $\mathbb{Z}[x]/\langle \phi_p(x) \rangle$. This requires showing that the defining polynomial $\phi_p(x)$ evaluates to zero at a_i modulo t_i. Note that $t_i = q_i^{p-1} - q_i^{p-2} + \cdots + 1 \equiv 1 \pmod{q_i}$, which ensures $\gcd(q_i, t_i) = 1$, making $a_i \equiv -q_i^{-1} \pmod{t_i}$ well-defined. Using the definition of the cyclotomic polynomial $\phi_p(x) = x^{p-1} + x^{p-2} + \cdots + 1$ and substituting $x = a_i$, we have

$$\phi_p(a_i) \equiv (-q_i^{-1})^{p-1} + (-q_i^{-1})^{p-2} + \cdots + (-q_i^{-1}) + 1 \pmod{t_i}.$$

Since p is an odd prime, $p - 1$ is even. We can factor out $q_i^{-(p-1)}$ from the expression,

$$\phi_p(a_i) \equiv q_i^{-(p-1)} \left(1 - q_i + q_i^2 - \cdots - q_i^{p-2} + q_i^{p-1}\right) \pmod{t_i}.$$

Notice that the term in the parentheses is exactly t_i as established in Lemma A1. Therefore,

$$\phi_p(a_i) \equiv q_i^{-(p-1)} \cdot t_i \equiv 0 \pmod{t_i}.$$

Thus, ψ_i respects the quotient relation $\phi_p(x) = 0$ and is a well-defined ring homomorphism.

Second, we show that $\psi_i(I_i) = 0$. The ideal I_i is generated by $1 + q_i x$ in the quotient ring. Evaluating this generator under ψ_i yields:

$$\psi_i(1 + q_i x) = 1 + q_i a_i \equiv 1 + q_i(-q_i^{-1}) \equiv 1 - 1 \equiv 0 \pmod{t_i}.$$

Because the generator of I_i evaluates to zero, every element in the ideal I_i maps to zero under the homomorphism ψ_i. This completes the proof. □

Appendix C. Flaws of Security Reductions in [22]

In [22], Construction 1 is claimed to be secure under worst-case hardness assumptions for lattice problems. The authors interpret the problem of recovering the plaintext from a ciphertext as solving a worst-case lattice problem. Recall that the encryption algorithm produces a ciphertext

$$c \;=\; \sum_{i=1}^{m} \bar{u}_i \, A'_i,$$

where the public keys A'_i are known ring elements. Given $(A'_1, \ldots, A'_m, c)$, recovering plaintext components $(u_1, \ldots, u_m)$ is formulated as an instance of the generalized compact knapsack problem over $(\mathbb{Z}^n, +, \cdot)$.

To establish a worst-case hardness guarantee, the authors extend Micciancio's result on cyclic lattices [29] to general ideal lattices. Specifically, they claim that solving this knapsack problem over $(\mathbb{Z}^n, +, \cdot)$ is at least as hard as solving the covering radius problem for any n-dimensional full rank cyclic lattice. Furthermore, they argue that for sufficiently large modulus q, the underlying search task can be viewed as an inhomogeneous Ring Short Integer Solution (RSIS) problem. Consequently, finding a short solution to the modular linear relation is reduced to standard worst-case lattice assumptions, such as the hardness of decisional Gap Shortest Vector Problem (GapSVP).

However, the above discussion can also not be justified for Construction 1. A worst-case reduction must show that solving an arbitrary instance of a standard lattice problem, for example like GapSVP or (Ring-)SIS, can be reduced to breaking the scheme. In contrast, the instances that come from the encryption equation are highly structured. The public elements $(A'_1, \cdots, A'_m)$ are not generic ring samples, but are constructed to satisfy the CRT congruences. This extra algebraic structure is precisely what enables the attack in this paper and is absent from the standard worst-case formulations.

Moreover, the claimed connection to inhomogeneous Ring-SIS relies on viewing the recovery of bounded unknowns from a modular relation as an SIS-type task for a sufficiently large modulus. Yet, in this scheme the relevant relation is induced by a fixed, publicly constructed family with some specific properties, and encryption introduces no fresh randomness once the public key is fixed. As a result, the resulting instances differ fundamentally from the random instance distributions for which worst-case-to-average-case reductions for (Ring-)SIS are known. Therefore, the security of Construction 1 cannot be substantiated solely by appealing to worst-case lattice problems.

References

1. Rivest, R.L.; Adleman, L.; Dertouzos, M.L. On Data Banks and Privacy Homomorphisms. In *Foundations of Secure Computation*; Academic Press: New York, NY, USA, 1978; pp. 165–179.
2. Rabin, M.O. *Digitalized Signatures and Public-Key Functions as Intractable as Factorization*; Report MIT/LCS/TR-212; Massachusetts Institute of Technology, Laboratory for Computer Science: Cambridge, MA, USA, 1979.
3. Paillier, P. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology — EUROCRYPT'99, Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 223–238. https://doi.org/10.1007/3-540-48910-X_16.
4. Boneh, D.; Goh, E.J.; Nissim, K. Evaluating 2-DNF Formulas on Ciphertexts. In *Theory of Cryptography, Proceedings of the Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, 10–12 February 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 325–341.
5. Cocks, C. An Identity Based Encryption Scheme Based on Quadratic Residues. In *Cryptography and Coding, Proceedings of the 8th IMA International Conference, Cirencester, UK, 17–19 December 2001*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 360–363. https://doi.org/10.1007/3-540-45325-3_32.
6. Gentry, C. Fully Homomorphic Encryption Using Ideal Lattices. In Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, Bethesda, MD, USA, 31 May–2 June 2009; pp. 169–178. https://doi.org/10.1145/1536414.1536440.
7. Brakerski, Z.; Gentry, C.; Vaikuntanathan, V. (Leveled) Fully Homomorphic Encryption without Bootstrapping. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, Cambridge, MA, USA, 8–10 January 2012; pp. 309–325. https://doi.org/10.1145/2090236.2090262.
8. Fan, J.; Vercauteren, F. Somewhat Practical Fully Homomorphic Encryption. 2012. Available online: https://eprint.iacr.org/2012/144 (accessed on 13 August 2026).
9. Cheon, J.H.; Kim, A.; Kim, M.; et al. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology—ASIACRYPT 2017, Proceedings of the 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, 3–7 December 2017*; Springer: Cham, Switzerland, 2017; Volume 10624, pp. 409–437. https://doi.org/10.1007/978-3-319-70694-8_15.
10. Chillotti, I.; Gama, N.; Georgieva, M.; et al. Faster Fully Homomorphic Encryption: Bootstrapping in Less than 0.1 Seconds. In *Advances in Cryptology—ASIACRYPT 2016, Proceedings of the 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, 4–8 December 2016*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 3–33. https://doi.org/10.1007/978-3-662-53887-6_1.
11. Gentry, C.; Sahai, A.; Waters, B. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. In *Advances in Cryptology—CRYPTO 2013, Proceedings of the 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 2013*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8042, pp. 75–92. https://doi.org/10.1007/978-3-642-40041-4_5.
12. Ducas, L.; Micciancio, D. FHEW: Bootstrapping Homomorphic Encryption in Less than a Second. In *Advances in Cryptology—EUROCRYPT 2015, Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, 26–30 April 2015*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9056, pp. 617–640. https://doi.org/10.1007/978-3-662-46800-5_24.
13. Kipnis, A.; Hibshoosh, E. Efficient Methods for Practical Fully Homomorphic Symmetric-Key Encryption, Randomization and Verification. 2012/637. Available online: https://eprint.iacr.org/2012/637 (accessed on 13 August 2026).
14. Tsaban, B.; Lifshitz, N. Cryptanalysis of the MORE Symmetric Key Fully Homomorphic Encryption Scheme. *J. Math. Cryptol.* **2015**, *9*, 75–78. https://doi.org/10.1515/jmc-2014-0013.
15. Vizár, D.; Vaudenay, S. Cryptanalysis of Enhanced More. *Tatra Mt. Math. Publ.* **2019**, *73*, 163–178. https://doi.org/10.2478/tmmp-2019-0012.
16. Alex, S.; Yang, B. Cryptanalysis and Modification of a Variant of Matrix Operation for Randomization or Encryption (V-MORE). In *Data and Applications Security and Privacy XXXIX, Proceedings of the 39th IFIP WG 11.3 Annual Conference on Data and Applications Security and Privacy, DBSec 2025, Gjøvik, Norway, 23–24 June 2025*; Springer Nature: Cham, Switzerland, 2025; pp. 439–455.
17. Li, J.; Wang, L. Noise-Free Symmetric Fully Homomorphic Encryption Based on Non-Commutative Rings. 2015/641. Available online: https://eprint.iacr.org/2015/641 (accessed on 13 August 2026).
18. Barenghi, A.; Mainardi, N.; Pelosi, G. Comparison-Based Attacks against Noise-Free Fully Homomorphic Encryption Schemes. In *Information and Communications Security, Proceedings of the 20th International Conference, ICICS 2018, Lille, France, 29–31 October 2018*; Springer: Cham, Switzerland, 2018; pp. 177–191. https://doi.org/10.1007/978-3-030-01950-1_11.
19. Yagisawa, M. Fully Homomorphic Encryption without Bootstrapping. 2015/474. Available online: https://eprint.iacr.org/2015/474 (accessed on 13 August 2026).
20. Wang, Y. Octonion Algebra and Noise-Free Fully Homomorphic Encryption (FHE) Schemes. *ArXiv* **2016**, arXiv:1601.06744.
21. Nuida, K. Towards Constructing Fully Homomorphic Encryption without Ciphertext Noise from Group Theory. In

International Symposium on Mathematics, Quantum Theory, and Cryptography; Springer: Singapore, 2021; pp. 57–78.

22. Zheng, Z.; Liu, F.; Tian, K. An Unbounded Fully Homomorphic Encryption Scheme Based on Ideal Lattices and Chinese Remainder Theorem. *J. Inf. Secur.* **2023**, *14*, 366–395. https://doi.org/10.4236/jis.2023.144021.
23. Chun, J.; Han, H.; Kang, S.V.; et al. Noiseless Homomorphic Encryption for Complex Numbers and Fully Homomorphic Encryption with Modular Arithmetic. *J. Algebra Its Appl.* **2026**, *25*, 2650060. https://doi.org/10.1142/s021949882650060x.
24. Chun, J.; Han, H.; Kang, S.V.; et al. Construction of Noiseless Leveled Fully Homomorphic Encryption Scheme. *J. Algebra Its Appl.* **2026**, *25*, 2650183. https://doi.org/10.1142/s0219498826501835.
25. van Dijk, M.; Gentry, C.; Halevi, S.; et al. Fully Homomorphic Encryption over the Integers. In *Advances in Cryptology—EUROCRYPT 2010, Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, 30 May–3 June 2010*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 24–43. https://doi.org/10.1007/978-3-642-13190-5_2.
26. Nuida, K.; Kurosawa, K. (Batch) Fully Homomorphic Encryption over Integers for Non-Binary Message Spaces. In *Advances in Cryptology—EUROCRYPT 2015, Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, 26–30 April 2015*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 537–555. https://doi.org/10.1007/978-3-662-46800-5_21.
27. Kim, E.; Tibouchi, M. FHE over the Integers and Modular Arithmetic Circuits. *IET Inf. Secur.* **2018**, *12*, 257–264. https://doi.org/10.1049/iet-ifs.2017.0024.
28. Koseki, R.; Ito, A.; Ueno, R.; et al. Homomorphic Encryption for Stochastic Computing. *J. Cryptogr. Eng.* **2023**, *13*, 251–263. https://doi.org/10.1007/s13389-022-00299-6.
29. Micciancio, D. Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions from Worst-Case Complexity Assumptions. In Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, 'Vancouver, BC, Canada, 16–19 November 2002; pp. 356–365.