Key-Dependent Message Secure (Multi-Key) Fully Homomorphic Encryption Based on a Novel Hardness Assumption

Liang Zhou, Ruwei Huang, and Sai Hu

[[1]](#footnote-1) ***Abstract*—Fully Homomorphic Encryption (FHE) is a cryptosystem that supports arbitrary computation on encrypted data, making it highly valuable in the current cloud computing environment. In STOC 2012, López-Alt et al. proposed Multi-Key Fully Homomorphic Encryption (MKFHE), enabling homomorphic operations on ciphertexts encrypted with distinct keys. However, most existing FHE and MKFHE schemes achieve only IND-CPA security and rely on additional security assumptions such as circular security, which limits their security guarantees and impacts parameter settings and performance. In this paper, we introduce a novel hardness assumption, the Augmented Invertible Ring Learning with Errors (AIRLWE), improving upon the standard RLWE assumption. Based on AIRLWE, we present Key-Dependent Message (KDM)-secure Brakerski-Gentry-Vaikuntanathan (BGV) type and ring-based Gentry-Sahai-Waters (GSW) type single-key FHE schemes, with the latter supporting ring element plaintext. KDM security guarantees that encryption remains secure even when the messages being encrypted are functions of the secret keys themselves. We then develop a BGV type MKFHE scheme that upholds the same level of KDM security, illustrating the use of hybrid homomorphic multiplication to produce the necessary evaluation keys without imposing constraints on the coefficients of the secret keys. Additionally, we introduce the concept of “dynamic exit” for MKFHE schemes, explaining its straightforward implementation in our framework. By ensuring circular security in the evaluation key generation and key switching processes, our scheme achieves notable performance improvements compared to existing solutions.**

***Index Terms*—Multi-key fully homomorphic encryption, key-dependent message security, ring learning with errors, dynamic exit.**

# I. INTRODUCTION

F

ULLY homomorphic encryption (FHE) is a powerful cryptographic primitive that enables computations on encrypted data without revealing the underlying plaintext, thus providing strong privacy guarantees. This capability has significant implications for secure data processing in various domains, including cloud computing, secure multi-party computation (MPC), and privacy-preserving data analysis [1], [2], [3], [4], [5], [6], [7], [8], [9], [10]. However, traditional FHE schemes restrict computations to ciphertexts that are decryptable with a single key, which does not accommodate secure computation scenarios with various data providers, each with distinct keys.

Multi-key full homomorphic encryption (MKFHE) builds on the FHE concept by enabling operations on ciphertexts encrypted with distinct and separate keys. This extension is crucial for applications like on-the-fly MPC, where multiple parties need to compute jointly on their private inputs without prior coordination. In 2012, López-Alt et al. [11] presented the initial MKFHE scheme utilizing the NTRU cryptosystem [12]. This scheme, however, depended on a non-standard security assumption concerning polynomial rings.

Subsequent research has focused on developing MKFHE schemes based on the Learning with Errors (LWE) assumption [13] and the Ring Learning with Errors (RLWE) assumption [14], which are considered more standard and well-studied hardness assumptions in cryptography. Clear and McGoldrick [15] developed an MKFHE scheme grounded in the LWE problem, enabling the use of numerous secret keys through the Gentry-Sahai-Waters (GSW) FHE scheme [8], [16], [17]. Mukherjee and Wichs [18] later refined this approach and presented a threshold decryption protocol, further enhancing the practicality of MKFHE schemes.

Despite these advancements, the aforementioned MKFHE schemes have several limitations. Both the Clear and McGoldrick scheme [15] and the Mukherjee and Wichs scheme [18] require all parties to be predetermined prior to the start of homomorphic computation, which means new parties cannot join once the computation is underway. This restriction is known as the single-hop limitation in MKFHE [19]. Peikert and Shiehian [19] devised a multi-hop MKFHE scheme that addresses this issue by allowing dynamic addition of parties during the computation, albeit resulting in significantly increased ciphertext size. Brakerski and Perlman [20] introduced a fully dynamic MKFHE scheme that does not require a predetermined number of users during setup and achieves linear growth of ciphertext size proportional to the number of involved keys. However, their approach necessitates the use of extended refresh keys to bootstrap ciphertexts for allowing new users to join dynamically, resulting in comparatively inefficient performance due to the generally high cost of bootstrapping. Chen et al. [21] extend [22] by employing special GSW ciphertexts to create an efficient ciphertext extension algorithm, enabling effective extension of evaluation keys and linear growth of ciphertext length with the number of users.

## A. Related Work

In TCC 2017, Chen et al. [23] introduced the initial Brakerski-Gentry-Vaikuntanathan (BGV) [4] type multi-hop MKFHE scheme. This approach employs a modified version of the RLWE based GSW scheme, where plaintexts are treated as ring elements, to generate evaluation keys required for key switching. This scheme retains the advantageous features of BGV, supports ciphertext packing through the Chinese Remainder Theorem (CRT), and is applicable for developing two-round MPC protocols as well as threshold decryption protocols. In 2019, Li et al. [24] introduced a BGV type MKFHE scheme that reduces ciphertext size by nearly half compared to [23]. Additionally, they apply hybrid homomorphic multiplication between RLWE-based BGV and RLWE-based GSW ciphertexts to generate evaluation keys, significantly reducing the dimensions of public parameters and evaluation keys. Chen et al. [25] later optimized the relinearization process in BGV type MKFHE, creating an efficient scheme for privacy-preserving neural network computations. This work includes both BFV [6] type and CKKS [9] type MKFHE schemes. Kim et al. [26] further optimized [25] by redesigning the multi-key multiplication algorithm, achieving optimal complexity with linear growth in the number of keys. They introduced homomorphic gadget decomposition to enhance performance.

However, it should be emphasized that all the mainstream homomorphic encryption schemes mentioned above are limited to Indistinguishability under Chosen Plaintext Attack (IND-CPA) security. This means they cannot reliably guarantee privacy when encrypting information associated with any user’s secret key within the cryptosystem. Additionally, they often depend on extra security assumptions, like circular security, to implement essential techniques like bootstrapping and relinearization, which involve using the users’ public key to encrypt their own secret key. BGV type leveled fully homomorphic encryption (LFHE) schemes, such as [4], [23], and [24], avoid potential circular security issues during key switching by replacing all users’ secret keys at each layer of the Boolean circuit. This method, while ensuring rigorous security, leads to increased parameter sizes and degraded performance.

Key-dependent message (KDM) secure encryption schemes ensure confidentiality even if messages pertain to the secret key . This means that even when messages of the form , where is chosen from some function family , are encrypted, the scheme remains secure. KDM security is crucial in applications like hard disk encryption, encrypted key backups, formal proofs [27], [28], homomorphic encryption, and advanced cryptographic protocols [29], [30], [31], [32]. In 1984, Goldwasser and Micali [60] implicitly hinted at the risks of encrypting secret keys with cryptosystems. Camenisch and Lysyanskaya [29] presented the concept of circular security and designed a public key encryption scheme that is provably circular-secure in the random oracle model, allowing each user to securely encrypt their secret keys with one of their public keys. Black et al. [34] argued that merely semantically secure encryption is inadequate for messages linked to the underlying secret key and proposed KDM security as a stronger security notion. They observed that while KDM security implies CPA security, the reverse is not true, and in the asymmetric model, circular security is notably weaker than KDM security. Subsequently, numerous works [33], [35], [36], [37], [38], [39], [40], [41] analyzed this problem under various number-theoretic assumptions, demonstrating that even CPA-secure public-key bit-encryption schemes cannot directly satisfy circular security. Alamati and Peikert [42] further constructed specific counterexamples showing that encryption schemes secure against Chosen Ciphertext Attack (CCA) under the LWE and RLWE assumptions do not necessarily imply circular security.

Relying on the Decisional Diffie-Hellman (DDH) assumption, Boneh et al. [43] proposed the pioneering public-key encryption scheme achieving -key KDM security for affine transformations of the secret key within the standard model. Applebaum et al. [44] showed how to transform standard hard problems into their Hermite Normal Form (HNF) and achieved -key KDM security for affine functions in lattice-based cryptosystems. They proposed a public-key encryption scheme grounded in the LWE assumption and a symmetric encryption scheme relying on the Learning Parity with Noise (LPN) assumption, both featuring compact ciphertexts and post-quantum security. Barak et al. [45] were pioneers in achieving KDM security for bounded circuit size functions in the standard model, using standard assumptions like DDH or LWE. They also observed that given specific non-standard hardness assumptions, any fully homomorphic encryption scheme that possesses both circular security and circuit privacy can attain full KDM security. Applebaum [46] constructed an encryption scheme with -key KDM security for projection functions and proved the inaugural general KDM amplification theorem, relying exclusively on the base scheme’s KDM security and not requiring additional assumptions.

In CRYPTO 2011, Brakerski and Vaikuntanathan [47] proposed the initial KDM-secure somewhat homomorphic encryption (SWHE) scheme, leveraging the simplified RLWE assumption, capable of maintaining security when encrypting polynomial expressions involving the secret key. However, their scheme required that the dimensions of the ciphertexts and secret key vectors grow linearly with the degree of the polynomial expressions involving the secret key being encrypted. Despite a series of subsequent works [32], [48], [49], [50], [51] exploring methods to enhance the KDM security of encryption systems, no clear techniques have yet been identified that can be applied to fully homomorphic encryption schemes.

## B. Our Contributions

In this work, we expand the challenge function set from [47] and propose both a KDM-secure BGV type single-key scheme and a KDM-secure ring-based GSW type single-key scheme. Utilizing these two basic schemes, we further develop a BGV type multi-hop MKFHE scheme with equivalent KDM security. As far as we know, this is the first study since [47] to explicitly provide KDM-secure FHE schemes.

Firstly, we improve the standard RLWE problem by proposing a new variant called the Augmented Invertible Ring Learning with Errors (AIRLWE) problem and introducing a decisional assumption that is more natural for cryptographic applications. We ensure its reliability by proving that the difficulty of the AIRLWE problem can ultimately be reduced to that of the RLWE problem. We also present an intuitive reduction from the standard AIRLWE problem to its Hermite normal form (HNF) [52], allowing the secret to be selected from the error distribution. Additionally, we apply the noise scaling technique. These modifications do not affect its hardness.

Next, based on the AIRLWE assumption, we construct our two single-key basic schemes. Using the BGV type scheme as an example, we provide a detailed analysis of their correctness and security. Despite KDM security being widely recognized as a stronger security notion, we rigorously prove both the semantic security and KDM security of our schemes to ensure the thoroughness of our work. In contrast to [47], which lacked a strict proof of KDM security for their scheme, we adhere to the standard KDM security definition [34] and employ a hybrid argument [53] to demonstrate that for any adversary running in probabilistic polynomial time (PPT), encryptions of key-dependent messages within the challenge function set are computationally indistinguishable from encryptions of zero, thus reliably proving the KDM security of our schemes.

Then, we combine these two single-key basic schemes to propose a BGV type multi-hop MKFHE scheme, utilizing the GSW type basic scheme for generating evaluation keys. Building on the BGV type ciphertext extension method proposed by [23] and optimized by [24], our MKFHE scheme maintains KDM security and potentially offers more interesting applications. We show that our BGV type single-key and multi-key schemes feature circular-secure evaluation key generation and key switching processes, significantly optimizing the parameters and performance of both schemes. In our BGV type schemes, it is no longer necessary to prepare different secret keys for each user at every circuit level. Assuming a circuit depth of for homomorphic computation, we reduce the quantity of secret keys required per user from to 1, greatly optimizing the secret key size.

Finally, from a practical application perspective, we construct a distributed decryption protocol suitable for our MKFHE scheme. Inspired by [19] and [20], we propose the concept of “dynamic exit” for MKFHE schemes, allowing several participants to dynamically exit the homomorphic computation at any time. Participants who dynamically exit the homomorphic computation can leave their associated data and intermediate results in ciphertext form for the remaining participants to continue using in the joint computation. Once a participant exits the homomorphic computation, they are no longer required to be involved in any subsequent activities of the MKFHE scheme. We demonstrate how to easily implement dynamic exits for participants in our MKFHE scheme.

# II. Preliminaries

## A. Basic Notation

Throughout this paper, scalars appear as plain letters, like . Vectors are denoted by lowercase boldface letters, for example, , while matrices are expressed by bold capitals, such as . The -th entry of vector appears as , while refers to the entry located at the intersection of the -th row and -th column of matrix . To enhance clarity, the notation is sometimes employed to indicate the inner product of vectors and .

We define as the security parameter and as a function of that is negligible. Let be the -th cyclotomic polynomial, having a degree equal to where represents Euler’s function. We define the polynomial rings and for a prime integer . Operations within these rings, including addition and multiplication, are executed on their coefficients in a component-wise manner. The notation ​ signifies that the coefficients of are reduced to the interval (except for ).

*Definition 1 (Negligible Function):*A function is considered negligible, denoted as , if for each positive polynomial and large enough, the inequality holds. This implies that as increases, approaches zero faster than for any , where is a constant. In addition, a probability described by a negligible function is considered negligible. Conversely, a probability expressed as is termed overwhelming.

Let be a -bounded error distribution over , with coefficients within the range . The error distribution is specifically the discrete Gaussian distribution for some standard deviation . Drawing from this distribution results in a noise polynomial like . This discrete Gaussian distribution is -dimensional and employs the standard isomorphism that connects degree polynomials in to vectors in . This isomorphism allows these vectors to be interpreted as coefficient vectors of terms derived from the ring . We emphasize the use of spherical Gaussian distributions with consistent standard deviation across all dimensions.

For a vector , the norms are described as follows: the -norm captures the maximum absolute value among the components; the ​-norm sums the absolute values; the -norm represents the Euclidean length. For a polynomial sampled from ring , the norm corresponds to the norm of its coefficient vector. The -norm of is also referred to as the length of the polynomial.

## B. Ring Learning with Errors (RLWE) Problem

The RLWE problem operates over a ring , where is a power of two, and defines as the quotient ring modulo , with being a modulus associated with the security parameter. The task involves differentiating between samples derived from two distributions: In the first distribution, denoted as distribution , samples where each pair is generated from the equation , with uniformly sampled from , a secret uniformly sampled from , and a noise term sampled from an error distribution over . In the second distribution, samples are uniformly random pairs from . The RLWE assumption posits that it is computationally infeasible for any efficient algorithm to differentiate between these two distributions beyond a negligible advantage.

## C. Two Useful Subroutines

In FHE schemes, and are commonly employed, especially for managing the noise growth in operations such as ciphertext multiplication and key switching process. When the noise primarily depends on a specific vector , the vector can be expanded into binary form to diminish the noise magnitude. This involves decomposing each element of the vector into its binary representation, thereby effectively reducing the -norm of the vector from to , which substantially reduces the noise.

: For and the modulus , let such that , output .

After a vector is expanded into its bit representation, to ensure that the dot product between two vectors remains invariant, the other vector must be transformed as follows:

: For and the modulus , output .

It can be easily proved that for all and vectors of equal length, the following property holds:

.

## D. Modulus Switching

Modulus switching is a technique utilized in FHE schemes to reduce the noise term in ciphertexts after homomorphic computations. By changing the inner modulus of ciphertexts from a larger to a smaller , the noise is diminished approximately by the ratio while maintaining decryption correctness with the identical secret key.

: On input a ciphertext encrypted for modulus , the operation outputs , adjusted to the smaller modulus ​. The output is given by: .

*Lemma 1 [4]:* Suppose is a ciphertext encrypted with the key modulo and noise , where has length at most , and let message where is the plaintext modulus. Let be the output of and . Consequently, the new noise is at most in length, where is the expansion factor of . Assuming this new noise is less than , it follows that .

## E. Key Switching

The technique of key switching permits transforming a ciphertext encrypted with one secret key to be decryptable under a different secret key, without decrypting or altering the underlying message. Additionally, key switching is particularly used in FHE schemes to decrease the dimension of ciphertexts expanded following homomorphic operations, resizing them to a normal level that ensures efficiency in subsequent computations. Define , the key switching approach includes two steps:

: Generates some auxiliary information , commonly known as the evaluation key, that enables the switching. This function computes and produces .

: On input the evaluation key and ciphertext under the secret key , outputs a new ciphertext under the secret key .

The evaluation key fundamentally involves encrypting portions of the secret key (formatted in a specific way) using the secret key . For key switching to be correct, it is necessary that for a small norm . Furthermore, we emphasize that the use of key switching technology necessitates that all existing leveled FHE schemes either prepare different secret keys for each user at every level of the circuit or just assume that the scheme is circular-secure to avoid potential circular security issues. Lastly, the key switching computation involves multiplying a high-dimensional vector by a high-dimensional matrix, resulting in a new ciphertext whose noise level is slightly higher than that of the original ciphertext.

*Lemma 2 [4]:*Suppose is a ciphertext encrypted with the key modulo and noise , where has length at most , and let message where is the plaintext modulus. Let be the output of and . Consequently, the new noise is at most in length, where is the expansion factor of and is an upper bound indicating that ring elements sampled from the error distribution do not exceed this length. Assuming this new noise is less than , it follows that .

## F. Some Useful Lemmas

This section presents key lemmas and a corollary that support our work, including both prior results and new ones proven in this paper.

*Lemma 3 [54]:* Given , any real such that , and any vector with , the maximum statistical distance between the distributions and is .

*Lemma 4 [1], [55]:* Let , , and let , . For any , , and .

*Lemma 5 [56]:* For any , and , it holds that: . Specifically, when and , we have: .

*Lemma 6 [57]:* Given , any real such that , it holds that .

*Lemma 7:* Let and be two independent random variables in , with uniformly distributed over . Then, is also uniformly distributed over .

*Proof:* Since and are independent, for any , the probability that is given by: .

Given has a uniform distribution over , it follows that:

, thus, , confirming that is uniformly distributed over .

*Corollary 1:* Assume and are two independent random variables in the polynomial ring , with uniformly distributed over and following a discrete Gaussian distribution . The sum follows a distribution that is computationally indistinguishable from the uniform distribution over , denoted as , if .

*Proof:* Using the standard isomorphism presented in Section II-A, we first map the elements and from the polynomial ring to vectors in . This representation involves expressing an element as a vector of its coefficients .

Given that follows a discrete Gaussian distribution , according to Lemma 5, the value of each component of satisfies with probability approximately . Therefore, if , then the value of each falls within with the same probability . Utilizing Lemma 7, it can be concluded that the distribution of cannot be distinguished from the uniform distribution over in a computational sense.

Finally, applying the reverse of the standard isomorphism, the result is converted back to the polynomial ring , concluding that cannot be distinguished computationally from the uniform distribution on .

## G. KDM Security Definition

*Definition 2 (KDM Security [34]):* Consider a public-key encryption scheme that meets the standard correctness criteria. Let be defined as some polynomial, and let represent some family of functions. The scheme is termed as -KDM secure if for all PPT adversary , the following holds:

where refers to the output generated by in the game described below:

1) For each , the challenger generates and supplies with .

2) The adversary submits an unlimited number of KDM encryption queries:

1. chooses a pair with ranging over and drawn from .

2. The challenger returns , with the message specified by:

If , then set , in which represents the output length of .

If , then set .

The adversary is permitted by default to select constant-0 or constant-1 functions, resulting in encrypting 0 or 1, even if these functions are not formally included within the class .

# III. Variants of the RLWE Problem

This section aims to enhance the KDM security of lattice-based FHE schemes by improving the standard RLWE problem and introducing a new variant called the Augmented Invertible Ring Learning with Errors (AIRLWE) problem. We demonstrate the decisional assumption inherent in the standard AIRLWE problem along with its hardness reduction. Furthermore, we provide a compact reduction to its Hermite Normal Form (HNF). This reduction permits choosing secret keys within the error distribution when constructing a cryptosystem. Additionally, we employ the noise scaling technique to generate more practical AIRLWE samples.

## A. The Augmented Invertible Ring Learning with Errors (AIRLWE) Problem

In [58], Stehlé et al. introduced a modification to the RLWE problem by uniformly sampling the ring elements from , where represents the set of invertible elements in . This variation maintains the other parameters consistent with the standard RLWE problem, resulting in the distribution consisting of sample pairs . Hence, they introduced an invertible variant of RLWE, denoted as RLWE🞨. It is noted that when , the probability that a uniformly sampled element from is invertible becomes overwhelming. Consequently, the RLWE problem retains its hardness even when and the uniform distribution are replaced respectively by and , ensuring the difficulty of the RLWE🞨 problem. Inspired by the work in [58], we present a novel variant of the RLWE problem designed to enhance the KDM security of FHE schemes, termed the Augmented Invertible Ring Learning with Errors (AIRLWE) problem. The AIRLWE problem is formally defined as follows:

*Definition 3 (Augmented Invertible Ring Learning with Errors (AIRLWE) Problem):*Let the ring , where is a power of two, and defines as the quotient ring modulo , with being a modulus related to the security parameter . Set as an error distribution over , and denote as the set of invertible elements within . The AIRLWE problem involves distinguishing between two distributions: Within the first distribution, denoted as distribution , samples where each pair is generated from the equation , by sampling uniformly at random, and drawing , uniformly. Within the second distribution, samples are uniformly random pairs from . The AIRLWE assumption asserts the computational infeasibility for any efficient algorithm to distinguish between these two distributions with more than a negligible advantage.

*Lemma 8:*Let and be parameters for both AIRLWE and RLWE🞨. A probabilistic polynomial-time reduction can be established between the AIRLWE problem and the RLWE🞨 problem, ensuring the distribution remains computationally indistinguishable from . That is, the AIRLWE distribution and any polynomial number of elements within it are pseudorandom.

*Proof:* Consider a scenario where we randomly select a sample from a given RLWE🞨 oracle, with belonging to . Then we generate a new instance , with uniformly chosen from .

1) If the sample is taken from the uniform distribution , then within the expression , both and are uniformly sampled from . Hence, the instance is statistically indistinguishable from uniform.

2) If the sample is taken from the distribution , then according to the RLWE🞨 assumption, the instance is computationally indistinguishable from uniform. Additionally, according to Corollary 1, uniformly chosen from can effectively be transformed into the sum of an unknown secret and an unknown random noise , i.e., . Consequently, we have the expression , where is a small noise vector. Thus, the samples  from the AIRLWE distribution are computationally indistinguishable from uniform.

To summarize, in the case that the RLWE🞨 problem (thereby the RLWE problem) is infeasible, then so is the AIRLWE problem. Consequently, the distribution is computationally indistinguishable from the uniform distribution over , and arbitrary polynomial number of its elements are pseudorandom.

## B. A Generic Transformation

Similar to the approach in [47], to ensure that cryptographic schemes constructed under the AIRLWE assumption are KDM-secure, we transform AIRLWE into its Hermite Normal Form (HNF). In this variant, instead of having the secret uniformly distributed within , it is sampled from the error distribution . As explained in [14], [44], [47], this modified representation is equivalent to the original in terms of security, ensuring no loss in security strength. Furthermore, as demonstrated in [52], [59], HNF can also enhance the performance of lattice-based cryptosystems. Here, we present a straightforward and compact reduction of the standard AIRLWE problem into its HNF.

*Lemma 9:* Let serve as a prime modulus. Thus, a deterministic polynomial-time transformation exists, guaranteeing for any and error distribution , it maps the distribution to , where , and maps to itself.

*Proof:* The transformation can access a distribution on , where can be or . The transformation involves two phases:

1) Phase One: uniformly samples from distribution , obtaining a sample . When , , where .

2) Phase Two: converts the (fresh) samples from to ones that may come from a different distribution. For a sample picked from , outputs , where , , , , and inverses are computed in . Note that is invertible modulo , is uniform, so is also uniform. We now consider the two cases for :

If , since are uniform, and is pseudorandom according to the AIRLWE assumption, also follows the uniform distribution.

If , then for some , and we have , where , . Thus, is distributed according to as expected.

Building on Definition 3, we now concisely present the formal definition of the HNF for the AIRLWE assumption.

*Definition 4 (The AIRLWE Assumption - Hermite Normal Form):*The AIRLWE assumption with parameters defined under Definition 3 states that given any polynomial quantity of samples formatted as , in which each is uniformly selected from and both and are derived from the error distribution , it is computationally hard to distinguish these samples from randomly generated pairs within .

*Scaling the noise:* As explained in [47] and [4], employing the noise scaling technique to generate AIRLWE samples of the form is extremely practical in our schemes. Here, , , and are consistent with the parameters as defined in either Definition 3 or Definition 4, with being any element from that is relatively prime to . This approach not only facilitates the generation of AIRLWE samples but also ensures that these samples are small, with generation achievable in quasi-linear time. Moreover, in all our cryptographic applications, it is possible to pre-compute such samples off-line, for example, before the message to be processed is known. This variant maintains equivalence with the AIRLWE problem, ensuring that the generated variant samples remain pseudorandom and computationally indistinguishable from uniform.

*Proposition 1:*Let be relatively prime to , and let other parameters be consistent with those specified in either Definition 3 or Definition 4. Then for any , the AIRLWE assumption implies that,

,

where the ring elements are uniformly random over .

The AIRLWE problem inherits many of the outstanding features and properties of the RLWE problem. Note that as analyzed in [14] regarding the RLWE, in AIRLWE, if the chosen error parameters surpass the smoothing parameter of , the AIRLWE distribution becomes statistically indistinguishable from uniform, regardless of the secret value . Consequently, the AIRLWE problem becomes trivially impossible to solve.

# IV. Two Basic KDM-Secure Schemes

In this part, we introduce KDM-secure variants of the BGV [4] and ring-based GSW [23] schemes based on the AIRLWE assumption. A crucial point to note is that in most RLWE-based cryptosystems, such as those in [4], [23], [26], the message space is for some , while the secret key drawn from the error distribution does not always belong to the message space of the schemes. To address this issue, we replace the parameter in our schemes by a greater integer , ensuring that with overwhelming probability, . As a result, we obtain a larger message space for some integer . Additionally, we utilize the modulus switching and key switching techniques outlined within Section II-D and Section II-E.

## A. KDM-Secure BGV Type Scheme

**:** Considering the security parameter and a bound for the circuit depth, generate moduli decreasing as at each level. Additionally, select an integer that is relatively prime to each value. Define . Our operations are performed within the rings , , and , where is the set of invertible elements in . Let be a -bounded error distribution over for . Consequently, does not require adjustment across different circuit levels.

**:** Sample a ring element to be the secret key, then set the secret key vector as . For each circuit depth , generate and randomly, compute the public key , as well as the evaluation key where .

Note that since our scheme satisfies KDM security, each user only needs to generate and use one secret key throughout the entire scheme, unlike other LFHE schemes that avoid the circular security assumption, where a different secret key needs to be generated for each level of the circuit, requiring secret keys. Additionally, when , the generation of the evaluation key can be omitted.

**:** Given a message along with the public key , select random elements , compute the level- ciphertext .

**:** Given a level- ciphertext along with its corresponding secret key vector , compute and then output message .

**:** At level of the circuit, given two ciphertexts and with the same modulus and length under the same secret key vector (if necessary, apply key switching and modulus switching algorithms to achieve this).

1) Compute the ciphertext , which encrypts the sum of the two underlying messages under the secret key vector .

2) Compute under the secret key vector .

3) Compute and output .

Actually, due to our optimizations of BGV type schemes, both steps 2 and 3 can be omitted in this algorithm. We revisit and elaborate on this point in detail in Section V-A.

**:** At level of the circuit, given two ciphertexts and with the same modulus and length under the same secret key vector (if necessary, apply key switching and modulus switching algorithms to achieve this).

1) Compute their tensor product and obtain the ciphertext , which encrypts the result of multiplying the two underlying messages under the secret key vector .

2) Compute under the secret key vector .

3) Compute and output .

To show that our scheme features circular-secure evaluation key generation and key switching processes, consider the key switching parameter we utilize, which is produced using . Essentially, this parameter is , where . Here, represents the tensor of with its own, where each coefficient results from multiplying two coefficients of within .

In other words, each is a product of integer coefficients not exceeding and a term involving the secret key with a degree of at most 2. These are all included in the challenge function set of our schemes (presented and proven in Section IV-C). Consequently, encrypting these messages using the scheme remains KDM-secure. The generated evaluation key appears computationally indistinguishable from encryptions of zero and uniform.

Therefore, in our scheme, neither the evaluation key generation nor the key switching algorithms reveal any information about the secret key. Users do not need to change their secret keys as the circuit depth changes during homomorphic computations.

## B. KDM-Secure GSW Type Scheme

**:** For the security parameter , given a modulus and an integer coprime with , let . We work over rings , , and , where is the set of invertible elements in . Let be a -bounded error distribution over for .

**:** Sample a ring element as the secret key, pick random vectors and , then produce the secret key vector along with the public key .

**:** Taking a message along with the public key , sample a random element and an error matrix , and output the ciphertext:

where and denotes a identity matrix. It is observed that for some small .

**:** Two ciphertext matrices are added using regular addition within . Given two ciphertexts , the resulting ciphertext is computed as: .

**:** On input of two ciphertexts and , the algorithm first computes the bit decomposition of such that . Then, it presents the homomorphic multiplication as: .

**:** Given a ciphertext along with its corresponding secret key vector , compute and then output the message as:

where and is a identity matrix.

**:** This procedure produces the encryption of randomness employed in the GSW type ciphertext extension. Taking , draw along with two noise vectors stochastically, then generate the encrypted randomness:

where and .

It is observed that for some small .

**:** This procedure is used for ciphertext extension in the multi-key GSW scheme. Taking a ciphertext from the -th user, a ciphertext containing the randomness , along with the public keys for all participating users , produce the extended ciphertext as:

where , , , featuring the corresponding secret key vector .

*Correctness of Ciphertext Extension:* Following the correctness analysis of ciphertext extension in multi-key GSW schemes as presented in [15], [18], [23], to prove the correctness of our GSW type ciphertext extension, we need to ensure the -th row of (denoted by ) meets the following condition:

where represents a small noise vector. The analysis is detailed below:

Then we have:

where the length of is bounded by .

Finally, as a result of the correctness of the ciphertext extension, we can obtain:

where and is a small noise vector.

## C. Correctness and Security

Within this subsection, detailed proofs regarding the correctness and security of our proposed two basic KDM-secure schemes are presented. Due to the structural similarity between and , where they primarily differ in the utilization of matrices (higher-dimensional vectors) in , we will focus on as an example for the sake of brevity. The proof for can be easily adapted to without significant effort.

*1) Correctness:* First, we show the correctness of the decryption process in our scheme. Consider a level- ciphertext that encrypts a message under modulus , with the associated secret key vector . In the decryption algorithm , we compute the inner product . When , decryption is correct, and we can correctly compute and output the encrypted message .

Next, we examine the correctness of the homomorphic operations. Consider and with the same modulus and length, individually encrypting and under the identical secret key vector .

For homomorphic addition, we have and

.

The noise present in results from adding the noise components of and .

For homomorphic multiplication, we have and

.

The noise in is approximately the square of the noise in or .

*2) Semantic Security and Key-Dependent Message (KDM) Security:* To validate our scheme’s semantic security, we proceed in two steps, demonstrating that the public key as well as the ciphertext at each level of the circuit are pseudorandom, thereby establishing semantic security.

1. As defined and proven in Section III, AIRLWE samples taken from the distribution exhibit pseudorandomness, even in the HNF form where . Therefore, the public key is pseudorandom and can be replaced with uniformly random pairs.

2. Similar to classical public-key cryptosystems that utilize the standard RLWE assumption, our encryption algorithm includes a re-randomization process based on the public key . The re-randomization lemma follows.

*Lemma 10:*Let be relatively prime to , and let other parameters be consistent with those specified in either Definition 3 or Definition 4. Then for uniformly sampled and , under the AIRLWE assumption and the RLWE assumption, we have:

where the ring elements and are uniformly random over .

*Proof:* Let and . Then, the pairs form the complete view available to a passive adversary and are, respectively, an AIRLWE sample and a RLWE sample, both with the secret . Therefore, they are pseudorandom, resulting in the conclusion. Consequently, a ciphertext containing a message component is computationally indistinguishable from the uniform and is also pseudorandom.

The above two points imply semantic security, which equivalently means IND-CPA security.

In [47], Brakerski et al. proposed a -KDM-secure SWHE scheme with a challenge function set comprising arbitrary polynomial coefficient polynomial functions of the secret key . However, their approach requires continuous increase of the ciphertext length (dimension) to securely encrypt higher-degree terms of . Specifically, they need to produce a -element ciphertext to encrypt a degree- polynomial of . Due to the inherent limitations of SWHE, the ciphertext is capable of handling only a fixed number of homomorphic computations without bootstrapping.

We demonstrate that our scheme, by extending the challenge function set, can securely encrypt polynomial functions with both constant and polynomial coefficients, including quadratic terms of the secret key , utilizing a formal ciphertext without expanding the ciphertext length. Furthermore, as our scheme is a LFHE scheme, the ciphertext is capable of performing any polynomial number of homomorphic computations without bootstrapping, as defined in our scheme. Hereafter, we offer a more formal proof of KDM security using a hybrid argument than in [47].

*Theorem 1:*Assume as a discrete Gaussian noise distribution possessing a larger standard deviation, where . Based on both the AIRLWE and RLWE assumptions, for the function family , and with outputs in the plaintext space , where and and , our scheme is a -KDM secure public-key homomorphic encryption scheme.

*Proof:* Against each PPT adversary , we prove the -KDM security of our scheme through the ensuing series of hybrids:

**Hybrid 0**: This represents the -KDM security game for the scheme where the challenge bit . The procedure involves these steps:

1. The challenger creates a public-secret key combination , then hands over the public key for the adversary .

2. The adversary submits an arbitrary number of KDM encryption queries: chooses a circuit , and the challenger sets for the specific values of chosen by , and responds to with the ciphertext .

**Hybrid 1**: We now modify the challenger’s replies to ’s KDM encryption queries.

We can presume, without loss of generality, that the adversary selected a circuit . Following this in Hybrid 0, the challenger chooses the ciphertext , where , .

Within Hybrid 1, the challenger substitutes the response ciphertext by using , where . Notice that by Lemma 4 and Lemma 6, we have . Let , then . Hence, by Lemma 3, the error distribution is statistically indistinguishable from the distribution , so it follows that and . Observing and are both valid AIRLWE instances, we can easily have that , where the ring elements and are uniformly random over . Therefore, in Hybrid 1 is computationally indistinguishable from in Hybrid 0.

Furthermore, observing that . Similarly, by Lemma 4 and Lemma 6, we have . Let , then . Hence, by Lemma 3, the error distribution is statistically indistinguishable from the distribution , so we have and .

In conclusion, Hybrid 1 is computationally indistinguishable from Hybrid 0.

**Hybrid 2**: The challenger’s responses to ’s KDM encryption queries are modified once more.

Now, we assume that the adversary chose a circuit . Then in Hybrid 0, the challenger chooses the ciphertext , where , .

In Hybrid 2, the challenger sets the ciphertext , where and . Observing that is also a valid AIRLWE instance using noise scaling for the “secret” and “noise” , we have , where the ring elements and are uniformly random over . Thus, , which can be seen as an encryption of , is a completely uniform pair. Therefore, in Hybrid 2 is computationally indistinguishable from in Hybrid 0.

Furthermore, for , we have . The small noise vector is just the normal-sized noise contained in a fresh ciphertext and will become smaller during decryption.

In conclusion, Hybrid 2 is computationally indistinguishable from Hybrid 0.

**Hybrid 3**: The responses to KDM encryption queries are further adjusted.

In Hybrid 1, the challenger sets the ciphertext .

In Hybrid 2, the challenger sets the ciphertext .

In Hybrid 3, the challenger sets the ciphertext , where . Here, indicates the uniform distribution over .

It should be emphasized that in Hybrid 1, it has been established that cannot be distinguished computationally from a random ring element sampled uniformly. Moreover, is indeed a RLWE instance, which implies that is pseudorandom and .

Similarly, in Hybrid 2, it has been established that is likewise computationally indistinguishable from a random ring element sampled uniformly, and is indeed a RLWE sample encrypting the “message” , implying that is pseudorandom and .

In conclusion, and thus Hybrid 1, Hybrid 2 and Hybrid 3 are computationally indistinguishable from each other.

**Hybrid 4**: The method for answering the KDM encryption queries is adjusted once again.

In Hybrid 3, the challenger sets the ciphertext , where the ring elements and are uniformly random over .

In Hybrid 4, the challenger sets the ciphertext .

By Lemma 10, is computationally indistinguishable from , making Hybrid 3 and Hybrid 4 computationally indistinguishable.

Observe that Hybrid 4 corresponds with respect to the -KDM security game for the scheme under the challenge bit . The procedure is as follows:

1. The challenger creates a public-secret key combination , then hands over the public key for the adversary .

2. The adversary submits an arbitrary number of KDM encryption queries: chooses a circuit , and the challenger sets , and responds to with the ciphertext .

Thus, as previously established, the computational indistinguishability between Hybrid 0 and Hybrid 4 confirms the KDM security of the scheme as asserted.

To achieve -KDM security when encrypting higher-order polynomials of the secret key , utilizing homomorphic operations to generate ciphertexts that encrypt such polynomials is a straightforward and effective approach. As demonstrated in [47], we can modify our scheme’s encryption algorithm to directly construct ciphertexts in the format of homomorphic computation results, with gradually increasing lengths, to securely encrypt polynomials of degree of . However, overly long ciphertexts are detrimental to the efficiency of homomorphic computations and would cause a significant increase in the ciphertext length in our subsequent construction of a multi-key homomorphic encryption scheme. Therefore, considering space constraints, we leave out the detailed implementation of this approach.

# V. KDM-Secure BGV Type MKFHE Scheme

This part provides the detailed methodology for extending the two basic schemes introduced earlier into a KDM-secure BGV type multi-key scheme. We utilize the modulus switching technique from Section II-D and demonstrate the circular-secure evaluation key generation and key switching process, specifically tailored to our new scheme architecture based on the AIRLWE assumption. For clarity, we use the notation () to indicate a BGV (GSW) ciphertext, as described in Section IV, which might not be fresh and is decryptable via the secret key vector for obtaining the message .

## A. KDM-Secure Multi-Key BGV

**:** Considering the security parameter and a bound for the circuit depth, along with as the limit for the quantity of keys, generate moduli decreasing as at each level. Additionally, select an integer that is relatively prime to each value. Our operations are performed within the rings , , and , where is the set of invertible elements in . Let be a -bounded error distribution over for . Define , then randomly select public vectors corresponding to . The public parameter  is implicitly considered as part of the input for all subsequent algorithms. Consider  as an ordered set of user indexes associated with the ciphertext, sorted in ascending order and containing no duplicates. A ciphertext may then be represented as a ciphertext tuple .

**:** Based on the public parameter , create keys corresponding to circuit depth  for the -th party, with  ranging from  to 0.

1) Draw the secret key and set the secret key vector as .

2) Select at random, then calculate the public key intended for the -th party

.

3) Utilize to compute the evaluation key generation material

where for :

**:** Given a message together with the public key , randomly draw , then calculate a level- ciphertext , followed by outputting the ciphertext tuple .

**:** Given a ciphertext tuple associated with parties, where is an encryption of some message , along with an additional user set such that , the generated result is an extended ciphertext tuple . The extension method is described below:

1) Split the ciphertext as ordered sub-vectors indexed by , resulting in , with the associated secret key vector .

2) The extended ciphertext comprises ordered sub-vectors, indexed via . Specifically, .

Assign . If an index from also exists within , then assign ; if not, assign . The associated secret key vector used regarding decryption is .

It can be easily confirmed that , where is a small noise vector.

**:** On input a level- ciphertext where and its corresponding secret key vectors . Let , output the message .

**:** This algorithm is designed to evaluate Boolean circuits on encrypted inputs. Assume we have a series of ciphertexts for , in which every ciphertext resides at an identical level (if necessary, apply key switching along with modulus switching algorithms to achieve this). The ciphertext inputs may be either freshly generated or intermediate results from any homomorphic computation, as permitted through the multi-hop property. Let . The process for evaluating a Boolean circuit is detailed as described below:

1) Regarding each , execute to obtain an extended ciphertext that maintains the encryption of the same message with the extended secret key vector .

2) Produce the evaluation keys through computing:

where .

3) Process the circuit with the two fundamental homomorphic operations and . After each layer, or specifically after each multiplication layer (as addition increases noise at a much slower rate compared to multiplication, making it unnecessary to refresh after additions, even with high fan-in), the ciphertext refreshing procedure is invoked to diminish noise and transition the ciphertext to another level. This refreshing procedure includes two steps: and .

**:** On input two (extended) ciphertexts  at the same level- under the same secret key vector .

1) Compute under the secret key vector .

2) In standard BGV type schemes, we should now interpret as a ciphertext under and compute under the secret key vector . However, since our scheme satisfies circular security, none of the parties need to change their keys, and because the homomorphic addition operation does not lead to ciphertext dimension expansion, this step can be safely omitted.

3) Compute and output .

**:** On input two (extended) ciphertexts  at the same level- under the same secret key vector .

1) Compute , where the resulting ciphertext corresponds to the tensored secret key vector .

2) Compute , corresponding to the secret key vector .

3) Compute and output .

*Correctness and Security:*Benefiting from the straightforward ciphertext extension method of BGV type MKFHE schemes [23], [24], an extended BGV ciphertext is simply a concatenation of multiple single-key BGV ciphertexts. Therefore, it is straightforward to demonstrate that the correctness and security of our scheme are consistent with those of our basic scheme.

Our scheme maintains -KDM security for participating parties. Recall that an extended ciphertext , with which the associated secret key vector used for decryption is . For , -KDM security implies that messages of the form are also valid quadratic polynomials involving the secret keys within the challenge function set , even if ,, and are distinct. Formally, proving this necessitates an internal hybrid argument across every KDM encryption query, wherein such secret keys of the involved participants are substituted one by one in each query. Due to space constraints, the repeated demonstration of correctness and security for the scheme is omitted.

## B. Evaluation Key Generation and Key Switching under Circular Security

Here, we provide an elaborate explanation of the evaluation key generation algorithm . This algorithm receives the public keys and evaluation key generation materials belonging to the involved parties and produces the (extended) BGV type evaluation key. Additionally, we will also demonstrate the circular security associated with the evaluation key generation and key switching processes. This allows our scheme to let users compute throughout the entire Boolean circuit without needing to change their secret keys, independent of any other security assumptions.

Considering our summary and analysis of the evaluation key generation and key switching under Section II-E, and the structure of our scheme’s (extended) ciphertexts, the required evaluation key is given by:

where and such that:

where is a short noise vector. In essence, is a ciphertext that encrypts the message with the secret key vector .

In the process of computing the evaluation keys, we leverage the hybrid homomorphic multiplication defined in [61] and [24]. This technique is utilized for performing homomorphic multiplication (tensor product) between GSW type and BGV type ciphertexts, reducing the size of ciphertexts and noise, and enhancing the performance of homomorphic computation.

*Lemma 11 (Hybrid Homomorphic Multiplication [24], [61]):*The hybrid homomorphic multiplication is defined as

Let represent a valid ciphertext containing the message , while denotes a valid ciphertext containing the message . Let denote the noise contained in ciphertext , and denote the variance of . Then, is a ciphertext of the message , and the noise properties satisfy:

where denotes the degree of the cyclotomic polynomial, refers to an integer, indicates the upper limit for the noise coefficients, and represents the standard deviation of the error distribution . From the above lemma, it can be seen that the primary requirement is to control the norm of the message .

**:** Observe that comprises elements, thus includes a set of evaluation key generation materials , as well as the public keys associated with the parties in . To produce a level- evaluation key as outlined in (1), proceed with the following computation:

1) For each , use the GSW extend algorithm to generate the extended ciphertext corresponding to the secret key vector , which encrypts :

2) For each and , use the BGV extend algorithm to generate the extended ciphertext corresponding to the secret key vector , which encrypts :

3) For each , , and , we can compute with the above components:

Finally, we can organize this auxiliary information and output the level- evaluation key required for the key switching technique:

Note that in the above computation steps, , and in practice:

Since the extended ciphertexts and are encryptions of shared values that can be securely made public when and , they can be generated using the public keys of any single participating party. This is achieved by creating single-key ciphertexts and then employing the corresponding type of ciphertext extension algorithms to obtain the extended ciphertexts under the secret key vector .

Next, we demonstrate the circular security of our scheme’s evaluation key generation algorithm and key switching algorithm. To prove their circular security, we need to show that the materials , , , , , and used during computation expose no information regarding the involved secret keys .

Observing that each ciphertext , , , , and is an encryption under its corresponding secret key vector of polynomials of the secret keys contained in that secret key vector, with degrees no greater than 2 and specific coefficients that belong to the challenge function set of our scheme. Additionally, are encryptions of different random values, independent of the participating secret keys (since public keys do not expose secret key information). Therefore, building upon the AIRLWE assumption and the -KDM security of the scheme, all these ciphertexts cannot be distinguished computationally from encryptions of zero and uniform. Consequently, the evaluation key generation process and the key switching process of our scheme are proven to possess circular security.

## C. Distributed Decryption and Dynamic Exit

In the classical definition of the MKFHE primitive, decrypting a corresponding extended ciphertext requires collecting all the secret keys of the relevant participants. For instance, given a ciphertext tuple , we need to use the extended key vector to decrypt the ciphertext . However, in practical applications, presuming that a single participant possesses the secret keys of multiple participants is unrealistic. This is particularly concerning for our KDM-secure MKFHE scheme.

In [18], Mukherjee et al. proposed a threshold decryption protocol to address this issue. Later, in [25], Chen et al. summarized it into a distributed decryption protocol based on secure solutions like the noise flooding technique. We demonstrate that this distributed decryption idea also applies to our scheme, implemented through the following two algorithms:

**:** In this algorithm, each participant receives the -th entry of the extended ciphertext (recall the ciphertext parsing in the algorithm), samples a noise , computes and returns a partial decryption result .

**:** This algorithm computes and outputs the actual message encrypted in the extended ciphertext, .

The correctness of the distributed decryption process in our scheme is evident. For an extended ciphertext , we have:

.

The security is also evident. In the algorithm, each participant essentially computes . Since cannot be distinguished computationally from uniform in as stated in Lemma 10, the pair is exactly a RLWE instance using the noise scaling technique. Thus, each is pseudorandom and can be securely shared.

For practical considerations, [19] and [20] respectively proposed the concepts of multi-hop MKFHE and fully dynamic MKFHE, which allow any participant to join the homomorphic evaluation dynamically at any point in time. However, in MPC scenarios, it is also common for one (or more) participants to wish to exit the homomorphic computation at any moment. These participants may allow the remaining participants to continue using the intermediate outcomes of the joint evaluation, along with the data of the exiting participants (both in encrypted form), to proceed with homomorphic operations.

Thus, we propose a new concept of “dynamic exit”, which allows any participant to exit the homomorphic evaluation dynamically at any point in time. Once they exit, they no longer participate in subsequent ciphertext extensions, homomorphic operations, decryptions, and other processes.

We demonstrate that implementing dynamic exit in our scheme is straightforward. Suppose participant wants to exit at level- of the circuit. For all extended ciphertext tuples associated with that the remaining participants still need and have been approved by for continued use, the following algorithm is invoked.

**:** Given the extended ciphertext , compute . Then set , i.e., , and . Output the new ciphertext tuple .

For single-key ciphertexts belonging exclusively to participant , the ciphertext extension algorithm can be invoked before executing this algorithm. One can readily demonstrate that the new extended ciphertext corresponds to the secret key vector and that , where is a small noise vector.

# VI. Parameters and Comparisons

In this paper, given a security parameter , we set the parameters and for the AIRLWE problem in accordance with the standard RLWE problem settings [4], [14]. We ensure that for each , to guarantee the security of our schemes, where . Additionally, we utilize error distributions and . To satisfy the correctness requirements of Lemma 3, Lemma 5, Lemma 6 and Corollary 1, the discrete Gaussian distribution must have a standard deviation of , and we choose such that . Following the analysis in [47], we adopt , a super-polynomial function, where represents a super-logarithmic value. This choice supports the worst-case to average-case reduction from the shortest vector problem on ideal lattices to the AIRLWE problem under the proposed parameter settings. The larger standard deviation should be set as because, in the hybrid argument of Section IV-C, we need elements sampled from to overwhelm the quadratic polynomials over . However, unlike in [47], is not utilized in our scheme (e.g., in the algorithm); it is only used in the hybrid argument to denote the coefficient range of the challenge function set, thus not causing any efficiency loss in the scheme.

In [24], Li et al. avoided using and in evaluation key generation by restricting all coefficients of each participant’s secret key to , i.e., . For security and practicality, we do not adopt this approach, providing each user with a richer key space. We select the plaintext modulus such that . Then according to Lemma 6, samples drawn from will, with overwhelming probability, lie within the plaintext space, allowing our scheme to encrypt messages like users’ secret keys correctly. Additionally, to ensure correctness when encrypting quadratic polynomials of secret keys, we need .

TABLE I

**Main Performance Comparisons among Single-Key Schemes.** Since Some Schemes Involve Ciphertext Extension Algorithms, We Utilize to Represent the Actual Participant Count in the Computation. Represents the Circuit Depth for Which the Scheme Is Intended for Homomorphic Evaluation. Indicates the Dimension of the Hardness Assumption, while Represents the Modulus.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Scheme | Hardness Assumption |  |  | Secret Key Vector Size | Ciphertext Size | Security |
| BV11 [47] | PLWE |  |  |  |  | -KDM security |
| BGV14 [4] | RLWE |  | / |  |  | IND-CPA |
|  | RLWE, AIRLWE |  | / | 2 |  | -KDM security |
| in CZW17 [23] | RLWE |  | / | 2 |  | IND-CPA |
| in LZY+19 [24] | RLWE |  | / | 2 |  | IND-CPA |
|  | RLWE, AIRLWE |  | / | 2 |  | -KDM security |

TABLE II

**Main Performance Comparisons among Multi-Key Schemes.** Represents the Actual Participant Count in the Computation. Represents the Circuit Depth for Which the Scheme Is Intended for Homomorphic Evaluation. Indicates the Dimension of the Hardness Assumption, while Represents the Modulus.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scheme | Hardness Assumption | Extended Secret Key Vector Size | Ciphertext Size | Security |
| CZW17 [23] | RLWE |  |  | IND-CPA |
| LZY+19 [24] | RLWE |  |  | IND-CPA |
| KKL+23 [26] | RLWE |  |  | IND-CPA, circular security assumption |
|  | RLWE, AIRLWE |  |  | -KDM security |

Since we have demonstrated that our leveled (MK)FHE schemes are KDM-secure (circular-secure), each participant can safely use the same secret key throughout the circuit evaluation without relying on an additional circular security assumption. In existing BGV type schemes [4], [23], [24], a unique random secret key is required for each user at each level of the Boolean circuit to avoid introducing circular security assumptions. This means their schemes require secret keys and public keys, where is the circuit depth, and denotes the participant count. In contrast, our scheme only requires secret keys, independent of the circuit depth. Similar to our scheme, existing BFV type and CKKS type schemes [6], [9], [25], [26] also just require secret keys, but they rely on circular security assumptions.

The primary performance of different schemes is compared in Tables 1 and 2. To ensure fairness in comparison, we select the parameters as well as performance of [47] with polynomial degree , which is just sufficient to securely encrypt quadratic terms of secret keys with polynomial coefficients. For [4], we use parameters and performance for the RLWE setting.

# VII. Conclusion

In this paper, we improve upon the standard RLWE assumption by introducing a new hardness assumption called AIRLWE. Based on the AIRLWE assumption, we construct a KDM-secure BGV type single-key FHE scheme and a KDM-secure ring-based GSW type single-key FHE scheme with ring element plaintexts, rigorously proving their correctness and security. We then leverage these basic schemes to build a BGV type MKFHE scheme with identical KDM security properties and demonstrate how to generate the necessary evaluation keys using hybrid homomorphic multiplication without restricting the coefficients of the secret keys. Additionally, we introduce the concept of “dynamic exit” for MKFHE schemes and show how it can be simply implemented in our work. Due to the circular security of our schemes’ evaluation key generation and key switching process, we achieve optimizations in performance compared to previous schemes.

References

1. C. Gentry, “Fully homomorphic encryption using ideal lattices,” in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169–178.
2. Z. Brakerski and V. Vaikuntanathan, “Efficient fully homomorphic encryption from (standard) LWE,” *SIAM Journal on computing*, vol. 43, no. 2, pp. 831–871, 2014.
3. I. Damgård, V. Pastro, N. Smart, and S. Zakarias, “Multiparty computation from somewhat homomorphic encryption,” in *Annual Cryptology Conference*, Springer, 2012, pp. 643–662.
4. Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “(Leveled) fully homomorphic encryption without bootstrapping,” *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, pp. 1–36, 2014.
5. Z. Brakerski, “Fully homomorphic encryption without modulus switching from classical GapSVP,” in *Annual cryptology conference*, Springer, 2012, pp. 868–886.
6. J. Fan and F. Vercauteren, “Somewhat practical fully homomorphic encryption,” *Cryptology ePrint Archive*, 2012.
7. C. Gentry, S. Halevi, and N. P. Smart, “Fully homomorphic encryption with polylog overhead,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2012, pp. 465–482.
8. 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*, Springer, 2013, pp. 75–92.
9. J. H. Cheon, A. Kim, M. Kim, and Y. Song, “Homomorphic encryption for arithmetic of approximate numbers,” in *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23*, Springer, 2017, pp. 409–437.
10. S. Halevi and V. Shoup, “Bootstrapping for helib,” *Journal of Cryptology*, vol. 34, no. 1, p. 7, 2021.
11. A. López-Alt, E. Tromer, and V. Vaikuntanathan, “On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption,” in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, 2012, pp. 1219–1234.
12. J. Hoffstein, J. Pipher, and J. H. Silverman, “NTRU: A ring-based public key cryptosystem,” in *International algorithmic number theory symposium*, Springer, 1998, pp. 267–288.
13. O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” *Journal of the ACM (JACM)*, vol. 56, no. 6, pp. 1–40, 2009.
14. V. Lyubashevsky, C. Peikert, and O. Regev, “On ideal lattices and learning with errors over rings,” *Journal of the ACM (JACM)*, vol. 60, no. 6, pp. 1–35, 2013.
15. M. Clear and C. McGoldrick, “Multi-identity and multi-key leveled FHE from learning with errors,” in *Advances in Cryptology--CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II 35*, Springer, 2015, pp. 630–656.
16. J. Alperin-Sheriff and C. Peikert, “Faster bootstrapping with polynomial error,” in *Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I 34*, Springer, 2014, pp. 297–314.
17. Z. Brakerski and V. Vaikuntanathan, “Lattice-based FHE as secure as PKE,” in *Proceedings of the 5th conference on Innovations in theoretical computer science*, 2014, pp. 1–12.
18. P. Mukherjee and D. Wichs, “Two round multiparty computation via multi-key FHE,” in *Advances in Cryptology–EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II 35*, Springer, 2016, pp. 735–763.
19. C. Peikert and S. Shiehian, “Multi-key FHE from LWE, revisited,” in *Theory of cryptography conference*, Springer, 2016, pp. 217–238.
20. Z. Brakerski and R. Perlman, “Lattice-based fully dynamic multi-key FHE with short ciphertexts,” in *Annual international cryptology conference*, Springer, 2016, pp. 190–213.
21. H. Chen, I. Chillotti, and Y. Song, “Multi-key homomorphic encryption from TFHE,” in *Advances in Cryptology–ASIACRYPT 2019: 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8–12, 2019, Proceedings, Part II 25*, Springer, 2019, pp. 446–472.
22. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, “Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE,” in *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, 2017, pp. 377–408.
23. L. Chen, Z. Zhang, and X. Wang, “Batched multi-hop multi-key FHE from ring-LWE with compact ciphertext extension,” in *Theory of Cryptography: 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II 15*, Springer, 2017, pp. 597–627.
24. N. Li, T. Zhou, X. Yang, Y. Han, W. Liu, and G. Tu, “Efficient multi-key FHE with short extended ciphertexts and directed decryption protocol,” *IEEE Access*, vol. 7, pp. 56724–56732, 2019.
25. H. Chen, W. Dai, M. Kim, and Y. Song, “Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 395–412.
26. T. Kim, H. Kwak, D. Lee, J. Seo, and Y. Song, “Asymptotically faster multi-key homomorphic encryption from homomorphic gadget decomposition,” in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 726–740.
27. P. Laud and R. Corin, “Sound computational interpretation of formal encryption with composed keys,” in *International Conference on Information Security and Cryptology*, Springer, 2003, pp. 55–66.
28. P. Adao, G. Bana, J. Herzog, and A. Scedrov, “Soundness of formal encryption in the presence of key-cycles,” in *Computer Security–ESORICS 2005: 10th European Symposium on Research in Computer Security, Milan, Italy, September 12-14, 2005. Proceedings 10*, Springer, 2005, pp. 374–396.
29. J. Camenisch and A. Lysyanskaya, “An efficient system for non-transferable anonymous credentials with optional anonymity revocation,” in *Advances in Cryptology—EUROCRYPT 2001: International Conference on the Theory and Application of Cryptographic Techniques Innsbruck, Austria, May 6–10, 2001 Proceedings 20*, Springer, 2001, pp. 93–118.
30. F. Kitagawa, T. Matsuda, and K. Tanaka, “CCA security and trapdoor functions via key-dependent-message security,” *Journal of Cryptology*, vol. 35, no. 2, p. 9, 2022.
31. A. Lombardi, W. Quach, R. D. Rothblum, D. Wichs, and D. J. Wu, “New constructions of reusable designated-verifier NIZKs,” in *Annual International Cryptology Conference*, Springer, 2019, pp. 670–700.
32. F. Kitagawa and T. Matsuda, “CPA-to-CCA transformation for KDM security,” in *Theory of Cryptography: 17th International Conference, TCC 2019, Nuremberg, Germany, December 1–5, 2019, Proceedings, Part II 17*, Springer, 2019, pp. 118–148.
33. T. Acar, M. Belenkiy, M. Bellare, and D. Cash, “Cryptographic agility and its relation to circular encryption,” in *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*, Springer, 2010, pp. 403–422.
34. J. Black, P. Rogaway, and T. Shrimpton, “Encryption-scheme security in the presence of key-dependent messages,” in *Selected Areas in Cryptography: 9th Annual International Workshop, SAC 2002 St. John’s, Newfoundland, Canada, August 15–16, 2002 Revised Papers 9*, Springer, 2003, pp. 62–75.
35. R. D. Rothblum, “On the circular security of bit-encryption,” in *Theory of Cryptography: 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*, Springer, 2013, pp. 579–598.
36. D. Cash, M. Green, and S. Hohenberger, “New definitions and separations for circular security,” in *International Workshop on Public Key Cryptography*, Springer, 2012, pp. 540–557.
37. A. Bishop, S. Hohenberger, and B. Waters, “New circular security counterexamples from decision linear and learning with errors,” in *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, 2015, pp. 776–800.
38. V. Koppula, K. Ramchen, and B. Waters, “Separations in circular security for arbitrary length key cycles,” in *Theory of Cryptography: 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II 12*, Springer, 2015, pp. 378–400.
39. V. Koppula and B. Waters, “Circular security separations for arbitrary length cycles from LWE,” in *Annual International Cryptology Conference*, Springer, 2016, pp. 681–700.
40. R. Goyal, V. Koppula, and B. Waters, “Separating IND-CPA and circular security for unbounded length key cycles,” in *IACR International Workshop on Public Key Cryptography*, Springer, 2017, pp. 232–246.
41. R. Goyal, V. Koppula, and B. Waters, “Separating semantic and circular security for symmetric-key bit encryption from the learning with errors assumption,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2017, pp. 528–557.
42. N. Alamati and C. Peikert, “Three’s compromised too: Circular insecurity for any cycle length from (ring-) lwe,” in *Annual International Cryptology Conference*, Springer, 2016, pp. 659–680.
43. D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky, “Circular-secure encryption from decision diffie-hellman,” in *Advances in Cryptology–CRYPTO 2008: 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings 28*, Springer, 2008, pp. 108–125.
44. B. Applebaum, D. Cash, C. Peikert, and A. Sahai, “Fast cryptographic primitives and circular-secure encryption based on hard learning problems,” in *Advances in Cryptology-CRYPTO 2009: 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, Springer, 2009, pp. 595–618.
45. B. Barak, I. Haitner, D. Hofheinz, and Y. Ishai, “Bounded key-dependent message security,” in *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*, Springer, 2010, pp. 423–444.
46. B. Applebaum, “Key-dependent message security: Generic amplification and completeness,” *Journal of cryptology*, vol. 27, no. 3, pp. 429–451, 2014.
47. Z. Brakerski and V. Vaikuntanathan, “Fully homomorphic encryption from ring-LWE and security for key dependent messages,” in *Annual cryptology conference*, Springer, 2011, pp. 505–524.
48. Z. Brakerski, S. Goldwasser, and Y. T. Kalai, “Black-box circular-secure encryption beyond affine functions,” in *Theory of Cryptography: 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings 8*, Springer, 2011, pp. 201–218.
49. M. Hajiabadi and B. M. Kapron, “Reproducible circularly secure bit encryption: Applications and realizations,” *Journal of Cryptology*, vol. 30, pp. 1187–1237, 2017.
50. F. Kitagawa and T. Matsuda, “Circular security is complete for KDM security,” in *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26*, Springer, 2020, pp. 253–285.
51. B. Waters and D. Wichs, “Universal amplification of kdm security: From 1-key circular to multi-key kdm,” in *Annual International Cryptology Conference*, Springer, 2023, pp. 674–693.
52. D. Micciancio, “Improving lattice based cryptosystems using the Hermite normal form,” in *International Cryptography and Lattices Conference*, Springer, 2001, pp. 126–145.
53. M. Fischlin and A. Mittelbach, “An overview of the hybrid argument,” *Cryptology ePrint Archive*, 2021.
54. S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan, “Robustness of the learning with errors assumption,” 2010.
55. V. Lyubashevsky and D. Micciancio, “Generalized compact knapsacks are collision resistant,” in *International Colloquium on Automata, Languages, and Programming*, Springer, 2006, pp. 144–155.
56. N. Stephens-Davidowitz, “Discrete Gaussian sampling reduces to CVP and SVP,” in *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2016, pp. 1748–1764.
57. D. Micciancio and O. Regev, “Worst-case to average-case reductions based on Gaussian measures,” *SIAM Journal on Computing*, vol. 37, no. 1, pp. 267–302, 2007.
58. D. Stehlé and R. Steinfeld, “Making NTRU as secure as worst-case problems over ideal lattices,” in *Advances in Cryptology–EUROCRYPT 2011: 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings 30*, Springer, 2011, pp. 27–47.
59. D. Micciancio and O. Regev, “Lattice-based cryptography,” in *Post-quantum cryptography*, Springer, 2009, pp. 147–191.
60. S. Goldwasser and S. Micali, “Probabilistic encryption,” *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270–299, 1984, doi: <https://doi.org/10.1016/0022-0000(84)90070-9>.
61. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachene, “Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds,” in *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*, Springer, 2016, pp. 3–33.

**Liang Zhou** received his B.Eng. degree in computer science and technology from Guangxi University in 2022. He is currently pursuing an M.Eng. degree in Computer Technology at Guangxi University. His research areas encompass cryptography and cloud computing security.

**Ruwei Huang** received her B.Eng., M.Eng. degree in computer science and technology at Guangxi University in 2001 and 2004 respectively, received Ph.D. degree in computer science and technology at Xi’an Jiaotong University, China, in 2012. She is recently engaged in cryptography and cloud computing security as an associate professor at Guangxi University.

**Sai Hu** received the B.S. degree in Software Engineering from East China University of Technology in 2022. He is currently a graduate student at the College of Computer Science and Electronic Information of Guangxi University. His research interests include cryptography, information security, and homomorphic encryption.

1. This work was supported by the National Natural Science Foundation Project of China (No. 62062009), Guangxi Key Research and Development Program Project (No.AB24010340). *(Corresponding author: Ruwei Huang.)*

   Liang Zhou is with the School of Computer and Electronic Information, G-uangxi University, Nanning 530004, China (e-mail: zhouliang\_143@qq.com).

   Ruwei Huang is with the School of Computer and Electronic Information, Guangxi University, Nanning 530004, China, and also with the Guangxi Key Laboratory of Multimedia Communications and Network Technology, Nanning 530004, China (e-mail: ruweih@gxu.edu.cn).

   Sai Hu is with the School of Computer and Electronic Information, Guang-xi University, Nanning 530004, China (e-mail: 1427223295@qq.com). [↑](#footnote-ref-1)