Diamond iO: A Straightforward Construction of Indistinguishability Obfuscation from Lattices

Sora Suegami*, Enrico Bottazzi†

February 2025

Abstract

Indistinguishability obfuscation (iO) has seen remarkable theoretical progress, yet it remains impractical due to its high complexity and inefficiency. A common bottleneck in recent iO schemes is the reliance on bootstrapping techniques from functional encryption (FE) into iO, which requires recursively invoking the FE encryption algorithm for each input bit—creating a significant barrier to practical iO schemes.

In this work, we propose diamond iO, a new lattice-based iO construction that replaces the costly recursive encryption process with lightweight matrix operations. Our construction is proven secure under the learning with errors (LWE) and evasive LWE assumptions, as well as our new assumption—all-product LWE—in the pseudorandom oracle model. By leveraging the FE scheme for pseudorandom functionalities introduced by Agrawal et al. (ePrint'24) in a non-black-box manner, we remove the reliance on prior FE-to-iO bootstrapping techniques and thereby significantly reduce complexity. A remaining challenge is to reduce our new assumption to standard assumptions such as LWE, further advancing the goal of a practical and sound iO construction.

1 Introduction

Program obfuscation aims to make programs unintelligible without altering their functionality. A general-purpose obfuscator that offers an ideal security guarantee converts any program into an obfuscated version that can be distributed safely, revealing no information beyond its input-output behavior.

Unfortunately, Barak et al. [BGI+01] proved that such an obfuscator, defined as a virtual black-box obfuscator, for arbitrary programs is impossible to achieve.[1] They proposed a weaker security notation of the obfuscator called indistinguishability obfuscation (iO), where the obfuscated versions of two arbitrary programs with the same size and functionality should be computationally indistinguishable. Despite its limited security guarantee, iO has been seen as a "central hub [SW14]" for its role in building numerous powerful cryptographic primitives [SW14, GGG+14, BP15, BGL+15, GGH+16, BZ17, AFH+20].

The first candidate of iO in [GGH+16] kick-started a quest for constructing iO from reasonable cryptographic assumptions. Following a series of studies in this context [GGH+16, PST14, GGH15, Lin16, AS17, AJL+19, Agr19, JLMS19, GJLS21], Jain et al. [JLS21] have succeeded in building iO relying solely on four standard assumptions. Furthermore, subsequent works [JLS22, RVV25] reduced the number of the necessary assumptions to just three.

However, most studies on iO have not focused on evaluating its concrete efficiency or implementing the proposed schemes, and thus iO remains more of a theoretical cryptographic primitive than a practical one. In recently proposed iO schemes [Lin16, AJL+19, BDGM20a, BDGM20b, JLS21,

*Machina IO, Privacy & Scaling Explorations, Ethereum Foundation, sorasuegami@pse.dev

†Machina IO, Privacy & Scaling Explorations, Ethereum Foundation, enrico@pse.dev

[1] However, the work of [JLLW23] showed a construction of an ideal obfuscation for arbitrary programs in the pseudorandom oracle model. In this model, a non-black-use of the random oracle is formally allowed.

JLS22, RVV25, GP21, GJLS21, DQV+21], one common bottleneck is the reliance on bootstrapping techniques from functional encryption (FE) to iO [AJ15, BV18]. We note that FE is a variant of public-key encryption in which a special secret key associated with a **public** function f—called functional secret key sk_f—allows its holder to decrypt an encryption of an input x and learn only the output $f(x)$ [BSW11]. These bootstrapping techniques are costly because they recursively invoke the FE encryption algorithm for each input bit—specifically, they run the FE encryption algorithm as f during decryption to produce other FE ciphertexts. As a result, the size and complexity of the FE encryption algorithm becomes a lower bound on the size of the functions that must be evaluated by the underlying FE scheme, creating a significant barrier to practical iO constructions.

Our contributions. To address this issue, we propose diamond iO: a straightforward construction of iO using lattice techniques. This replaces the costly FE recursive encryption process with simple matrix operations. The security of our construction relies on the learning with errors (LWE) and evasive LWE assumptions, as well as a new lattice assumption we introduce, called all-product LWE, in the pseudorandom oracle model (PROM) [JLLW23]. A non-black-box use of the FE scheme for pseudorandom functionalities proposed in [AKY24a] enables us to remove the reliance on the existing FE-to-iO bootstrapping techniques.

This paper is organized as follows. The rest of Section 1 reviews related works. Section 2 provides a technical overview of our iO construction. Section 3 introduces the notions and preliminaries necessarily to describe our construction. Section 4 presents our iO construction along with its correctness and security proofs. Finally, Section 5 concludes the paper.

1.1 Related Works

Bootstrapping from FE to iO. We review the bootstrapping technique from FE to iO, which is a common bottleneck in recent iO schemes. Goldwasser et al. [GGG+14] showed how to build iO from secret-key multi-input functional encryption (MiFE). Subsequently, Ananth and Jain [AJ15] proved that this secret-key MIFE scheme can be constructed from a public-key single-key FE scheme that is compact, i.e., the running time of the encryption algorithm must be independent of the output size. Furthermore, Bitansky et al. [BV18] presented a similar bootstrapping technique from FE to iO but with a weaker requirement on the FE scheme: that is, the encryption running time only must be sublinear in the output size. Taken together, these results imply that the single-key FE whose encryption algorithm runs in sublinear time in the output size suffices for building iO.

Specifically, the technique proposed in [AJ15] constructs a secret-key MiFE scheme for any function with n distinct inputs, denoted by MiFE_i, by recursively increasing the arity of MiFE_{i-1} by one bit. As an initial ingredient for this recursion, the public-key compact FE, referred to as PKFE, is employed. To obfuscate a circuit f, the obfuscator provides a functional secret key sk_f with respect to a function $U(\mathsf{Sym.Enc}(\cdot, f), \cdot)$: it takes as input a key of the symmetric key encryption (SKE) scheme and an input $\mathbf{x}_L := (x_1, \ldots, x_n)$, decrypts the hardcoded ciphertext of f—denoted by $\mathsf{Sym.Enc}(\cdot, f)$—with the provided SKE key, and returns $f(\mathbf{x}_L)$ by evaluating a universal circuit U. Additionally, the obfuscator provides the evaluator with a pair of ciphertexts for each of the n input bits. For $i = 1$, these two ciphertexts are, respectively, the encryption of $(0, sk)$ and $(1, sk)$ under MiFE_2. For $i \in \{2, \ldots, L-1\}$, the pair of ciphertexts are functional secret keys with respect to MiFE_i for a function that takes the $(i-1)$-bit input vector $\mathbf{x}_{i-1} := (x_1, \ldots, x_{i-1})$, concatenates it with either 0 or 1, and returns its re-encryption under MiFE_{i+1}. Lastly, for $i = L$, the two ciphertexts are functional secret keys with respect to MiFE_L for a function that takes the $(L-1)$-bit input vector $\mathbf{x}_{L-1}$, concatenates it with either 0 or 1, and returns its re-encryption with respect to PKFE.

To evaluate the obfuscated circuit on an input $\mathbf{x}_L$, the evaluator picks the ciphertext corresponding to their first input bit and recursively decrypts it with the functional secret key corresponding to the next input bit. By iterating this process for each input bit, the evaluator eventually obtains an encryption of $(\mathbf{x}_L, K)$ under FE, which can be decrypted with sk_f to recover $f(\mathbf{x}_n)$. In this way, the recursive FE encryption lets the evaluator obtain a single ciphertext that contains both the obfuscator's private date, namely the symmetric key K, and the evaluator's L-bit input $\mathbf{x}_L$.

The technique proposed in [BV18] is similar to the one in [AJ15]. However, instead of building a MiFE scheme, it constructs a variant of the public-key single-input FE scheme called puncturable FE (PFE). While it is impossible to have a functional secret key hide the function in the public-key setting [BV18], that of PFE guarantees a weaker nortion of the function-hiding property. The authors first showed that a public-key single-input FE scheme with sublinear encryption efficiency can be extended to a PFE scheme. They then constructed iO by recursively invoking the PFE encryption algorithm for each input bit. In this manner, even though multiple studies have addressed FE-to-iO bootstrapping techniques, to the best of our knowledge, they all rely on the recursive FE encryption process.

Lattice-based iO. Existing iO schemes from standard assumptions [JLS21, JLS22, RVV25] rely on bilinear maps and are therefore not post-quantum secure. To achieve (plausibly) post-quantum iO, a parallel line of research [BDGM20b, DQV+21, GP21, WW21, BDGM20a, AKY24b, BDJ+24] focuses on lattice-based iO constructions.

As the first study in this direction, Brakerski et al. [BDGM20a] introduced a new primitive called split-FHE and extended it to exponential iO (XiO), which implies iO [LPST16]. Their proposed split-FHE construction employs a standard LWE-based FHE scheme, a linearly-homomorphic encryption (LHE) scheme with short decryption hints, and a hash function that instantiates the random oracle.[2] Unfortunately, its security analysis remains heuristic due to partial leakage of smudging noises.

Subsequent works [BDGM20b, DQV+21, GP21, WW21] focused on improving the security analysis by providing constructions that rely on different flavors of the circular security assumption. However, Hopkins et al. [HJL21], followed by Jain et al. [JLLS23], demonstrated the fragility of these constructions by providing polynomial-time attacks against some of their new assumptions.

Unlike the above works, Agrawal et al. [AKY24b] and Branco et al. [BDJ+24] independently propose constructions of iO for pseudorandom functionalities, proven secure under commonly used assumptions, namely LWE and evasive LWE assumptions [Wee22]. Although the latter is still relatively new, it has already been employed in various cryptographic schemes such as broadcast encryption [Wee22], attribute-based encryption [WWW22, HLL23, HLL24], and witness encryption [Tsa22, VWW22]. Furthermore, Branco et al. [BDJ+24] introduce a transformation from iO for pseudorandom functionalities into one for general circuits in PROM.

Our work is directly inspired by these two schemes that utilize LWE and evasive LWE assumptions. Specifically, we build on their approach to lattice-based iO by relying on the same two assumptions and the transformation in PROM, while also introducing a new assumption—all-product LWE—for our construction. We believe that the aforementioned attacks do not directly apply to our assumption because it does not permit for leakage of noises or private randomness used in encryption. Nevertheless, it is evident that further cryptanalysis is necessary.

2 Technical Overview

2.1 Review of the FE scheme in [AKY24a]

Before introducing our technical ideas, we recall the construction of the FE scheme in [AKY24a]. Their FE scheme employs the BGG+ encoding proposed in [BGG+14]. The encoding of bits $\mathbf{x}$ is defined by $\mathbf{c}_{\mathsf{att}} := \underline{\mathbf{s}^T(\mathbf{A}_{\mathsf{att}} - \mathbf{x}^T \otimes \mathbf{G}_{n+1})}$, where $\mathbf{s} \in \mathbb{Z}_q^{n+1}$ is a secret key, $\mathbf{A}_{\mathsf{att}} \in \mathbb{Z}_q^{(n+1)\times|\mathbf{x}|m}$ is a public matrix with the column size $m = (n+1)\lceil \log_2 q \rceil$, and an underlined term contains LWE errors. Here, $\mathbf{G}_{n+1} \in \mathbb{Z}_q^{(n+1)\times m}$ is a public matrix known as a gadget matrix, defined by $\mathbf{G}_{n+1} = \mathbf{I}_{n+1} \otimes \mathbf{g}^T$ with $\mathbf{g}^T := (1, \ldots, 2^{\lceil \log_2 q \rceil - 1})$. The encoding $\mathbf{c}_{\mathsf{att}}$ can be decomposed into an encoding of each encoded bit, namely $\mathbf{c}_{\mathsf{att},i} := \underline{\mathbf{s}^T(\mathbf{A}_{\mathsf{att},i} - x_i\mathbf{G}_{n+1})}$ for each bit x_i of $\mathbf{x}$.

[2] While the first construction in [BDGM20b] still relies on the decisional composite residuosit (DCR) assumption for the LHE scheme, which is not post-quantum secure, the construction in Subsection 4.4 of [BDGM20b] are based on LWE, which removes the DCR assumption.

Notably, one can homomorphically compute the encodings corresponding to addition and multiplication of bits of two input encodings, respectively, as follows:

$$\begin{aligned}\mathbf{c}_{\mathsf{att},x_u+x_v} &:= \underline{\mathbf{s}^T((\mathbf{A}_{\mathsf{att},x_u}+\mathbf{A}_{\mathsf{att},x_v})-(x_u+x_v)\mathbf{G}_{n+1})}\\ &= \mathbf{c}_{\mathsf{att},x_u}+\mathbf{c}_{\mathsf{att},x_v},\\ \mathbf{c}_{\mathsf{att},x_ux_v} &:= \underline{\mathbf{s}^T(\mathbf{A}_{\mathsf{att},x_u}\mathbf{G}_{-1}^{-1}(\mathbf{A}_{\mathsf{att},x_v})-x_ux_v\mathbf{G})}\\ &= \mathbf{c}_{\mathsf{att},x_u}\mathbf{G}_{n+1}^{-1}(\mathbf{A}_{\mathsf{att},x_v})+x_u\mathbf{c}_{\mathsf{att},x_v},\end{aligned}$$

where for any matrix $\mathbf{A}$, $\mathbf{G}_{n+1}^{-1}(\mathbf{A})$ is defined as a low-norm matrix such that $\mathbf{G}_{n+1}\mathbf{G}_{n+1}^{-1}(\mathbf{A}) = \mathbf{A}$ holds. By repeating these operations according to every gate in a **public** matrix-valued circuit $C:\{0,1\}^L\to\mathbb{Z}_q^{n\times m'}$, one can compute the encoding of the circuit output $C(\mathbf{x})\in\mathbb{Z}_q^{m'}$, defined by $\mathbf{c}_{\mathsf{att},C(\mathbf{x})} := \underline{\mathbf{s}^T(\mathbf{A}_{\mathsf{att},C}-C(\mathbf{x}))}$ [BTVW17]. Furthermore, the matrix $\mathbf{A}_{\mathsf{att},C}$ in the first term after evaluating C depends only on C and is independent of the input $\mathbf{x}$. A trusted party can leverage this property to ensure that an untrusted party can obtain non-trivial information only when evaluating a permitted circuit on the given encodings as described later.

However, the primitive implementation of the BGG+ encoding described above cannot handle private inputs because the encoded bit x_i needs to be public if it is used in the multiplication.[3] A dual-use technique proposed in [BTVW17] overcomes this limitation. Specifically, a trusted party who knows both $\mathbf{s}$ and private input bits $\mathbf{x}$ encrypts $\mathbf{x}$ under the public key of the fully homomorphic encryption (FHE) scheme, obtaining a FHE ciphertext denoted by $\mathbf{X}$. Notably, the technique requires using the same private key $\mathbf{s}$ both for the BGG+ encoding and for the FHE decryption. The trusted party additionally releases the encoding of $\mathsf{bits}(\mathbf{X})$, defined by $\mathbf{c}_{\mathsf{att},\mathbf{X}} = \underline{\mathbf{s}^T(\mathbf{A}_{att}-(1,\mathsf{bits}(\mathbf{X}))\otimes\mathbf{G}_{n+1})}$. Since the ciphertext $\mathbf{X}$ itself can be public, an untrusted party can compute a circuit C on input $\mathsf{bits}(\mathbf{X})$ over the encoding. This implies that if the circuit C computes an homomorphic evaluation algorithm of the FHE scheme, one can obtain the encoding of the FHE-homomorphic evaluation result of another circuit f.

The FHE ciphertext is automatically decrypted inside the encoding, and the decryption result is masked by the term $\mathbf{s}^T\mathbf{A}_{\mathsf{att},C}$. In lattice-based FHE schemes such as [GSW13], multiplication between the FHE ciphertext and the secret key $\mathbf{s}$ yields the decryption result to which some LWE errors are added. Therefore, the dual-use of the secret key $\mathbf{s}$ automatically converts the the second term in the output encoding corresponds to the plaintext of the circuit output plus the LWE errors $\mathbf{e}$, i.e., $\mathbf{s}^T\mathbf{Y} = \mathbf{y}^T+\mathbf{e}^T$ for the FHE encryption $\mathbf{Y}$ of the output $\mathbf{y}$. This decryption result can be recovered from the encoding if the mask $\mathbf{s}^T\mathbf{A}_{\mathsf{att},C}$ is available as a distinct term.

The FE scheme in [AKY24a] employs the BGG+ encoding enhanced by the dual-use technique to ensure that a functional secret key sk_f allows an untrusted decryptor to obtain only the output of a permitted **public** circuit f on the private input under the FE ciphertext. Specifically, the FE ciphertext ct for the input $\mathbf{x}$ contains the encoding of the FHE ciphertext $\mathbf{c}_{\mathsf{att},\mathbf{X}}$. A trusted party in the FE scheme outputs sk_f in advance, allowing the decryptor to produce the mask $\mathbf{s}^T\mathbf{A}_{\mathsf{att},C}$ but only for the circuit C corresponding to the homomorphic evaluation of f on the FHE ciphertext. This is possible for the trusted party because the matrix $\mathbf{A}_{\mathsf{att},C}$ depends only on the circuit C and is independent of the input $\mathsf{bits}(\mathbf{X})$ as described above. Using them, the decryptor can obtain the output of the circuit f on the input $\mathbf{x}$ by decrypting the FE ciphertext ct with sk_f—i.e., evaluating C on $\mathbf{c}_{\mathsf{att},\mathbf{X}}$ to obtain $\underline{\mathbf{s}^T\mathbf{A}_{\mathsf{att},C}-f(\mathbf{x})}$ and then removing the first term with sk_f.

Using a technique similar to that of [AJ15], this FE scheme can be extended to iO for pseudorandom functionalities as shown in [AKY24b]. Branco et al. [BDJ+24] also introduced a similar iO concept and presented a generic compiler that transforms iO for pseudorandom functionalities into iO for general circuits under PROM. Combining these works, we can construct iO for general circuits from the FE

[3] If the encoded bit x_i is multiplied only once with a public input, it can remain private because the multiplication requires only the bit of the left-side input is public. In fact, the work in [GVW15] as well as our scheme utilize this property to evaluate an inner product between public and private vectors on the BGG+ encodings.

scheme in [AKY24a]. However, this combination still suffers from the costly FE recursive encryption process.

2.2 Our initial insight

Our initial insight into the iO construction without the reliance on the existing bootstrapping techniques from FE to iO is this: why must we run the entire FE encryption algorithm again just to produce a fresh ciphertext from an existing one? In fact, if we can allow an evaluator to directly alter the input under the FE ciphertext provided by an obfuscator, but only in some restricted manner, we can straightforwardly construct iO from FE as follows:

1. The obfuscator provides an FE encryption of a **private** circuit C being obfuscated, defined by $\mathsf{ct}_C := \mathsf{FE.Enc}(C)$, where a public key of the FE scheme is omitted for simplicity.
2. The obfuscator also provides a functional secret key sk_f for a function that takes as input C and the evaluator's input $\mathbf{x}$ and outputs $U(C, \mathbf{x}) = C(\mathbf{x})$, where U is a universal circuit.
3. The evaluator with the input $\mathbf{x} \in \{0,1\}^L$ alters ct_C into $\mathsf{ct}_{C,\mathbf{x}} := \mathsf{FE.Enc}((C, \mathbf{x}))$.
4. The evaluator finally decrypts $\mathsf{ct}_{C,\mathbf{x}}$ with sk_f, obtaining $C(\mathbf{x})$.

Therefore, the problem we should solve is how to concretely realize the third step.

One of straightforward but inefficient solution is to generate functional secret keys for all 2^L input patterns, where L is the input bit size. Specifically, we define a circuit $U_{\mathbf{x}}(\cdot) := U(\cdot, \mathbf{x})$ for every input $\mathbf{x} \in \{0,1\}^L$ and include $\{\mathsf{sk}_{U_{\mathbf{x}}}\}_{\mathbf{x}\in\{0,1\}^L}$ in the obfuscated circuit. While it allows the evaluator to directly decrypt the FE encryption of C with arbitrary input, the size of the obfuscated circuit and the obfuscator's running time increase in the order $\mathcal{O}(2^L)$, which is infeasible.

2.3 Merging public matrices

To address the inefficiency of the above solution, we adopt an essence of the existing bootstrapping techniques from FE to iO. Specifically, we put each input bit one-by-one into the encoding. Hereafter, we utilize the structure of the FE scheme [AKY24a] in a non-black-box manner.

The obfuscator provides the initial BGG+ encoding

$$\mathbf{c}_{\mathsf{att},\epsilon}^T := \underline{\mathbf{s}^T(\mathbf{A}_{\mathsf{att},0} - (1, \mathsf{bits}(\mathbf{X}), \mathbf{0}_L) \otimes \mathbf{G}_{n+1})},$$

where $\mathbf{X} := \mathsf{FHE.Enc}(\mathbf{s}, (C, K_B, K_H))$, and the sequence of 0 denoted by $\mathbf{0}_L$ will be altered into the input bits $\mathbf{x}_L$ later.[4]

For every $i \in [L]$ and $b \in \{0,1\}$, we let $\mathsf{ADD}[i,b]$ be a circuit that adds b to the i-th bit of the input. For the first input bit x_1, the evaluator can apply $\mathsf{ADD}[1, x_1]$ to the encoding $\mathbf{c}_{\mathsf{att},\epsilon}$ to obtain the encoding of $(1, \mathsf{bits}(\mathbf{X}), x_1, \mathbf{0}_{L-1})$. The first term in this encoding is expressed as $\mathbf{s}^T\mathbf{A}_{\mathsf{att},\mathsf{ADD}[1,x_1]}$. Unlike the property of the original BGG+ encoding described above, the matrix $\mathbf{A}_{\mathsf{att},\mathsf{ADD}[1,x_1]}$ depends on the input bit x_1. When this process is repeated until the last input bit, even though it allows the evaluator to insert the input bits to the obfuscator's initial encoding, it still requires the obfuscator to perform $\mathcal{O}(2^L)$ amount of work to generate the functional secret keys for all possible input patterns.

Therefore, we consider merging $\mathsf{ADD}[i,0]$ and $\mathsf{ADD}[i,1]$, dependeing on x_i, into a common public matrix $\mathbf{A}_{\mathsf{att},i}$, which does not depend on x_i. As a result, when the circuit $\mathsf{ADD}[i+1, x_{i+1}]$ is applied to the modified encoding, the resulting public matrix depends only on x_{i+1} and no longer on $\mathbf{x}_i := (x_1, \ldots, x_i)$. In order for the evaluator to perform this replacement, for each $i \in \{1, \ldots, L\}$, the obfuscator only needs to include two variants of the functional secret keys that can produce two

[4]The K_B denotes a key of a PRF used to mask the LWE errors after the decryption [AKY24a]. The K_H is a key of another PRF corresponding to pseudorandom oracle [JLLW23], used to extend the functionality of iO to arbitrary circuits [BDJ+24].

masks in the obfuscation: $\mathbf{s}^T(-\mathbf{A}_{\mathsf{att},\mathsf{ADD}[i,0]} + \mathbf{A}_{\mathsf{att},i+1})$ and $\mathbf{s}^T(-\mathbf{A}_{\mathsf{att},\mathsf{ADD}[i,1]} + \mathbf{A}_{\mathsf{att},i+1})$. By adding $\mathbf{s}^T(-\mathbf{A}_{\mathsf{att},\mathsf{ADD}[i,x_i]} + \mathbf{A}_{\mathsf{att},i+1})$ to the encoding after applying $\mathsf{ADD}[i, x_i]$, the evaluator can obtain the following encoding

$$\mathbf{c}^T_{\mathsf{att},\mathbf{x}_i} := \underline{\mathbf{s}^T(\mathbf{A}_{\mathsf{att},i} - (1, \mathsf{bits}(\mathbf{X}), \mathbf{x}_i^T, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1})}.$$

In this manner, this replacement technique can maintain the size of the obfuscated circuit and the obfuscator's running time at $\mathcal{O}(L)$.

Unfortunately, the above construction is insecure. Since the same secret key $\mathbf{s}$ is reused for all encodings, an adversary can remove $\mathbf{s}^T\mathbf{A}_{\mathsf{att},i}$ from two encoding of two distinct encoded bits, allowing the adversary to recover $\mathbf{s}$ by solving an easy LWE problem $\underline{\mathbf{s}G}$ [Wee22].

This suggests that distinct secret keys should be used for all input bits $\mathbf{x}_L$. In other words, the secret key should depend on a trace of input bits that have been inserted. This raises the following two questions:

1. Concretely, how can we switch the secret key dynamically as the input bits are inserted?
2. How can we decrypt the FHE ciphertext on the encoding when the secret key changes dynamically?

2.4 Diamond iO

Solution for the first issue. To solve the first issue, we define a input-dependent secret key for every $i \in [L]$ and every trace of input bits $\mathbf{x}_i$ by

$$\hat{\mathbf{s}}^T_{\mathbf{x}_i} := (\bar{\mathbf{s}}^T \prod_{j\in[i]} \bar{\mathbf{R}}_{x_j}, -1),$$

where $\bar{\mathbf{s}} \leftarrow\!\!\$\ \{0,1\}^n$ and $\bar{\mathbf{R}}_b \leftarrow\!\!\$\ \{0,1\}^{n\times n}$ for every $b \in \{0,1\}$. For $i = 0$, an initial secret key is defined by $\hat{\mathbf{s}}^T_\epsilon := (\bar{\mathbf{s}}^T, -1)$. In the following, we modify the definition of $\mathbf{c}^T_{\mathsf{att},\mathbf{x}_i}$ as follows:

$$\mathbf{c}^T_{\mathsf{att},\mathbf{x}_i} := \underline{\hat{\mathbf{s}}^T_{\mathbf{x}_i}(\mathbf{A}_{\mathsf{att},i} - (1, \mathsf{bits}(\mathbf{X}), \mathbf{x}_i^T, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1})}.$$

This implies that the obfuscator provides the evaluator with the encoding $\mathbf{c}^T_{\mathsf{att},\epsilon}$.

The new definition of the secret key is useful because one can update the key simply by multiplying the current secret key $\hat{\mathbf{s}}^T_{\mathbf{x}_i}$ with a matrix $\bar{\mathbf{R}}_{x_{i+1}}$, provided that $\bar{\mathbf{R}}_0$ and $\bar{\mathbf{R}}_1$ are publicly available. This allows the evaluator to switch the secret key in the second term of $\mathbf{c}^T_{\mathsf{att},\mathbf{x}_i}$ without the obfuscator's help as follows:

$$\begin{aligned}
&\mathbf{c}^T_{\mathsf{att},\mathbf{x}_i}\left(\mathbf{I}_{L'} \otimes \mathbf{G}^{-1}_{n+1}(\mathbf{R}_{x_{i+1}\mathbf{G}_{n+1}})\right) \\
&= \underline{\hat{\mathbf{s}}^T_{\mathbf{x}_i}\mathbf{A}_{\mathsf{att},i}\left(\mathbf{I}_{L'} \otimes \mathbf{G}^{-1}_{n+1}(\mathbf{R}_{x_{i+1}\mathbf{G}_{n+1}})\right) - (1, \mathsf{bits}(\mathbf{X}), \mathbf{x}_i^T, \mathbf{0}_{L-i}) \otimes \hat{\mathbf{s}}^T_{\mathbf{x}_{i+1}}\mathbf{G}_{n+1}},
\end{aligned}$$

where L' is the bit size of $(1, \mathsf{bits}(\mathbf{X}), \mathbf{x}_i^T, \mathbf{0}_{L-i})$, $\mathbf{I}_{L'}$ is an identity matrix with the size $L' \times L'$, and $\mathbf{R}_b$ is defined by $\bar{\mathbf{R}}_b := \begin{pmatrix} \bar{\mathbf{R}}_b & \mathbf{0}_{n\times 1} \\ \mathbf{0}_{1\times n} & 1 \end{pmatrix}$ for every $b \in \{0,1\}$.

We next explain how the evaluator can switch the secret key in the first term. Specifically, we want to replace $\hat{\mathbf{s}}^T_{\mathbf{x}_i}\mathbf{A}_{\mathsf{att},i}\left(\mathbf{I}_{L'} \otimes \mathbf{G}^{-1}_{n+1}(\mathbf{R}_{x_{i+1}\mathbf{G}_{n+1}})\right)$ with $\hat{\mathbf{s}}^T_{\mathbf{x}_{i+1}}\mathbf{A}_{\mathsf{att},i+1}$—the secret key depends on the trace of input bits but the public matrix $\mathbf{A}_{\mathsf{att},i+1}$ does not. The data necessarily to help the evaluator to perform this replacement is provided by the obfuscator as follows:

1. For every $i \in [0, L]$ and $b \in \{0, 1, \star\}$, sample a random matrix $\mathbf{B}_{i,b}$ and its trapdoor $\mathbf{B}^{-1}_{i,b}$.

2. Compute $\mathbf{p}_\epsilon^T := \underline{(\hat{\mathbf{s}}_\epsilon^T, \hat{\mathbf{s}}_\epsilon^T)\mathbf{B}_{0,\star}}$.

3. For every $i \in [L]$ and $b \in \{0,1\}$, sample preimages

$$\mathbf{M}_{i,b} \leftarrow\$ \mathbf{B}_{i-1,\star}^{-1}(\mathbf{U}_b \mathbf{B}_{i,b})$$
$$\mathbf{N}_{i,b} \leftarrow\$ \mathbf{B}_{i,b}^{-1}(\mathbf{U}_\star \mathbf{B}_{i,\star}),$$

where $\mathbf{U}_b := \begin{pmatrix} \mathbf{I}_{n+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_b \end{pmatrix}$ and $\mathbf{U}_\star := \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{n+1} & \mathbf{I}_{n+1} \end{pmatrix}$.

4. For every $i \in [L]$ and $b \in \{0,1\}$, sample a preimage

$$\mathbf{K}_{i,b} \leftarrow\$ \mathbf{B}_{i,b}^{-1} \begin{pmatrix} -\mathbf{A}_{\mathsf{att},i-1}\left(\mathbf{I}_{L'} \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{R}_{x_{i+1}\mathbf{G}_{n+1}})\right) \\ \mathbf{A}_{\mathsf{att},i} - (\mathbf{0}_{L_{\mathsf{fhe}}+i}, b, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1} \end{pmatrix}.^{5}$$

5. Add $\mathbf{p}_\epsilon$ and $\{\mathbf{M}_{i,b}, \mathbf{N}_{i,b}, \mathbf{K}_{i,b}\}_{i\in[L],b\in\{0,1\}}$ to the obfuscated circuit.

We note that for any matrix $\mathbf{A}$, a preimage $\mathbf{K} \leftarrow\$ \mathbf{B}^{-1}(\mathbf{A})$ is a low-norm matrix that satisfies $\mathbf{BK} = \mathbf{A}$; however, the preimage can be sampled only by one who knows a trapdoor $\mathbf{B}^{-1}$ for a uniformly random matrix $\mathbf{B}$ [GPV08, MP12, GM18].

Using the above additional data, the evaluator can replace the first term in $\mathbf{c}_{\mathsf{att},\mathbf{x}_{i-1}}^T$ for every $i \in [L]$ as follows:

1. Compute $\mathbf{q}_{\mathbf{x}_i}^T := \mathbf{p}_{\mathbf{x}_{i-1}}^T \mathbf{M}_{i,x_i} = \underline{(\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T)\mathbf{B}_{i,x_i}}$.

2. Compute $\mathbf{p}_{\mathbf{x}_i}^T := \mathbf{q}_{\mathbf{x}_i}^T \mathbf{N}_{i,x_i} = \underline{(\hat{\mathbf{s}}_{\mathbf{x}_i}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T)\mathbf{B}_{i,\star}}$.

3. Compute

$$\begin{aligned}\mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T &:= \mathbf{q}_{\mathbf{x}_i}^T \mathbf{K}_{i,x_i} \\ &= \underline{-\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T \mathbf{A}_{\mathsf{att},i-1}\left(\mathbf{I}_{L'} \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{R}_{x_{i+1}\mathbf{G}_{n+1}})\right) + \hat{\mathbf{s}}_{\mathbf{x}_i}^T\left(\mathbf{A}_{\mathsf{att},i} - (\mathbf{0}_{L_{\mathsf{fhe}}+i}, b, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1}\right)}.\end{aligned}$$

4. Compute

$$\begin{aligned}\mathbf{c}_{\mathsf{att},\mathbf{x}_i}^T &:= \mathbf{c}_{\mathsf{att},\mathbf{x}_{i-1}}^T \left(\mathbf{I}_{L'} \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{R}_{x_{i+1}}\mathbf{G}_{n+1})\right) + \mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T \\ &= \underline{\hat{\mathbf{s}}_{\mathbf{x}_i}^T \mathbf{A}_{\mathsf{att},i} - (1, \mathsf{bits}(\mathbf{X}), \mathbf{x}_i^T, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1}}.\end{aligned}$$

In this manner, the evaluator can switch the secret key in both terms of $\mathbf{c}_{\mathsf{att},\mathbf{x}_{i-1}}^T$ to $\hat{\mathbf{s}}_{\mathbf{x}_i}^T$. The diamond-shaped key switching operation in our final construction is illustrated in Figure 1. It culminates in two encodings $\mathbf{c}_{\mathsf{att},\mathbf{x}_L}^T$ and $\mathbf{c}_{t,\mathbf{x}_L}^T$, the latter of which is defined later in this subsection. Notably:

- The public keys $\mathbf{A}_{\mathsf{att},L}$ and $\mathbf{A}_{t,L}$ used in these encodinga are independent from the evaluator's input bits $\mathbf{x}_L$.
- The i-th preimages provided by the obfuscator—denoted in red—are independent of the trace of the previous input bits $\mathbf{x}_{i-1}$, maintaining the size of the obfuscated circuit and the obfuscator's running time at $\mathcal{O}(L)$.

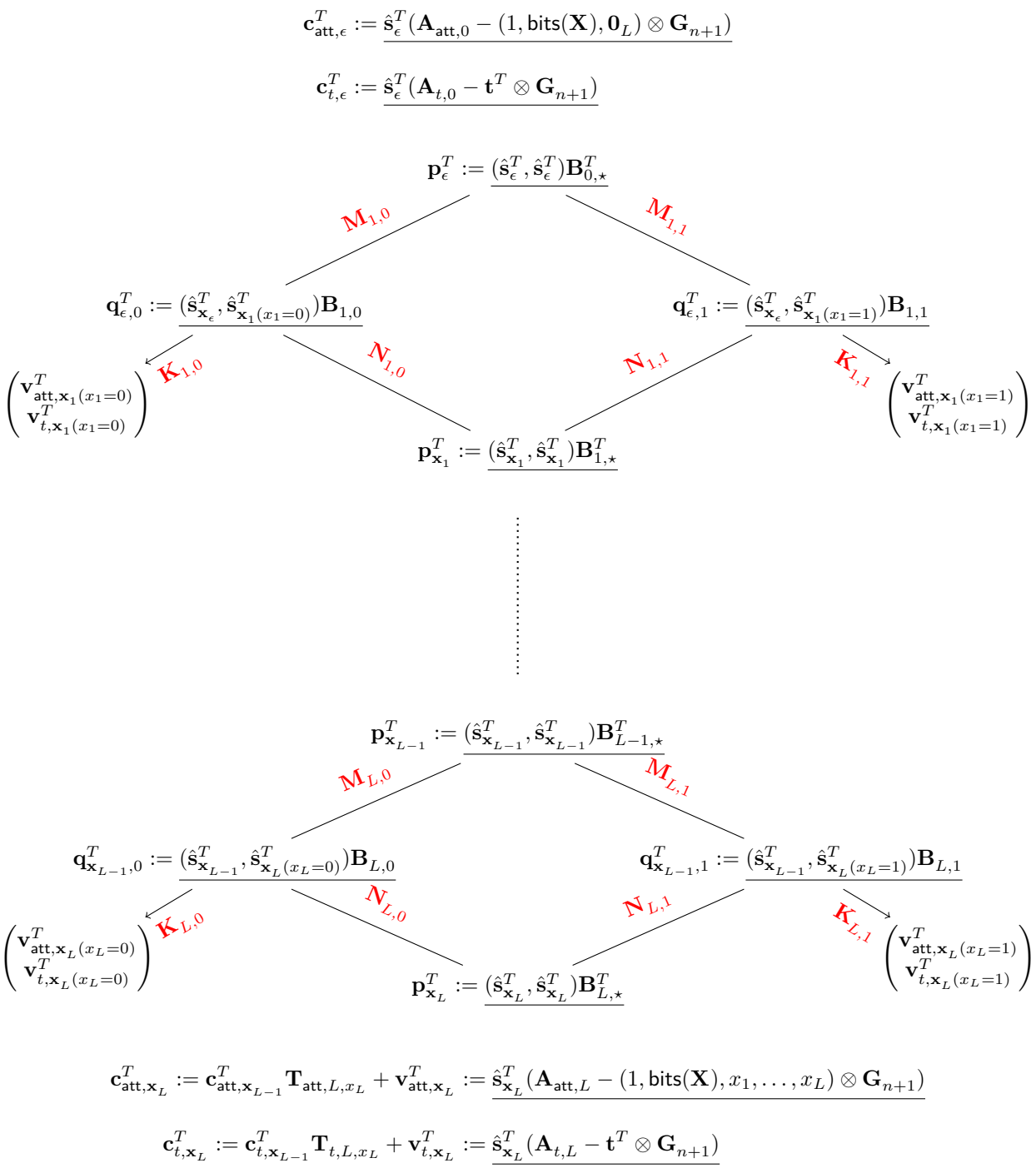

Figure 1: An illustration of the diamond-shaped key switching operation in our final construction.

Since a different secret key $\hat{\mathbf{s}}_{\mathbf{x}_L}$ is associated to each of the possible 2^L encodings, the above attack, which uses the encodings of different bits under the same secret key, seems to be prevented. However, to prove this formally, we need to introduce a new lattice assumption called all-product LWE, claiming the pseudorandomness of the following distribution:

$$\left(\begin{array}{l}\{\mathbf{A}_i\}_{i\in[L]}, \{\mathbf{B}_i\}_{i\in[0,L]}, \{\bar{\mathbf{R}}_b\}_{b\in\{0,1\}},\\ \{\mathbf{c}_{A,\mathbf{x}_i} := \underline{\hat{\mathbf{s}}_{\mathbf{x}_i}^T(\mathbf{A}_i - \left(\mathbf{x}_i^T, \mathbf{0}_{L-i}\right)\otimes \mathbf{G}_{n+1})}\}_{\substack{i\in[L],\\ \mathbf{x}_i\in\{0,1\}^i}}\\ \{\mathbf{c}_{B,\mathbf{x}_i} := \underline{\hat{\mathbf{s}}_{\mathbf{x}_i}^T\mathbf{B}_i}\}_{\substack{i\in[0,L],\\ \mathbf{x}_i\in\{0,1\}^i}}\end{array}\right)$$
$$\approx_c \left(\begin{array}{l}\{\mathbf{A}_i\}_{i\in[L]}, \{\mathbf{B}_i\}_{i\in[0,L]}, \{\bar{\mathbf{R}}_b\}_{b\in\{0,1\}},\\ \{\mathbf{c}_{A,\mathbf{x}_i} \leftarrow\!\$\ \mathbb{Z}_q^{(n+1)\lceil\log_2 q\rceil}\}_{\substack{i\in[L],\\ \mathbf{x}_i\in\{0,1\}^i}}\\ \{\mathbf{c}_{B,\mathbf{x}_i} \leftarrow\!\$\ \mathbb{Z}_q^{m_{B,i}}\}_{\substack{i\in[0,L],\\ \mathbf{x}_i\in\{0,1\}^i}}\end{array}\right),$$

where $\mathbf{A}_i \leftarrow\!\$\ \mathbb{Z}_q^{(n+1)\times(n+1)\lceil\log_2 q\rceil}$ and $\mathbf{B}_i \leftarrow\!\$\ \mathbb{Z}_q^{(n+1)\times m_{B,i}}$ with the column size $m_{B,i}$ given as the parameter. It remains an open problem whether this assumption can be reduced to standard assumptions such as LWE.

Solution for the second issue. To solve the second issue, we forgo the dual-use technique employed in [AKY24a] due to 2^L parrerns of the dynamically changing secret keys used for the encodings. Instead, the obfuscator samples an independent secret key $\mathbf{t}^T = (\bar{\mathbf{t}}^T, -1)$ in the Gentry, Sahai, and Waters (GSW) FHE scheme [GSW13] and uses the corresponding public key $\mathbf{A}_{\mathsf{fhe}}$ to encrypt the private circuit f concatenated with the PRF keys $K_B, K_H \leftarrow\!\$\ \{0,1\}^\lambda$. Additionally, the obfuscator encodes $\mathbf{t}^T$ into an encoding

$$\mathbf{c}_{t,\epsilon}^T := \underline{\hat{\mathbf{s}}_\epsilon^T(\mathbf{A}_{t,0} - \mathbf{t}^T\otimes\mathbf{G}_{n+1})},$$

where $\mathbf{A}_{t,0} \leftarrow\!\$\ \mathbb{Z}_q^{(n+1)\times(n+1)m}$ is another public matrix that does not depend on the input bits. We later explain why it remains secure to directly encode the **private** secret key $\mathbf{t}$ into the encoding without encrypting it under FHE.

Using the encoding $\mathbf{c}_{\mathsf{att},\mathbf{x}_L}$ built as above, the evaluator homomorphically computes the encoding of $\mathbf{Y}$, which is the homomorphic evaluation result of the encrypted circuit $\mathbf{X}$ on the inserted input bits $\mathbf{x}_L$. To allow the evaluator to eventually decrypt $\mathbf{Y}$ by using the encoding of $\mathbf{t}$, the evaluator also key-switches $\mathbf{c}_{t,\epsilon}^T$ in the same manner as $\mathbf{c}_{\mathsf{att},\mathbf{x}_L}^T$.

After that, the evaluator computes an inner product bwteen the encodings of $\mathbf{Y}$ and $\mathbf{t}^T$ under the secret key $\hat{\mathbf{s}}_{\mathbf{x}_L}^T$ by a similar method introduced in [GVW15], producing the encoding of the partial decryption result $\mathbf{y}^T + \mathbf{e}^T$. Specifically, provided the encoding of the k-th bit of the (i,j) entry of $\mathbf{Y}$, defined by $\mathbf{c}_{y_{i,j,k},\mathbf{x}_L} := \underline{\hat{\mathbf{s}}_{\mathbf{x}_L}^T(\mathbf{A}_{\mathsf{att},C,y_{i,j,k}} - y_{i,j,k}\mathbf{G}_{n+1})}$, and the encoding of the i-th element of $\mathbf{t}$, defined by $\mathbf{c}_{t_i,\mathbf{x}_L}^T := \underline{\hat{\mathbf{s}}_{\mathbf{x}_L}^T(\mathbf{A}_{t,L,i} - t_i\mathbf{G}_{n+1})}$, the evaluator can compute the encoding of $y_{i,j,k}t_i$ as follows:

$$\begin{aligned}\mathbf{c}_{y_{i,j,k}t_i,\mathbf{x}_L}^T &:= \mathbf{c}_{y_{i,j,k},\mathbf{x}_L}^T\mathbf{G}_{n+1}^{-1}(\mathbf{A}_{t,L,i}) + \mathbf{c}_{t_i,\mathbf{x}_L}^T(y_{i,j,k}\mathbf{I}_m)\\ &= \underline{\hat{\mathbf{s}}_{\mathbf{x}_L}^T(\mathbf{A}_{\mathsf{att},C,y_{i,j,k}}\mathbf{G}_{n+1}^{-1}(\mathbf{A}_{t,L,i}) - y_{i,j,k}t_i\mathbf{G}_{n+1})}.\end{aligned}$$

Notably, since the encoded bit needs to be public only for the left-side input, the encoding $\mathbf{c}_{t_i,\mathbf{x}_L}^T$ can remain t_i private without FHE. Additionally, we can observe that

$$\mathbf{t}^T\mathbf{Y} = \sum_{k\in\{1,\ldots,\lceil\log_2 q\rceil\}} 2^{k-1}\sum_{j\in\{1,\ldots,\ell\}}\Big(\sum_{i\in\{1,\ldots,n+1\}} t_i y_{i,j,k}\Big)\mathbf{u}_{\ell,j}$$

holds, where $\mathbf{u}_{\ell,j}$ is the j-th unit vector of length ℓ. Therefore, by homomorphicallyc computing the linear combinations of the encodings of $y_{i,j,k}t_i$, the evaluator can obtain the encoding of $\mathbf{t}^T\mathbf{Y}$, i.e., $\mathbf{c}^T_{F,\mathbf{x}_L} := \hat{\mathbf{s}}^T_{\mathbf{x}_L}\mathbf{A}_F - (\mathbf{y}^T + \mathbf{e}^T)$. The public matrix $\mathbf{A}_F$ corresponds to the computation of the above homomorphic evaluation and partial decryption, and does not depend on the input $\mathbf{x}_L$.

The evaluator's final task is to recover the decryption result by removing the mask $\hat{\mathbf{s}}^T_{\mathbf{x}_L}\mathbf{A}_F$. To do that, the obfuscator additionally includes a preimage

$$\mathbf{K}_F \leftarrow\!\!\$\ \mathbf{B}_F^{-1}\begin{pmatrix}\mathbf{A}_F\\ \mathbf{0}\end{pmatrix}.$$

This allows the evaluator to compute $\mathbf{v}^T_{F,\mathbf{x}_L} := \mathbf{p}^T_{\mathbf{x}_L}\mathbf{K}_F = \hat{\mathbf{s}}^T_{\mathbf{x}_L}\mathbf{A}_F$. Hence, the evaluator can remove it from $\mathbf{c}^T_{F,\mathbf{x}_L}$, obtaining the evaluation result $f(\mathbf{x}_L)$.

3 Preliminaries

3.1 Notations

For non-negative integers $i, j \in \mathbb{Z}$ such that $i \le j$ holds, $[i, j]$ denotes the set $\{i, i+1, \ldots, j\}$. For simplicity, $[n]$ and $[0]$ represent the sets $\{1, \ldots, n\}$ and $\emptyset$, respectively.

A vector and a matrix are denoted by bold letters, such as $\mathbf{a}$ and $\mathbf{A}$. Unless otherwise specified, we define vectors as column vectors. For any vector $\mathbf{a}$, the size of $\mathbf{a}$ is denoted by $|\mathbf{a}|$, and the i-th element of $\mathbf{a}$ is denoted by a_i for every $i \in [|\mathbf{a}|]$. For any matrix $\mathbf{A}$ with n rows and m columns, let $a_{i,j}$ denote the (i,j)-th entry of $\mathbf{A}$, for every $i \in [n]$ and $j \in [m]$. The ℓ_∞-norm of the vector $\mathbf{a}$ is defined by $\|a\|_\infty := \max_{i\in[|\mathbf{a}|]}\{|a_i|\}$. Similarly, that of the matrix $\mathbf{A}$ with n rows and m columns is defined by $\|a\|_\infty := \max_{i\in[n]}\{\Sigma_{j\in[m]}|a_{i,j}|\}$. For a row vector $\mathbf{a}^T \in \mathbb{Z}_q^{1\times n}$, it holds that $\left\|\mathbf{a}^T\right\|_\infty \le n\|\mathbf{a}\|_\infty$. The ℓ_2-norm of the n-sized vector $\mathbf{a}$ is defined by $\|a\|_2 := \sqrt{\Sigma_{i\in[|\mathbf{a}|]}a_i^2}$. For any $n \in \mathbb{N}$ and $i \in [n]$, $\mathbf{u}_{n,i} \in \mathbb{Z}_q^n$ is an unit vector whose i-th entry is 1 and the others are 0.

Let $\mathbb{Z}_q$ denote the ring of integers modulo q. For an integer $x \in \mathbb{Z}_q$, define $\mathsf{bits}(x) := (x_1, \ldots, x_{\lceil\log_2 q\rceil}) \in \{0,1\}^{1\times\lceil\log_2 q\rceil}$ as the vector of bits corresponding to the binary representation of x in little-endian form—i.e., $x = \Sigma_{i\in[\lceil\log_2 q\rceil]}x_i 2^{i-1}$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathsf{bits}(\mathbf{A}) \in \{0,1\}^{nm\lceil\log_2 q\rceil}$ is similarly defined as below:

$$\mathsf{bits}(\mathbf{A}) := (\mathsf{bits}(a_{1,1}), \ldots, \mathsf{bits}(a_{n,1}), \ldots, \mathsf{bits}(a_{1,n}), \ldots, \mathsf{bits}(a_{n,n}))^T$$

For any distribution $\mathcal{X}$, $x \leftarrow\!\!\$\ \mathcal{X}$ indicates that x is sampled from $\mathcal{X}$. Let $x \leftarrow\!\!\$\ \mathbb{Z}_q$ denote that x is sampled from a unifrom distribution over $\mathbb{Z}_q$.

Let λ be a security parameter. A function $\mathsf{negl}(\lambda) : \mathbb{N} \to \mathbb{R}$ is called negligible, if for all constants $c > 0$, there exists $n \in \mathbb{N}$ such that $\mathsf{negl}(\lambda) < \lambda^{-c}$ holds for all $\lambda > n$.

3.2 Lattices and Gaussians

We recall some facts on lattices and gaussian distributions.

Lattices. Fix some integers $n, m, q \in \mathbb{N}$. For a full-rank integer matrix $\mathbf{B} \in \mathbb{Z}^{n\times m}$, an integer lattice is defined by $\Lambda(\mathbf{B}) := \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^m\}$. For any matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a q-ary orthogonal lattice is defined by $\Lambda^\perp(\mathbf{A}) := \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{Ax} = \mathbf{0} \mod q\}$. For any vector $\mathbf{u} \in \mathbb{Z}_q^n$, its coset is defined by $\Lambda^\perp_{\mathbf{u}}(\mathbf{A}) := \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{Ax} = \mathbf{u} \mod q\}$.

Gadget vector and matrix. A gadget vector $\mathbf{g}$ and a gadget matrix $\mathbf{G}_n$ are defined as follows, respectively:

$$\mathbf{g}^T := (2^0, 2^1, \ldots, 2^{\lceil\log_2 q\rceil-1}) \in \mathbb{Z}_q^{\lceil\log_2 q\rceil}$$
$$\mathbf{G}_n := \mathbf{I}_n \otimes \mathbf{g}^T \in \mathbb{Z}_q^{n\times n\lceil\log_2 q\rceil}$$

For any vector $\mathbf{a} \in \mathbb{Z}_q^n$, $\mathbf{G}_n^{-1}(\mathbf{a}) \in \Lambda_{\mathbf{a}}^{\perp}(\mathbf{G}_n)$ is defined as below:

$$\mathbf{G}_n^{-1}(\mathbf{a}) := (\mathsf{bits}(a_1), \ldots, \mathsf{bits}(a_n))^T \in \mathbb{Z}_q^{n\lceil \log_2 q \rceil}$$

This is straightforwardly extended to a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ as $\mathbf{G}_n^{-1}(\mathbf{A}) := (\mathbf{G}_n^{-1}(\mathbf{a}_1), \ldots, \mathbf{G}_n^{-1}(\mathbf{a}_m)) \in \mathbb{Z}_q^{n\lceil \log_2 q \rceil \times m}$, which satisfies $\mathbf{G}_n \mathbf{G}_n^{-1}(\mathbf{A}) = \mathbf{A}$.

Gaussian distributions. We adopt the definition of the discrete Gaussian distribution from [GPV08, BDJ+24]. For any lattice $\Lambda \subseteq \mathbb{R}^b$ and any Gaussian parameter $\sigma > 0$, the discrete Gaussian distribution $\mathcal{D}_{\Lambda,\sigma}$ is defined by the discrete probability distribution via the probability mass function

$$\Pr[\mathbf{e} = \mathbf{x}] \ := \ \begin{cases} \frac{\rho_\sigma(\mathbf{x})}{\rho_\sigma(\Lambda)} & \text{if } \mathbf{x} \in \Lambda, \\ 0 & \text{otherwise.} \end{cases}$$

for an $\mathbf{e} \sim \mathcal{D}_{\Lambda,\sigma}$, where $\rho_\sigma : \mathbb{R}^n \to \mathbb{R}_{\leq 0}$ is the Gaussian function defined by $\rho_\sigma(\mathbf{x}) := \exp(-\pi \|\mathbf{x}\|_2^2 / \sigma^2)$. Here, $\rho_\sigma(\Lambda) := \Sigma_{\mathbf{y} \in \Lambda} \ \rho_\sigma(\mathbf{y})$ is finite, ensuring that $\Pr[\mathbf{e} = \mathbf{x}]$ is well-defined as a probability mass function.

The following lemma bounds the tail probability of the discrete Gaussian distribution [MP12, BDJ+24].

Lemma 1 (Tail Bound of the Discrete Gaussian Distribution [MP12, BDJ+24])**.** For any Gaussian parameter $\sigma > 0$, $\lambda \in \mathbb{N}$, and $e \sim \mathcal{D}_{\mathbb{Z},\sigma}$, it holds that

$$\Pr\Big[e \geq \sqrt{\lambda}\sigma\Big] \leq 2\exp(-\pi\lambda) \leq \widetilde{O}\big(2^{-\lambda}\big)$$

We adopt the smudging lemma from [WWW22, AKY24a].

Lemma 2 (Smudging Lemma [WWW22, AKY24a])**.** Let λ be a security parameter. Take any $a \in \mathbb{Z}$ where $|a| \leq B$. Suppose $\sigma \geq B\lambda^{\omega(1)}$. Then the statistical distance between the distributions $\{e \mid e \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\sigma}\}$ and $\{e + a \mid e \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\sigma}\}$ is $\mathsf{negl}(\lambda)$.

Trapdoor and preimage sampling. We adopt the definitions and syntax of trapdoor and preimage sampling algorithms from [AKY24a]. For integers $n, q \in \mathbb{N}$, let $m := \mathcal{O}(n \log q)$ and $\gamma := \omega(\sqrt{n \log q \log m})$. Previous works [GPV08, MP12, GM18] propose an algorithm $\mathsf{TrapGen}(1^n, 1^m, q)$ that outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ which is 2^{-n} close to unifrom, and a γ-trapdoor for $\mathbf{A}$ denoted by $\mathbf{A}_\gamma^{-1}$.

The trapdoor allows efficient sampling of a preimage for a target vector $\mathbf{u} \in \mathbb{Z}_q^n$ from the discrete Gaussian distribution over $\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A})$, namely $\mathcal{D}_{\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A}),\gamma}$. To sample a preimage for a matrix $\mathbf{U} \in \mathbb{Z}_q^{n \times l}$, we sample l preimages, one for each column of $\mathbf{U}$—that is,

$$\mathbf{A}_\gamma^{-1}(\mathbf{U}) := (\mathbf{A}_\gamma^{-1}(\mathbf{u}_1), \ldots, \mathbf{A}_\gamma^{-1}(\mathbf{u}_l)),$$

where $\mathbf{u}_i$ denotes the i-th column of $\mathbf{U}$.

3.3 Hardness Assumptions

We rely on the following hardness assumptions.

Assumption 1 (Learning With Errors (LWE) [Reg09]). Let $n, m, q \in \mathbb{N}$ be parameters determinted by $\mathsf{poly}(\lambda)$, and χ be a Gaussian parameter. The LWE assumption is said to be hard if for every probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, it holds that

$$\big|\Pr\big[\mathcal{A}(\mathbf{A}, \mathbf{s}^T\mathbf{A} + \mathbf{e}^T) = 1\big] \ - \ \Pr\big[\mathcal{A}(\mathbf{A}, \mathbf{u}^T) = 1\big]\big| \ \leq \ \mathsf{negl}(\lambda)$$

where $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi}^m$, $\mathbf{u} \leftarrow\!\$\ \mathbb{Z}_q^m$, and the probability is taken over the randomness of $\mathcal{A}$.

We adopt a variant of the evasive LWE assumption defined in [AKY24b], called "non-uniform κ-evasive LWE assumption [AKY24b]". However, unlike their definition, our definition assumes that an adversary can access to a public matrix $\mathbf{P}$. This is because Brzuska et al. [BÜW24] show that the adversary who can see only another public matrix $\mathbf{B}$ can break the security of the evasive LWE with a non-negligible advantage in the private-coin setting. To fix that vulnerability, they introduced a definition, called "private-coin binding evasive LWE [BÜW24]", which includes $\mathbf{P}$ in the adversary's view.

Assumption 2 (Private-coin Binding Non-Uniform κ-Evasive LWE [AKY24b, BÜW24]). Let $n, m, t, m', q \in \mathbb{N}$ be parameters and λ be a security parameter. Let χ and χ' be Gaussian parameters. Let Samp be a PPT algorithm that outputs

$$\mathbf{S} \in \mathbb{Z}_q^{m' \times n}, \mathbf{P} \in \mathbb{Z}_q^{n \times t}, \mathsf{aux} \in \{0,1\}^*$$

on input 1^λ. For non-uniform adversaries $\mathcal{A}_0 = \{\mathcal{A}_{0,\lambda}\}_\lambda$, $\mathcal{A}_1 = \{\mathcal{A}_{1,\lambda}\}_\lambda$, we define the following advantage functions:

$$\begin{aligned}
\mathrm{Adv}^{\mathrm{PRE}}_{\mathcal{A}_0}(\lambda) &:= \Pr\left[\mathcal{A}_0\left(\mathbf{B}, \mathbf{P}, \mathbf{SB}+\mathbf{E}, \mathbf{SP}+\mathbf{E}', \mathsf{aux}\right) = 1\right] \\
&\quad - \Pr\left[\mathcal{A}_0\left(\mathbf{B}, \mathbf{P}, \mathbf{C}_0, \mathbf{C}', \mathsf{aux}\right) = 1\right], \\
\mathrm{Adv}^{\mathrm{POST}}_{\mathcal{A}_1}(\lambda) &:= \Pr\left[\mathcal{A}_1\left(\mathbf{B}, \mathbf{P}, \mathbf{SB}+\mathbf{E}, \mathbf{K}, \mathsf{aux}\right) = 1\right] \\
&\quad - \Pr\left[\mathcal{A}_1\left(\mathbf{B}, \mathbf{P}, \mathbf{C}_0, \mathbf{K}, \mathsf{aux}\right) = 1\right],
\end{aligned}$$

where

$$\begin{aligned}
&(\mathbf{S}, \mathbf{P}, \mathsf{aux}) \leftarrow\!\!\$\ \mathsf{Samp}(1^\lambda), \mathbf{B} \leftarrow\!\!\$\ \mathbf{Z}_q^{n \times m}, \\
&\mathbf{C}_0 \leftarrow\!\!\$\ \mathbf{Z}_q^{m' \times m}, \mathbf{C}' \leftarrow\!\!\$\ \mathbf{Z}_q^{m' \times t}, \\
&\mathbf{E} \leftarrow\!\!\$\ \mathcal{D}_{\mathbb{Z},\chi}^{m' \times m}, \mathbf{E}' \leftarrow\!\!\$\ \mathcal{D}_{\mathbb{Z},\chi'}^{m' \times t}, \\
&\mathbf{K} \leftarrow\!\!\$\ \mathbf{B}_\gamma^{-1}(\mathbf{P}),
\end{aligned}$$

and $\gamma = \mathcal{O}(\sqrt{m \log q})$.

For a function $\kappa = \kappa(\lambda)$ of the security parameter λ, the private-coin binding non-uniform κ-evasive LWE assumption is said to hold, if for every non-uniform sampler Samp and every non-uniform adversary $\mathcal{A}_1$ such that $\mathrm{Size}(\mathsf{Samp}) \le \mathsf{poly}(\lambda')$ and $\mathrm{Size}(\mathcal{A}_1) \le \mathsf{poly}(\kappa)$ for $\lambda'(\lambda) \le \kappa(\lambda)$, there exists another non-uniform adversary $\mathcal{A}_0$ and a polynomial $Q(\cdot)$ such that the following relations hold:

$$\begin{aligned}
\mathrm{Adv}^{\mathrm{PRE}}_{\mathcal{A}_0}(\lambda) &\ge \mathrm{Adv}^{\mathrm{POST}}_{\mathcal{A}_1}(\lambda)/Q(\lambda') - \mathrm{negl}(\kappa), \\
\mathrm{Size}(\mathcal{A}_0) &\le Q(\lambda') \cdot \mathrm{Size}(\mathcal{A}_1).
\end{aligned}$$

Additionally, we adopt Lemma 2.5 from [AKY24b] with necessarily modifications. This distinguishes between the portion of the auxiliary information that depends on the secret $\mathbf{S}$ and the portion that does not, and claims that if the former portion in the precondition is pseudorandom, the same holds in the postcondition.

Lemma 3. Let $n, m, t, m', q \in \mathbb{N}$ be parameters and λ be a security parameter. Let χ and χ' be Gaussian parameters. Let Samp be a PPT algorithm that takes as input 1^λ and outputs

$$\mathbf{S} \in \mathbb{Z}_q^{m' \times n}, \mathbf{P} \in \mathbb{Z}_q^{n \times t}, \mathsf{aux} = (\mathsf{aux}_1, \mathsf{aux}_2) \in \mathcal{S} \times \{0,1\}^*$$

for some set $\mathcal{S}$. Furthermore, we assume that there exists a public deterministic polynomial-time algorithm $\mathsf{Reconstruct}$ that allows to derive $\mathbf{P}$ from aux_2, i.e., $\mathbf{P} = \mathsf{Reconstruct}(\mathsf{aux}_2)$. For a non-

uniform adversary $\mathcal{A}$, we define the following advantage functions:

$$\begin{aligned}\mathrm{Adv}^{\mathrm{PRE}}_{\mathcal{A}}(\lambda) &:= \Pr\left[\mathcal{A}\left(\mathbf{B},\, \mathbf{P},\, \mathbf{SB}+\mathbf{E},\, \mathbf{SP}+\mathbf{E}',\, \mathsf{aux}_1,\, \mathsf{aux}_2\right)=1\right] \\ &\quad -\ \Pr\left[\mathcal{A}\left(\mathbf{B},\, \mathbf{P},\, \mathbf{C}_0,\, \mathbf{C}',\, \mathbf{c},\, \mathsf{aux}_2\right)=1\right], \\ \mathrm{Adv}^{\mathrm{POST}}_{\mathcal{A}}(\lambda) &:= \Pr\left[\mathcal{A}\left(\mathbf{B},\, \mathbf{P},\, \mathbf{SB}+\mathbf{E},\, \mathbf{K},\, \mathsf{aux}_1,\, \mathsf{aux}_2\right)=1\right] \\ &\quad -\ \Pr\left[\mathcal{A}\left(\mathbf{B},\, \mathbf{P},\, \mathbf{C}_0,\, \mathbf{K},\, \mathbf{c},\, \mathsf{aux}_2\right)=1\right],\end{aligned}$$

where

$$\begin{aligned}&(\mathbf{S}, \mathbf{P}, \mathsf{aux} = (\mathsf{aux}_1, \mathsf{aux}_2)) \leftarrow\!\!\$\ \mathsf{Samp}(1^\lambda), \mathbf{B} \leftarrow\!\!\$\ \mathbf{Z}_q^{n\times m}, \\ &\mathbf{C}_0 \leftarrow\!\!\$\ \mathbf{Z}_q^{m'\times m}, \mathbf{C}' \leftarrow\!\!\$\ \mathbf{Z}_q^{m'\times t}, \mathbf{c} \leftarrow\!\!\$\ \mathcal{S}, \\ &\mathbf{E} \leftarrow\!\!\$\ \mathcal{D}_{\mathbb{Z},\chi}^{m'\times m}, \mathbf{E}' \leftarrow\!\!\$\ \mathcal{D}_{\mathbb{Z},\chi'}^{m'\times t}, \mathbf{K} \leftarrow\!\!\$\ \mathbf{B}_\gamma^{-1}(\mathbf{P}),\end{aligned}$$

and $\gamma = \mathcal{O}(\sqrt{m \log q})$.

Then, for a function $\kappa := \kappa(\lambda)$ of the security parameter λ, under the private-coin binding non-uniform κ-evaisve LWE assumption (Assumption 2), if $\mathrm{Size}(\mathsf{Samp}) \le \mathsf{poly}(\lambda')$ and $\mathrm{Size}(\mathcal{A}_1) \le \mathsf{poly}(\kappa)$ for $\lambda' \le \kappa(\lambda)$, there exists another non-uniform adversary $\mathcal{A}_0$ and a polynomial $Q(\cdot)$ such that the following relations hold:

$$\begin{aligned}\mathrm{Adv}^{\mathrm{PRE}}_{\mathcal{A}_0}(\lambda) &\ \ge\ \mathrm{Adv}^{\mathrm{POST}}_{\mathcal{A}_1}(\lambda)/Q(\lambda') \ -\ \mathrm{negl}(\kappa), \\ \mathrm{Size}(\mathcal{A}_0) &\ \le\ Q(\lambda') \ \cdot\ \mathrm{Size}(\mathcal{A}_1).\end{aligned}$$

Proof. The only difference between our lemma and the original lemma in [AKY24b] is that $\mathbf{P}$ is added to the adversaries' views. Since the proof of the original lemma does not depend on the presence of $\mathbf{P}$, their proof can be directly applied to the proof of our lemma. □

3.4 GSW Homomorphic Encryption

Our iO construction relies on the leveled homomorphic encryption scheme proposed by Gentry, Sahai, and Waters (GSW) [GSW13]. We adopt its syntax and properties from [HLL23] with necessarily modifications.

Lemma 4 (GSW Leveled FHE [GSW13, HLL23])**.** The GWE leveled FHE scheme works as follows:

- The keys are

$$\begin{aligned}&\text{(public)} \quad \mathbf{A}_{\mathsf{fhe}} \ =\ \begin{pmatrix}\bar{\mathbf{A}}_{\mathsf{fhe}} \\ \bar{\mathbf{s}}^T\bar{\mathbf{A}}_{\mathsf{fhe}} + \mathbf{e}_{\mathsf{fhe}}^T\end{pmatrix} \ \in\ \mathbb{Z}_q^{(n+1)\times m}, \\ &\text{(secret)} \quad \mathbf{s}^T \ =\ \left(\bar{\mathbf{s}}^T,\, -1\right) \ \in\ \mathbb{Z}_q^{n+1},\end{aligned}$$

  where $\bar{\mathbf{s}} \in \mathbb{Z}_q^n$, $\bar{\mathbf{A}}_{\mathsf{fhe}} \in \mathbb{Z}_q^{n\times m}$, $\bar{\mathbf{e}}_{\mathsf{fhe}} \in \mathbb{Z}^m$.

- A ciphertext of $x \in \{0,1\}$ is

$$\mathbf{X} \ =\ \mathbf{A}_{\mathsf{fhe}}\mathbf{R} - x\mathbf{G}_{n+1} \ \in\ \mathbb{Z}_q^{(n+1)\times m},$$

  where $\mathbf{R} \in \mathbb{Z}^{m\times m}$ is the encryption randomness. The decryption equation is

$$\mathbf{s}^T\mathbf{X} \ =\ -\mathbf{e}_{\mathsf{fhe}}^T\mathbf{R} - x\mathbf{s}^T\mathbf{G} \ \in\ \mathbb{Z}_q^m,$$

  which can be used to extract x via multiplication by $\mathbf{G}^{-1}(\lceil q/2 \rceil \mathbf{u}_{n+1,n+1})$.

- There is an efficient algorithm

$$\mathsf{MakeVEvalCkt}(n, m, q, C) \;=\; \mathsf{VEval}_C$$

that takes as input n, m, q and a vector-valued circuit

$$C : \{0,1\}^L \;\to\; \mathbb{Z}_q^{1\times m'}$$

and outputs a circuit

$$\mathsf{VEval}_C\big(\mathbf{X}_1, \ldots, \mathbf{X}_L\big) \;=\; \mathbf{C}$$

taking L ciphertexts as input and outputting a new ciphertext $\mathbf{C}$ of different format.

  - The depth of VEval_C is $\big(d\,\mathcal{O}(\log m \log\log q) + \mathcal{O}(\log^2 \log q)\big)$ for a circuit C of depth d.
  - Suppose $\mathbf{X}_\ell = \mathbf{A}_{\mathsf{fhe}}\mathbf{R}_\ell - x_\ell \mathbf{G}$ for $\ell \in [L]$ with $\mathbf{x} \in \{0,1\}^L$, then

$$\mathbf{C} \;=\; \mathbf{A}_{\mathsf{fhe}}\mathbf{R}_C \;-\; \begin{pmatrix} \mathbf{0}_{n\times m'} \\ C(\mathbf{x}) \end{pmatrix} \;\in\; \mathbb{Z}_q^{(n+1)\times m'},$$

where $\big\|\mathbf{R}_C^T\big\|_\infty \le (m+2)^d \lceil \log_2 q \rceil \max_{\ell\in[L]} \big\|\mathbf{R}_\ell^T\big\|_\infty$. The new decryption equation is

$$\mathbf{s}^T\mathbf{C} \;=\; -\mathbf{e}_{\mathsf{fhe}}^T\mathbf{R}_C + C(\mathbf{x}) \;\in\; \mathbb{Z}_q^{1\times m'}.$$

3.5 Homomorphic Evaluation

We adopt the syntax of homomorphic evaluation algorithms used for the BGG+ encoding [BGG+14] from [AKY24a]. For parameters $q, n, L, \ell \in \mathbb{N}$ and $m = n\lceil \log_2 q \rceil$, a public matrix for the input encoding is denoted by $\mathbf{A}_{\mathsf{att}} \in \mathbb{Z}_q^{n\times(L+1)m}$.

The following two deterministic algorithms allows homomorphically evaluating a boolean circuit $C : \{0,1\}^L \to \{0,1\}^\ell$ with bits $\mathbf{x} \in \{0,1\}^L$ on the input encoding [BGG+14].

- $\mathsf{MEvalC}(\mathbf{A}_{\mathsf{att}}, C) \to \mathbf{H}_C$: it takes as input the public matrix $\mathbf{A}_{\mathsf{att}}$ and the circuit C and outputs a matrix $\mathbf{H}_C \in \mathbb{Z}_q^{(L+1)m\times \ell m}$.
- $\mathsf{MEvalCX}(\mathbf{A}_{\mathsf{att}}, C, \mathbf{x}) \to \mathbf{H}_{C,x}$: it takes as input the public matrix $\mathbf{A}_{\mathsf{att}}$, the circuit C, and the input $\mathbf{x}$, and outputs a matrix $\mathbf{H}_{C,x} \in \mathbb{Z}_q^{(L+1)m\times \ell m}$.

The outputs of these algorithms satisfies the following properties:

- For a circuit C of a depth d, $\big\|\mathbf{H}_C^T\big\|_\infty, \big\|\mathbf{H}_{C,x}^T\big\|_\infty \le (m+2)^d$ holds.
- For any $\mathbf{A}_{\mathsf{att}}$, C, and $\mathbf{x}$, $(\mathbf{A}_{\mathsf{att}} - (1, \mathbf{x}^T)\otimes \mathbf{G}_n)\mathbf{H}_{C,x} = \mathbf{A}_{\mathsf{att}}\mathbf{H}_C - C(\mathbf{x})\otimes \mathbf{G}_n$ holds, where $\mathbf{H}_C \leftarrow \mathsf{MEvalC}(\mathbf{A}_{\mathsf{att}}, C)$ and $\mathbf{H}_{C,x} \leftarrow \mathsf{MEvalCX}(\mathbf{A}_{\mathsf{att}}, C, \mathbf{x})$.

Additionally, we adopt evaluation algorithms for an inner product proposed in [GVW15]. The original algorithms are defined for an inner product bettween a vector $\mathbf{t} \in \mathbb{Z}_q^\ell$ and the output of the boolean circuit $C(\mathbf{x}) \in \{0,1\}^\ell$; ours in Lemma 5 extends the output of the circuit to a matrix $\mathbf{Y} \in \mathbb{Z}_q^{\ell\times\ell'}$ by combining a technique introduced in Section 4.1 of [BTVW17].

Lemma 5 (Evaluation of BGG+ encodings for a vector-matrix multiplication)**.** For every parameters $q, n, L, \ell, \ell' \in \mathbb{N}$, and $m = n\lceil \log_2 q \rceil$, the following two deterministic algorithms are defined:

- $\mathsf{MEvalC}^{\mathsf{VMM}}((\mathbf{A}_{\mathsf{att}}, \mathbf{A}_t), F) \to \mathbf{H}_F$: it takes as input the public matrixes $\mathbf{A}_{\mathsf{att}} \in \mathbb{Z}_q^{n\times(L+1)m}$ and $\mathbf{A}_t \in \mathbb{Z}_q^{n\times\ell m}$, and the matrix-valued circuit $F : \{0,1\}^L \to \mathbb{Z}_q^{\ell\times\ell'}$, and outputs a matrix $\mathbf{H}_F \in \mathbb{Z}_q^{(L+1+\ell)m\times\ell' m}$.

- $\mathsf{MEvalCX}^{\mathsf{VMM}}((\mathbf{A}_{\mathsf{att}}, \mathbf{A}_t), F, \mathbf{x}) \to \mathbf{H}_{F,x}$: it takes as input the public matrixes $\mathbf{A}_{\mathsf{att}} \in \mathbb{Z}_q^{n\times(L+1)m}$ and $\mathbf{A}_t \in \mathbb{Z}_q^{n\times\ell m}$, the matrix-valued circuit $F : \{0,1\}^L \to \mathbb{Z}_q^{\ell\times\ell'}$, and the input $\mathbf{x} \in \{0,1\}^L$, and outputs a matrix $\mathbf{H}_{F,x} \in \mathbb{Z}_q^{(L+1+\ell)m\times\ell' m}$.

The outputs of these algorithms satisfy the following properties:

- For a circuit F of a depth d, $\left\|\mathbf{H}_F^T\right\|_\infty, \left\|\mathbf{H}_{F,x}^T\right\|_\infty \le \ell\ell'\lceil\log_2 q\rceil^2(m+2)^{d+1}$ holds.
- For any $\mathbf{A}_{\mathsf{att}}$, $\mathbf{A}_t$, F, $\mathbf{x}$, and $\mathbf{t}$, $((\mathbf{A}_{\mathsf{att}}-(1,\mathbf{x}^T)\otimes\mathbf{G}_n), (\mathbf{A}_t-\mathbf{t}^T\otimes\mathbf{G}_n))\mathbf{H}_{F,x} = (\mathbf{A}_{\mathsf{att}}, \mathbf{A}_t)\mathbf{H}_F - \mathbf{t}^T\mathbf{Y}\otimes\mathbf{G}_n$ holds, where $\mathbf{H}_F \leftarrow \mathsf{MEvalC}^{\mathsf{VMM}}((\mathbf{A}_{\mathsf{att}}, \mathbf{A}_t), F)$, $\mathbf{H}_{F,x} \leftarrow \mathsf{MEvalCX}^{\mathsf{VMM}}((\mathbf{A}_{\mathsf{att}}, \mathbf{A}_t), F, \mathbf{x})$, and $\mathbf{Y} \leftarrow F(\mathbf{x})$.

Proof. Before describing the concrete consturctions of the algorithms, we present how to compute the encoding of $\mathbf{t}^T\mathbf{Y}$. Let F' be a boolean circuit such that the $((k-1)\ell\ell' + (j-1)\ell + i)$-th output bit is the k-th bit of the (i,j)-th element of $\mathbf{Y} = F(\mathbf{x})$, denoted by $y_{i,j,k} \in \{0,1\}$, for every $i \in [\ell]$, $j \in [\ell']$, and $k \in \lceil\log_2 q\rceil$. The matrixes $\mathbf{H}_{F'} \leftarrow \mathsf{MEvalC}(\mathbf{A}_{\mathsf{att}}, F')$ and $\mathbf{H}_{F',x} \leftarrow \mathsf{MEvalCX}(\mathbf{A}_{\mathsf{att}}, F', \mathbf{x})$ satisfies an equation $(\mathbf{A}_{\mathsf{att}} - (1,\mathbf{x}^T)\otimes\mathbf{G}_n)\mathbf{H}_{F',x} = \mathbf{A}_{\mathsf{att}}\mathbf{H}_{F'} - \mathsf{bits}(\mathbf{Y})\otimes\mathbf{G}_n$.

We observe that the multiplication $\mathbf{t}^T\mathbf{Y}$ is decomposed as follows:

$$\mathbf{t}^T\mathbf{Y} = \sum_{k\in\lceil\log_2 q\rceil} 2^{k-1} \sum_{j\in[\ell']} \Big(\sum_{i\in[\ell]} t_i y_{i,j,k}\Big)\mathbf{u}_{\ell',j}$$

, where $\mathbf{u}_{\ell',j} \in \mathbb{Z}_q^{\ell'}$ is a unit vector whose j-th element is 1 and the other elements are 0. For every $j \in [\ell']$ and $k \in \lceil\log_2 q\rceil$, the encoding of $\sum_{i\in[\ell]} t_i y_{i,j,k}$ is computed as below:

$$\begin{aligned}
&\Sigma_{i\in[\ell]} \left((\mathbf{A}_{\mathsf{att}} - (1,\mathbf{x}^T)\otimes\mathbf{G}_n), (\mathbf{A}_t - \mathbf{t}^T\otimes\mathbf{G}_n)\right) \begin{pmatrix} \mathbf{H}_{F',x,y_{i,j,k}} & \mathbf{0}_m \\ \mathbf{0}_m & \mathbf{H}_{t,x,i} \end{pmatrix} \begin{pmatrix} \mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i}) \\ y_{i,j,k}\mathbf{I}_m \end{pmatrix} \\
&= \Sigma_{i\in[\ell]} \left((\mathbf{A}_{\mathsf{att}}\mathbf{H}_{F',y_{i,j,k}} - y_{i,j,k}\mathbf{G}_n), (\mathbf{A}_t\mathbf{H}_{t,x,i} - t_i\mathbf{G}_n)\right) \begin{pmatrix} \mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i}) \\ y_{i,j,k}\mathbf{I}_m \end{pmatrix} \\
&= \mathbf{A}_{\mathsf{att}}(\Sigma_{i\in[\ell]}\mathbf{H}_{F',y_{i,j,k}}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})) - \Big(\sum_{i\in[\ell]} t_i y_{i,j,k}\Big)\mathbf{G}_n
\end{aligned}$$

, where $\mathbf{H}_{F',y_{i,j,k}}, \mathbf{H}_{F',x,y_{i,j,k}} \in \mathbb{Z}_q^{(L+1)m\times m}$ are the submatrix of $\mathbf{H}_{F'}$ and $\mathbf{H}_{F',x}$ from the $(((k-1)\ell\ell' + (j-1)\ell + i - 1)m + 1)$-th column to $(((k-1)\ell\ell' + (j-1)\ell + i)m)$-th column, respectively, and $\mathbf{H}_{t,x,i} \in \mathbb{Z}_q^{\ell m\times m}$ is a matrix such that a submatrix from the $((i-1)m+1)$-th row to (im)-th row is an identity matrix $\mathbf{I}_m$ and the other elements are 0.

We next compute a linear combination of the above encodings to obtain the encoding of $\mathbf{t}^T\mathbf{y}_j$, where $\mathbf{y}_j$ is the j-th row of $\mathbf{Y}$ for every $j \in [\ell']$, as follows:

$$\begin{aligned}
&\Sigma_{k\in\lceil\log_2 q\rceil} \left(\mathbf{A}_{\mathsf{att}}(\Sigma_{i\in[\ell]}\mathbf{H}_{F',y_{i,j,k}}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})) - \Big(\sum_{i\in[\ell]} t_i y_{i,j,k}\Big)\mathbf{G}_n\right)\mathbf{G}_n^{-1}(2^{k-1}\mathbf{G}_n) \\
&= \mathbf{A}_{\mathsf{att}}\left(\Sigma_{k\in\lceil\log_2 q\rceil}\Sigma_{i\in[\ell]}\mathbf{H}_{F',y_{i,j,k}}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})\mathbf{G}_n^{-1}(2^{k-1}\mathbf{G}_n)\right) - \mathbf{t}^T\mathbf{y}_j\mathbf{G}_n
\end{aligned}$$

Hence, the encoding of $\mathbf{t}^T\mathbf{Y}$ can be constructed by concatenating the above encodings in the row

direction as below:

$$
\begin{aligned}
&\Sigma_{j\in[\ell']}\mathbf{u}_{\ell',j}^T \otimes \left(\mathbf{A}_{\mathsf{att}}\left(\Sigma_{k\in\lceil\log_2 q\rceil}\Sigma_{i\in[\ell]}\mathbf{H}_{F',y_{i,j,k}}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})\mathbf{G}_n^{-1}(2^{k-1}\mathbf{G}_n)\right) - \mathbf{t}^T\mathbf{y}_j\mathbf{G}_n\right)\\
&= \mathbf{A}_{\mathsf{att}}\left(\Sigma_{j\in[\ell']}\mathbf{u}_{\ell',j}^T \otimes (\Sigma_{k\in\lceil\log_2 q\rceil}\Sigma_{i\in[\ell]}\mathbf{H}_{F',y_{i,j,k}}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})\mathbf{G}_n^{-1}(2^{k-1}\mathbf{G}_n))\right)\\
&- \Sigma_{j\in[\ell']}\mathbf{u}_{\ell',j}^T \otimes \mathbf{t}^T\mathbf{y}_j\mathbf{G}_n\\
&= \mathbf{A}_{\mathsf{att}}\left(\Sigma_{j\in[\ell']}(\Sigma_{k\in\lceil\log_2 q\rceil}\Sigma_{i\in[\ell]}\mathbf{H}_{F',y_{i,j,k}}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})\mathbf{G}_n^{-1}(2^{k-1}\mathbf{G}_n))(\mathbf{u}_{\ell',j}^T \otimes \mathbf{I}_m)\right)\\
&- \mathbf{t}^T\mathbf{Y} \otimes \mathbf{G}_n
\end{aligned}
$$

The above analysis indicates that the output of the $\mathsf{MEvalC}^{\mathsf{VMM}}$ and $\mathsf{MEvalCX}^{\mathsf{VMM}}$ algorithms are defined as follows, respectively:

$$
\begin{aligned}
\mathbf{H}_F &:= \Sigma_{j\in[\ell']}\left(\Sigma_{k\in\lceil\log_2 q\rceil}\Sigma_{i\in[\ell]}\begin{pmatrix}\mathbf{H}_{F',y_{i,j,k}}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})\\ \mathbf{0}_{\ell m\times m}\end{pmatrix}\mathbf{G}_n^{-1}(2^{k-1}\mathbf{G}_n)\right)(\mathbf{u}_{\ell',j}^T \otimes \mathbf{I}_m)\\
\mathbf{H}_{F,x} &:= \Sigma_{j\in[\ell']}\left(\Sigma_{k\in\lceil\log_2 q\rceil}\Sigma_{i\in[\ell]}\begin{pmatrix}\mathbf{H}_{F',x,y_{i,j,k}} & \mathbf{0}_m\\ \mathbf{0}_m & \mathbf{H}_{t,x,i}\end{pmatrix}\begin{pmatrix}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})\\ y_{i,j,k}\mathbf{I}_m\end{pmatrix}\mathbf{G}_n^{-1}(2^{k-1}\mathbf{G}_n)\right)(\mathbf{u}_{\ell',j}^T \otimes \mathbf{I}_m)
\end{aligned}
$$

They satisfy the claimed equation $((\mathbf{A}_{\mathsf{att}} - (1,\mathbf{x}^T)\otimes\mathbf{G}_n),(\mathbf{A}_t - \mathbf{t}^T\otimes\mathbf{G}_n))\mathbf{H}_{F,x} = (\mathbf{A}_{\mathsf{att}},\mathbf{A}_t)\mathbf{H}_F - \mathbf{t}^T\mathbf{Y}\otimes\mathbf{G}_n$.

We finally confirm the upper bounds of $\left\|\mathbf{H}_F^T\right\|_\infty$ and $\left\|\mathbf{H}_{F,x}^T\right\|_\infty$. Since $\left\|\mathbf{H}_{F'}^T\right\|_\infty, \left\|\mathbf{H}_{F',x}^T\right\|_\infty \leq (m+2)^d$, these norms are bounded as follows:

$$
\begin{aligned}
&\left\|\mathbf{H}_F^T\right\|_\infty\\
&\leq \left\|\left(\Sigma_{j\in[\ell']}\left(\Sigma_{k\in\lceil\log_2 q\rceil}\Sigma_{i\in[\ell]}\begin{pmatrix}\mathbf{H}_{F',y_{i,j,k}}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})\\ \mathbf{0}_{\ell m\times m}\end{pmatrix}\mathbf{G}_n^{-1}(2^{k-1}\mathbf{G}_n)\right)(\mathbf{u}_{\ell',j}^T \otimes \mathbf{I}_m)\right)^T\right\|_\infty\\
&\leq \ell'\left\|\left(\Sigma_{k\in\lceil\log_2 q\rceil}\Sigma_{i\in[\ell]}\begin{pmatrix}\mathbf{H}_{F',y_{i,j,k}}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})\\ \mathbf{0}_{\ell m\times m}\end{pmatrix}\mathbf{G}_n^{-1}(2^{k-1}\mathbf{G}_n)\right)^T\right\|_\infty\\
&\leq \ell'\lceil\log_2 q\rceil^2\left\|\Sigma_{i\in[\ell]}\begin{pmatrix}\mathbf{H}_{F',y_{i,j,k}}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})\\ \mathbf{0}_{\ell m\times m}\end{pmatrix}^T\right\|_\infty\\
&\leq \ell'\lceil\log_2 q\rceil^2\Sigma_{i\in[\ell]}\left\|\mathbf{H}_{F',y_{i,j,k}}^T\right\|_\infty\left\|\left(\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})\right)^T\right\|_\infty\\
&\leq \ell\ell'\lceil\log_2 q\rceil^2 m\left\|\mathbf{H}_{F',y_{i,j,k}}^T\right\|_\infty\\
&\leq \ell\ell'\lceil\log_2 q\rceil^2(m+2)^{d+1}\\
&\left\|\mathbf{H}_{F,x}^T\right\|_\infty\\
&\leq \left\|\left(\Sigma_{j\in[\ell']}\left(\Sigma_{k\in\lceil\log_2 q\rceil}\Sigma_{i\in[\ell]}\begin{pmatrix}\mathbf{H}_{F',x,y_{i,j,k}} & \mathbf{0}_m\\ \mathbf{0}_m & \mathbf{H}_{t,x,i}\end{pmatrix}\begin{pmatrix}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})\\ y_{i,j,k}\mathbf{I}_m\end{pmatrix}\mathbf{G}_n^{-1}(2^{k-1}\mathbf{G}_n)\right)(\mathbf{u}_{\ell',j}^T \otimes \mathbf{I}_m)\right)^T\right\|_\infty\\
&\leq \ell'\left\|\left(\Sigma_{k\in\lceil\log_2 q\rceil}\Sigma_{i\in[\ell]}\begin{pmatrix}\mathbf{H}_{F',x,y_{i,j,k}} & \mathbf{0}_m\\ \mathbf{0}_m & \mathbf{H}_{t,x,i}\end{pmatrix}\begin{pmatrix}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})\\ y_{i,j,k}\mathbf{I}_m\end{pmatrix}\mathbf{G}_n^{-1}(2^{k-1}\mathbf{G}_n)\right)^T\right\|_\infty\\
&\leq \ell'\lceil\log_2 q\rceil^2\Sigma_{i\in[\ell]}\left\|\begin{pmatrix}\mathbf{G}_n^{-1}(\mathbf{A}_t\mathbf{H}_{t,x,i})\\ y_{i,j,k}\mathbf{I}_m\end{pmatrix}^T\right\|_\infty\left\|\begin{pmatrix}\mathbf{H}_{F',x,y_{i,j,k}} & \mathbf{0}_m\\ \mathbf{0}_m & \mathbf{H}_{t,x,i}\end{pmatrix}^T\right\|_\infty\\
&\leq \ell\ell'\lceil\log_2 q\rceil^2(m+1)(m+2)^d\\
&\leq \ell\ell'\lceil\log_2 q\rceil^2(m+2)^{d+1}
\end{aligned}
$$

This completes the proof.

□

3.6 Pseudorandom Oracle Model

We recall the definition of pseudorandom oracle model (PROM) in [JLLW23].

Definition 1 (Pseudorandom Oracle Model [JLLW23])**.** Let PRF be a pseudorandom function with a key domain $\{0,1\}^\lambda$. The pseudorandom oracle model for PRF is the model in which all algorithms, including adversaries, can access to the oracle O. This oracle internally uses a random permutation $\mathsf{hMap} : \{0,1\}^\lambda \to \{0,1\}^\lambda$ and responds to the following two types of queries:

- $\mathsf{O}(\mathsf{hGen}, K) = \mathsf{hMap}(K)$,
- $\mathsf{O}(\mathsf{hEval}, h, \mathbf{x}) = \mathsf{PRF}(\mathsf{hMap}^{-1}(h), \mathbf{x})$.

4 Indistinguishability Obfuscation

4.1 Definition

We recall the definition of iO in [BGI+01].

Definition 2 (Indistinguishability Obfuscation [BGI+01])**.** For input and output sizes $L, \ell \in \mathbb{N}$, an indistinguishability obfuscator (iO) scheme is defined by two PPT algorithms $(\mathsf{Obf}, \mathsf{Eval})$ with the following syntax.

- $\mathsf{Obf}(1^\lambda, C) \to \widetilde{C}$: it takes as input a security parameter 1^λ and a circuit $C : \{0,1\}^L \to \{0,1\}^\ell$ and outputs an obfuscated circuit $\widetilde{C}$.
- $\mathsf{Eval}(\widetilde{C}, \mathbf{x}) \to \mathbf{y}$: it takes as input an obfuscated circuit $\widetilde{C}$ and an input $\mathbf{x} \in \{0,1\}^L$ and outputs the evaluation result $\mathbf{y} \in \{0,1\}^\ell$.

These algorithms must satisfy the following properties.

Definition 3 (Correctness of iO [BGI+01])**.** For any $\lambda \in \mathbb{N}$, any circuit $C : \{0,1\}^L \to \{0,1\}^\ell$, and any input $\mathbf{x} \in \{0,1\}^L$, it holds that

$$\mathsf{Eval}(\mathsf{Obf}(1^\lambda, C), \mathbf{x}) = C(\mathbf{x}).$$

Definition 4 (Polynomial Slowdown of iO [BGI+01])**.** There exists a polynomial poly such that for any $\lambda \in \mathbb{N}$ and any circuit $C : \{0,1\}^L \to \{0,1\}^\ell$, it holds that

$$|\mathsf{Obf}(1^\lambda, C)| \leq \mathsf{poly}(|C|).$$

Definition 5 (Indistinguishability of iO [BGI+01])**.** For any $\lambda \in \mathbb{N}$ and any two circuits C_0, C_1 that computes the same function and are of the same size, there exists a polynomial $\kappa = \kappa(\lambda)$ such that for every non-uniform adversary $\mathcal{A}$, it holds that

$$\begin{aligned} &\left|\Pr\left[\mathcal{A}(C_0, C_1, \widetilde{C_0}) = 1\right] \;-\; \Pr\left[\mathcal{A}(C_0, C_1, \widetilde{C_1}) = 1\right]\right| \leq \mathsf{negl}(\kappa), \\ &\mathrm{Size}(\mathcal{A}) \leq \mathsf{poly}(\kappa), \end{aligned}$$

where $\widetilde{C_0} := \mathsf{Obf}(1^\lambda, C_0)$ and $\widetilde{C_1} := \mathsf{Obf}(1^\lambda, C_1)$.

4.2 Construction

We straightforwardly construct an iO scheme for polynomial-sized circuits, denoted by $C : \{0,1\}^L \to \{0,1\}^\ell$, based on the ideas shown in Section 2.

Parameters: our construction depends on the following parameters:

- A security parameter $\lambda \in \mathbb{N}$.
- Input and output sizes of the circuit $L, \ell \in \mathbb{N}$.
- A depth of the circuit $\mathsf{dep}(\lambda) = \mathsf{poly}(\lambda)$.
- A modulus $q \in \mathbb{N}$.
- Dimentions of binary LWE $n \in \mathbb{N}$ and $m := (n+1)\lceil \log_2 q \rceil$.
- A Gaussian parameter σ_E.
- LWE dimentions $n_B := 2(n+1)$ and $m_B := \mathcal{O}(n\lceil \log_2 q \rceil)$.
- A Gaussian parameter $\gamma_B = \omega(\sqrt{n\lceil \log_2 q \rceil \log m})$.
- A bound $B \in \mathbb{N}$, which is exponentially smaller than $\frac{q}{4}$.

Ingredients: the following ingredients are employed:

- A pseudorandom function $\mathsf{PRF}_B : \{0,1\}^\lambda \times \{0,1\}^L \to [-\frac{q}{4} + B, \frac{q}{4} - B]^{1\times\ell}$.
- A pseudorandom oracle O. The output of O corresponds to another pseudorandom function $\mathsf{PRF}_H : \{0,1\}^\lambda \times \{0,1\}^L \to \{0,1\}^{1\times\ell}$.

Our construction consists of two algorithms: Obf and Eval defined as follows.

- $\mathsf{Obf}(1^\lambda, C) \to \widetilde{C}$:
  1. Sample $\bar{\mathbf{s}} \leftarrow\!\$\ \{0,1\}^n$ and set $\mathbf{s}^T := (\bar{\mathbf{s}}^T, -1)$.
  2. For every $b \in \{0,1\}$, sample $\bar{\mathbf{R}}_b \leftarrow\!\$\ \{0,1\}^{n\times n}$ and set $\mathbf{R}_b := \begin{pmatrix} \bar{\mathbf{R}}_b & \mathbf{0}_{n\times 1} \\ \mathbf{0}_{1\times n} & 1 \end{pmatrix}$.
  3. For every $i \in [L]$, define $\hat{\mathbf{s}}_{\mathbf{x}_i}^T := \mathbf{s}^T \prod_{j\in[i]} \mathbf{R}_{x_j} = (\bar{\mathbf{s}}^T \prod_{j\in[i]} \bar{\mathbf{R}}_{x_j}, -1)$. For $i = 0$, set $\hat{\mathbf{s}}_\epsilon^T := \mathbf{s}^T$.
  4. Sample $\bar{\mathbf{t}} \leftarrow\!\$\ \{0,1\}^n$, $\bar{\mathbf{A}}_{\mathsf{fhe}} \leftarrow\!\$\ \mathbb{Z}_q^{n\times m}$, and $\mathbf{e}_{\mathsf{fhe}} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\sigma_E}^m$, and compute $\mathbf{t}^T := (\bar{\mathbf{t}}^T, -1)$ and $\mathbf{A}_{\mathsf{fhe}} := \begin{pmatrix} \bar{\mathbf{A}}_{\mathsf{fhe}} \\ \bar{\mathbf{t}}^T \bar{\mathbf{A}}_{\mathsf{fhe}} + \mathbf{e}_{\mathsf{fhe}}^T \end{pmatrix}$.
  5. Sample the seeds of PRF_B and PRF_H denoted by $K_B, K_H \leftarrow\!\$\ \{0,1\}^\lambda$, respectively.
  6. Call PRO to obtain a handle $h := \mathsf{O}(\mathsf{hGen}, K_H)$.
  7. Sample $\mathbf{R}_{\mathsf{fhe}} \leftarrow\!\$\ \{0,1\}^{m\times m(|C|+2\lambda)}$ and compute $\mathbf{X} := \mathbf{A}_{\mathsf{fhe}}\mathbf{R}_{\mathsf{fhe}} - (C, K_B, K_H) \otimes \mathbf{G}_{n+1}$, where the bit size of $\mathbf{X}$ is $L_{\mathsf{fhe}} := (n+1)m(|C|+2\lambda)\lceil \log_2 q \rceil$. In the following, we define $L' := 1 + L_{\mathsf{fhe}} + L$.
  8. For every $i \in [0, L]$, sample $\mathbf{A}_{\mathsf{att},i} \leftarrow\!\$\ \mathbb{Z}_q^{(n+1)\times L'm}$ and $\mathbf{A}_{t,i} \leftarrow\!\$\ \mathbb{Z}_q^{(n+1)\times(n+1)m}$, respectively.
  9. Compute $\mathbf{c}_{\mathsf{att},\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T(\mathbf{A}_{\mathsf{att},0} - (1, \mathsf{bits}(\mathbf{X}), \mathbf{0}_L) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{\mathsf{att},\epsilon}}^T$, where $\mathbf{e}_{c_{\mathsf{att},\epsilon}} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\sigma_E}^{L'm}$.
  10. Compute $\mathbf{c}_{t,\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T(\mathbf{A}_{t,0} - \mathbf{t}^T \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{t,\epsilon}}^T$, where $\mathbf{e}_{t_\epsilon} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\sigma_E}^{(n+1)m}$.
  11. Execute $(\mathbf{B}_{0,\star}, \mathbf{B}_{0,\star,\gamma_B}^{-1}) \leftarrow \mathsf{TrapGen}(1^{n_B}, 1^{m_B}, q)$.
  12. For every $i \in [L]$ and $b \in \{0,1,\star\}$, execute $(\mathbf{B}_{i,b}, \mathbf{B}_{i,b,\gamma_B}^{-1}) \leftarrow \mathsf{TrapGen}(1^{n_B}, 1^{m_B}, q)$.

13. Compute $\mathbf{p}_\epsilon^T := (\hat{\mathbf{s}}_\epsilon^T, \hat{\mathbf{s}}_\epsilon^T)\mathbf{B}_{0,\star} + \mathbf{e}_{p_\epsilon}^T$, where $\mathbf{e}_{p_\epsilon} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\sigma_E}^{m_B}$.
14. Let $\mathbf{U}_0, \mathbf{U}_1, \mathbf{U}_\star$ be defined as follows, respectively.

$$\mathbf{U}_0 := \begin{pmatrix} \mathbf{I}_{n+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_0 \end{pmatrix}, \mathbf{U}_1 := \begin{pmatrix} \mathbf{I}_{n+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_1 \end{pmatrix}, \mathbf{U}_\star := \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{n+1} & \mathbf{I}_{n+1} \end{pmatrix}$$

15. For every $i \in [L]$ and $b \in \{0,1\}$, sample $\mathbf{M}_{i,b} \leftarrow\!\$\ \mathbf{B}_{i-1,\star,\gamma_B}^{-1}(\mathbf{U}_b\mathbf{B}_{i,b})$ and $\mathbf{N}_{i,b} \leftarrow\!\$\ \mathbf{B}_{i,b,\gamma_B}^{-1}(\mathbf{U}_\star\mathbf{B}_{i,\star})$.
16. For $i \in [L]$ and $b \in \{0,1\}$, compute $\mathbf{T}_{\mathsf{att},i,b} := \mathbf{I}_{L'} \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{R}_b\mathbf{G}_{n+1})$ and $\mathbf{T}_{t,i,b} := \mathbf{I}_{n+1} \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{R}_b\mathbf{G}_{n+1})$.
17. For every $i \in [L]$ and $b \in \{0,1\}$, sample $\mathbf{K}_{i,b}$ as follows.

$$\mathbf{K}_{i,b} \leftarrow\!\$\ \mathbf{B}_{i,b,\gamma_B}^{-1}\begin{pmatrix} -(\mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,b}, \mathbf{A}_{t,i-1}\mathbf{T}_{t,i,b}) \\ (\mathbf{A}_{\mathsf{att},i} - (\mathbf{0}_{L_{\mathsf{fhe}}+i}, b, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1}, \mathbf{A}_{t,i}) \end{pmatrix}$$

18. Let $f[\mathbf{x}_L]$ be a circuit that on input (C, K_B, K_H) outputs $\lceil \frac{q}{2} \rceil C(\mathbf{x}_L) + \mathsf{PRF}_B(K_B, \mathbf{x}_L) + \lceil \frac{q}{2} \rceil \mathsf{PRF}_H(K_H, \mathbf{x}_L)$.
19. Let F be a circuit that on input $(1, \mathsf{bits}(\mathbf{X}), \mathbf{x}_L)$ outputs $\mathsf{VEval}_{f[\mathbf{x}_L]}(\mathbf{X})$, where $\mathsf{VEval}_{f[\mathbf{x}_L]} := \mathsf{MakeVEvalCkt}(n, m, q, f[\mathbf{x}_L])$. This output denoted by $\mathbf{Y}$ satisfies

$$\mathbf{Y} = \mathbf{A}_{\mathsf{fhe}}\mathbf{R}_F - \begin{pmatrix} \mathbf{0}_{n\times\ell} \\ f[\mathbf{x}_L](C, K_B, K_H) \end{pmatrix}.$$

20. Compute $\mathbf{H}_F := \mathsf{MEvalC}^{\mathsf{VMM}}((\mathbf{A}_{\mathsf{att},L}, \mathbf{A}_{t,L}), F)$.
21. Compute $\mathbf{A}_F := (\mathbf{A}_{\mathsf{att},L}, \mathbf{A}_{t,L})\mathbf{H}_F(\mathbf{I}_\ell \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{u}_{n+1,n+1}))$, and sample $\mathbf{K}_F \leftarrow\!\$\ \mathbf{B}_{L,\star,\gamma_B}^{-1}\begin{pmatrix} \mathbf{A}_F \\ \mathbf{0} \end{pmatrix}$.
22. Output

$$\widetilde{C} := \begin{pmatrix} \{\mathbf{R}_b\}_{b\in\{0,1\}}, \mathbf{A}_{\mathsf{fhe}}, h, \mathbf{X}, \{\mathbf{A}_{\mathsf{att},i}\}_{i\in[0,L]}, \{\mathbf{A}_{t,t}\}_{i\in[0,L]}, \mathbf{c}_{\mathsf{att},\epsilon}, \mathbf{c}_{t,\epsilon}, \\ \mathbf{p}_\epsilon, \{\mathbf{M}_{i,b}\}_{i\in[L],b\in\{0,1\}}, \{\mathbf{N}_{i,b}\}_{i\in[L],b\in\{0,1\}}, \{\mathbf{K}_{i,b}\}_{i\in[L],b\in\{0,1\}}, \mathbf{K}_F \end{pmatrix}$$

- $\mathsf{Eval}(\widetilde{C}, \mathbf{x}) \to y$:
  1. Parse $\widetilde{C}$ as

$$\begin{pmatrix} \{\mathbf{R}_b\}_{b\in\{0,1\}}, \mathbf{A}_{\mathsf{fhe}}, h, \mathbf{X}, \{\mathbf{A}_{\mathsf{att},i}\}_{i\in[0,L]}, \{\mathbf{A}_{t,i}\}_{i\in[0,L]}, \mathbf{c}_{\mathsf{att},\epsilon}, \mathbf{c}_{t,\epsilon}, \\ \mathbf{p}_\epsilon, \{\mathbf{M}_{i,b}\}_{i\in[L],b\in\{0,1\}}, \{\mathbf{N}_{i,b}\}_{i\in[L],b\in\{0,1\}}, \{\mathbf{K}_{i,b}\}_{i\in[L],b\in\{0,1\}}, \mathbf{K}_F \end{pmatrix}$$

  2. For every $i \in [L]$, repeat the following process:
     (a) Compute $\mathbf{T}_{\mathsf{att},i,x_i} = \mathbf{I}_{L'} \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{R}_{x_i}\mathbf{G}_{n+1})$ and $\mathbf{T}_{t,i,x_i} = \mathbf{I}_{n+1} \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{R}_{x_i}\mathbf{G}_{n+1})$.
     (b) Compute $\mathbf{q}_{\mathbf{x}_{i-1}x_i}^T := \mathbf{p}_{\mathbf{x}_{i-1}}^T\mathbf{M}_{i,x_i}$.
     (c) Compute $\mathbf{p}_{\mathbf{x}_i}^T := \mathbf{q}_{\mathbf{x}_{i-1}x_i}^T\mathbf{N}_{i,x_i}$.
     (d) Compute $(\mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T, \mathbf{v}_{t,\mathbf{x}_i}^T) := \mathbf{q}_{\mathbf{x}_{i-1}x_i}^T\mathbf{K}_{i,x_i}$.
     (e) Compute $\mathbf{c}_{\mathsf{att},\mathbf{x}_i}^T := \mathbf{c}_{\mathsf{att},\mathbf{x}_{i-1}}^T\mathbf{T}_{\mathsf{att},i,x_i} + \mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T$.
     (f) Compute $\mathbf{c}_{t,\mathbf{x}_i}^T := \mathbf{c}_{t,\mathbf{x}_{i-1}}^T\mathbf{T}_{t,i,x_i} + \mathbf{v}_{t,\mathbf{x}_i}^T$.
  3. Compute $F = \mathsf{MakeVEvalCkt}(n, m, q, f[\mathbf{x}_L])$, where $f[\mathbf{x}_L]$ is defined in the same manner as one in the Obf algorithm.

4. Compute $\mathbf{H}_{F,\mathbf{x}_L} := \mathsf{MEvalCX}^{\mathsf{VMM}}((\mathbf{A}_{\mathsf{att},L}, \mathbf{A}_{t,L}), F, (1, \mathbf{X}, \mathbf{x}_L))$.
5. Compute $\mathbf{c}^T_{F,\mathbf{x}_L} := (\mathbf{c}^T_{\mathsf{att},\mathbf{x}_L}, \mathbf{c}^T_{t,\mathbf{x}_L})\mathbf{H}_{F,\mathbf{x}_L}(\mathbf{I}_\ell \otimes \mathbf{G}^{-1}_{n+1}(\mathbf{u}_{n+1,n+1}))$.
6. Compute $\mathbf{v}^T_{F,\mathbf{x}_L} := \mathbf{p}^T_{\mathbf{x}_L}\mathbf{K}_F$.
7. Call PRO to obtain the randomness $\mathbf{r}_{H,\mathbf{x}_L} := \mathsf{O}(\mathsf{hEval}, h, \mathbf{x}_L)$.
8. Compute $\mathbf{z}^T_{\mathbf{x}_L} := \mathbf{c}^T_{F,\mathbf{x}_L} - \mathbf{v}^T_{F,\mathbf{x}_L} - \lceil \frac{q}{2} \rceil \mathbf{r}_{H,\mathbf{x}_L}$.
9. For every $i \in [\ell]$, set $y_i = 0$ if $z_i \in [-\frac{q}{4}, \frac{q}{4})$ and $y_i = 1$ otherwise.
10. Output $\mathbf{y} = (y_1, \ldots, y_\ell)$.

We can easily confirm that the size of the obfuscated circuit $\widetilde{C}$ is polynomially bounded by the size of the original circuit C; thus, our construction satisfies the polynomial slowdown property (Definition 4).

4.3 Correctness Proof

Theorem 1. The construction in Subsection 4.2 satisfies correctness of iO (Definition 3).

Proof. We recall that for a vector $\mathbf{e} \in \mathbb{Z}$ sampled from $\mathcal{D}^m_{\mathbb{Z},\sigma_E}$, it holds that $\|\mathbf{e}\|_\infty \leq \sigma_E\sqrt{\lambda}$ except with negligible probability as shown in Lemma 1. In similar to [CDCG+18], for a multiplication between a vector $\mathbf{e} \in \mathbb{Z}^m_q$ and a matrix $\mathbf{A} \in \mathbb{Z}^{n\times m}_q$ that are sampled from the discrete Gaussian distribution with Gaussian parameters σ_E, σ_A, respectively, we apply Central Limit Theorem to claim that the following holds.

$$\|\mathbf{A}\mathbf{e}\|_\infty \leq \sqrt{m}\sigma_A\sqrt{\lambda}\,\|\mathbf{e}\|_\infty \leq \sqrt{m}\sigma_A\sigma_E\lambda$$

For every $i \in [L]$ and any $\mathbf{x}_i \in \{0,1\}^i$, the following relations hold.

1.
$$\begin{aligned}\mathbf{q}^T_{\mathbf{x}_i} &= \mathbf{p}^T_{\mathbf{x}_{i-1}}\mathbf{M}_{i,x_i}\\ &= (\hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}, \hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}\mathbf{R}_{x_i})\mathbf{B}_{i,x_i} + \mathbf{e}^T_{p_{\mathbf{x}_{i-1}}}\mathbf{M}_{i,x_i}\\ &= (\hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}, \hat{\mathbf{s}}^T_{\mathbf{x}_i})\mathbf{B}_{i,x_i} + \mathbf{e}^T_{q_{\mathbf{x}_i}},\end{aligned}$$

where $\left\|\mathbf{e}_{q_{\mathbf{x}_i}}\right\|_\infty \leq \left\|\mathbf{M}^T_{i,x_i}\right\|_\infty \left\|\mathbf{e}_{p_{\mathbf{x}_{i-1}}}\right\|_\infty \leq \gamma_B\sqrt{\lambda m_B}\left\|\mathbf{e}_{p_{\mathbf{x}_{i-1}}}\right\|_\infty$ holds.

2.
$$\begin{aligned}\mathbf{p}^T_{\mathbf{x}_i} &= \mathbf{q}^T_{\mathbf{x}_i}\mathbf{N}_{i,x_i}\\ &= (\hat{\mathbf{s}}^T_{\mathbf{x}_i}, \hat{\mathbf{s}}^T_{\mathbf{x}_i})\mathbf{B}_{i,\star} + \mathbf{e}^T_{q_{\mathbf{x}_i}}\mathbf{N}_{i,x_i}\\ &= (\hat{\mathbf{s}}^T_{\mathbf{x}_i}, \hat{\mathbf{s}}^T_{\mathbf{x}_i})\mathbf{B}_{i,\star} + \mathbf{e}^T_{p_{\mathbf{x}_i}},\end{aligned}$$

where $\left\|\mathbf{e}_{p_{\mathbf{x}_i}}\right\|_\infty \leq \left\|\mathbf{N}^T_{i,x_i}\right\|_\infty \left\|\mathbf{e}_{q_{\mathbf{x}_i}}\right\|_\infty \leq \gamma_B\sqrt{\lambda m_B}\left\|\mathbf{e}_{q_{\mathbf{x}_i}}\right\|_\infty$ holds.

3.
$$\begin{aligned}(\mathbf{v}^T_{\mathsf{att},\mathbf{x}_i}, \mathbf{v}^T_{t,\mathbf{x}_i}) &= \mathbf{q}^T_{\mathbf{x}_{i-1}x_i}\mathbf{K}_{i,x_i}\\ &= -\hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}(\mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,x_i}, \mathbf{A}_{t,i-1}\mathbf{T}_{t,i,x_i})\\ &\quad + \hat{\mathbf{s}}^T_{\mathbf{x}_i}(\mathbf{A}_{\mathsf{att},i} - (\mathbf{0}_{L_{\mathsf{fhe}}+i}, b, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1}, \mathbf{A}_{t,i}) + \mathbf{e}^T_{q_{\mathbf{x}_i}}\mathbf{K}_{i,x_i}\\ &= -\hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}(\mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,x_i}, \mathbf{A}_{t,i-1}\mathbf{T}_{t,i,x_i})\\ &\quad + \hat{\mathbf{s}}^T_{\mathbf{x}_i}(\mathbf{A}_{\mathsf{att},i} - (\mathbf{0}_{L_{\mathsf{fhe}}+i}, b, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1}, \mathbf{A}_{t,i}) + (\mathbf{e}^T_{v_{\mathsf{att},\mathbf{x}_i}}, \mathbf{e}^T_{v_{t,\mathbf{x}_i}}),\end{aligned}$$

where $\left\|\mathbf{e}_{v_{\mathsf{att},\mathbf{x}_i}}\right\|_\infty, \left\|\mathbf{e}_{v_{t,\mathbf{x}_i}}\right\|_\infty \leq \left\|\mathbf{K}_{i,x_i}\right\|_\infty \left\|\mathbf{e}_{q_{\mathbf{x}_i}}\right\|_\infty \leq \gamma_B\sqrt{\lambda m_B}\left\|\mathbf{e}_{q_{\mathbf{x}_i}}\right\|_\infty$ holds.

4.

$$\begin{aligned}\mathbf{c}^T_{\mathsf{att},\mathbf{x}_{i-1}}\mathbf{T}_{\mathsf{att},i,x_i} &= \hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}(\mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,x_i} - \mathbf{R}_{x_i}((1,\mathsf{bits}(\mathbf{X}),\mathbf{x}^T_{i-1},\mathbf{0}_{L-i+1})\otimes\mathbf{G}_{n+1}))\\ &\quad+\mathbf{e}^T_{c_{\mathsf{att},\mathbf{x}_{i-1}}}\mathbf{T}_{\mathsf{att},i,x_i}\\ &= \hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}\mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,x_i} - \hat{\mathbf{s}}^T_{\mathbf{x}_i}((1,\mathsf{bits}(\mathbf{X}),\mathbf{x}^T_{i-1},\mathbf{0}_{L-i+1})\otimes\mathbf{G}_{n+1})\\ &\quad+\mathbf{e}^T_{c_{\mathsf{att},\mathbf{x}_{i-1}}}\mathbf{T}_{\mathsf{att},i,x_i},\end{aligned}$$

where $\left\|\mathbf{T}^T_{\mathsf{att},i,x_i}\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_{i-1}}}\right\|_\infty \le \left\|\mathbf{T}^T_{\mathsf{att},i,x_i}\right\|_\infty \left\|\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_{i-1}}}\right\|_\infty \le m\left\|\mathbf{e}^T_{c_{\mathsf{att},\mathbf{x}_{i-1}}}\right\|_\infty$ holds because each column of $\mathbf{T}_{\mathsf{att},i,x_i} = \mathbf{I}_{L'}\otimes\mathbf{G}^{-1}_{n+1}(\mathbf{R}_{x_i}\mathbf{G}_{n+1})$ contains 1 at most m times.

5.

$$\begin{aligned}\mathbf{c}^T_{t,\mathbf{x}_{i-1}}\mathbf{T}_{t,i,x_i} &= \hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}(\mathbf{A}_{t,i-1}\mathbf{T}_{t,i,x_i} - \mathbf{R}_{x_i}(\mathbf{t}^T\otimes\mathbf{G}_{n+1}))\\ &\quad+\mathbf{e}^T_{c_{t,\mathbf{x}_{i-1}}}\mathbf{T}_{t,i,x_i}\\ &= \hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}\mathbf{A}_{t,i-1}\mathbf{T}_{t,i,x_i} - \hat{\mathbf{s}}^T_{\mathbf{x}_i}(\mathbf{t}^T\otimes\mathbf{G}_{n+1})\\ &\quad+\mathbf{e}^T_{c_{t,\mathbf{x}_{i-1}}}\mathbf{T}_{t,i,x_i},\end{aligned}$$

where $\left\|\mathbf{T}^T_{t,i,x_i}\mathbf{e}_{c_{t,\mathbf{x}_{i-1}}}\right\|_\infty \le \left\|\mathbf{T}^T_{t,i,x_i}\right\|_\infty \left\|\mathbf{e}_{c_{t,\mathbf{x}_{i-1}}}\right\|_\infty \le m\left\|\mathbf{e}^T_{c_{t,\mathbf{x}_{i-1}}}\right\|_\infty$ holds for the same reason.

Based on the above analysis of the error norm, we can see that each error is bounded as follows for every $i\in[L]$:

$$\begin{aligned}&\left\|\mathbf{e}_{p_{\mathbf{x}_i}}\right\|_\infty \le \sigma_E\gamma_B^{2i}m_B^i\lambda^{i+\frac{1}{2}}, \left\|\mathbf{e}_{q_{\mathbf{x}_i}}\right\|_\infty \le \sigma_E\gamma_B^{2i-1}m_B^{i-\frac{1}{2}}\lambda^i,\\ &\left\|\mathbf{e}_{v_{\mathsf{att},\mathbf{x}_i}}\right\|_\infty, \left\|\mathbf{e}_{v_{t,\mathbf{x}_i}}\right\|_\infty \le \sigma_E\gamma_B^{2i}m_B^i\lambda^{i+\frac{1}{2}}\end{aligned}$$

Besides, the following equations hold for every $\mathbf{x}_i\in\{0,1\}^i$:

$$\begin{aligned}\mathbf{c}^T_{\mathsf{att},\mathbf{x}_i} &= \mathbf{c}^T_{\mathsf{att},\mathbf{x}_{i-1}}\mathbf{T}_{\mathsf{att},i,x_i} + \mathbf{v}^T_{\mathsf{att},\mathbf{x}_i}\\ &= \hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}\mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,x_i} - \hat{\mathbf{s}}^T_{\mathbf{x}_i}((1,\mathsf{bits}(\mathbf{X}),\mathbf{x}^T_{i-1},\mathbf{0}_{L-i+1})\otimes\mathbf{G}_{n+1}) + \mathbf{e}^T_{c_{\mathsf{att},\mathbf{x}_{i-1}}}\mathbf{T}_{\mathsf{att},i,x_i}\\ &\quad-\hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}\mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,x_i} + \hat{\mathbf{s}}^T_{\mathbf{x}_i}(\mathbf{A}_{\mathsf{att},i} - (\mathbf{0}_{L_{\mathsf{fhe}}+i},b,\mathbf{0}_{L-i})\otimes\mathbf{G}_{n+1}) + \mathbf{e}^T_{v_{\mathsf{att},\mathbf{x}_i}}\\ &= \hat{\mathbf{s}}^T_{\mathbf{x}_i}(\mathbf{A}_{\mathsf{att},i} - ((1,\mathsf{bits}(\mathbf{X}),\mathbf{x}^T_i,\mathbf{0}_{L-i})\otimes\mathbf{G}_{n+1})) + \mathbf{e}^T_{c_{\mathsf{att},\mathbf{x}_i}}\\ \mathbf{c}^T_{t,\mathbf{x}_i} &= \mathbf{c}^T_{t,\mathbf{x}_{i-1}}\mathbf{T}_{t,i,x_i} + \mathbf{v}^T_{t,\mathbf{x}_i}\\ &= \hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}\mathbf{A}_{t,i-1}\mathbf{T}_{t,i,x_i} - \hat{\mathbf{s}}^T_{\mathbf{x}_i}(\mathbf{t}^T\otimes\mathbf{G}_{n+1}) + \mathbf{e}^T_{c_{t,\mathbf{x}_{i-1}}}\mathbf{T}_{t,i,x_i}\\ &\quad-\hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}\mathbf{A}_{t,i-1}\mathbf{T}_{t,i,x_i} + \hat{\mathbf{s}}^T_{\mathbf{x}_i}\mathbf{A}_{t,i} + \mathbf{e}^T_{t,v_{\mathbf{x}_i}}\\ &= \hat{\mathbf{s}}^T_{\mathbf{x}_i}(\mathbf{A}_{t,i} - \mathbf{t}^T\otimes\mathbf{G}_{n+1}) + \mathbf{e}^T_{c_{t,\mathbf{x}_i}}\end{aligned}$$

, where $\left\|\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_i}}\right\|_\infty \le m\left\|\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_{i-1}}}\right\|_\infty + \sigma_E\gamma_B^{2i}m_B^i\lambda^{i+\frac{1}{2}}$ and $\left\|\mathbf{e}_{c_{t,\mathbf{x}_i}}\right\|_\infty \le m\left\|\mathbf{e}_{c_{t,\mathbf{x}_{i-1}}}\right\|_\infty + \sigma_E\gamma_B^{2i}m_B^i\lambda^{i+\frac{1}{2}}$.

By treating the above relationships between $\left\|\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_{i-1}}}\right\|_\infty$ and $\left\|\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_i}}\right\|_\infty$ as recurrence relations, we can see that $\left\|\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_L}}\right\|_\infty, \left\|\mathbf{e}_{c_{t,\mathbf{x}_L}}\right\|_\infty \le B_c$ holds, where B_c is defined as follows.

$$B_c := \sigma_E\sqrt{\lambda}m^L + \frac{\sigma_E\gamma_B^2 m_B\lambda^{\frac{3}{2}}}{m-\gamma_B^2 m_B\lambda}\left(m^L - (\gamma_B^2 m_B\lambda)^L\right)$$

We next confirm that $\mathbf{c}_{F,\mathbf{x}_L}$ encodes the decryption result of $\mathbf{Y}$.

$$\begin{aligned}
\mathbf{c}_{F,\mathbf{x}_L}^T &= (\mathbf{c}_{\mathsf{att},\mathbf{x}_L}^T, \mathbf{c}_{t,\mathbf{x}_L}^T)\mathbf{H}_{F,\mathbf{x}_L}(\mathbf{I}_\ell \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{u}_{n+1,n+1})) \\
&= \hat{\mathbf{s}}_{\mathbf{x}_L}^T \left((\mathbf{A}_{\mathsf{att},L}, \mathbf{A}_{t,L})\mathbf{H}_F - \mathbf{t}^T\mathbf{Y} \otimes \mathbf{G}_{n+1}\right)(\mathbf{I}_\ell \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{u}_{n+1,n+1})) \\
&+ (\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_L}}^T, \mathbf{e}_{c_{t,\mathbf{x}_L}}^T)\mathbf{H}_{F,\mathbf{X}}(\mathbf{I}_\ell \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{u}_{n+1,n+1})) \\
&= \hat{\mathbf{s}}_{\mathbf{x}_L}^T (\mathbf{A}_{\mathsf{att},L}, \mathbf{A}_{t,L})\mathbf{H}_F(\mathbf{I}_\ell \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{u}_{n+1,n+1})) - \hat{\mathbf{s}}_{\mathbf{x}_L}^T \left(\mathbf{t}^T\mathbf{Y} \otimes \mathbf{u}_{n+1,n+1}\right) \\
&+ (\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_L}}^T, \mathbf{e}_{c_{t,\mathbf{x}_L}}^T)\mathbf{H}_{F,\mathbf{X}}(\mathbf{I}_\ell \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{u}_{n+1,n+1})) \\
&= \hat{\mathbf{s}}_{\mathbf{x}_L}^T \mathbf{A}_F - \left(\mathbf{t}^T\mathbf{Y} \otimes \hat{\mathbf{s}}_{\mathbf{x}_L}^T \mathbf{u}_{n+1,n+1}\right) \\
&+ (\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_L}}^T, \mathbf{e}_{c_{t,\mathbf{x}_L}}^T)\mathbf{H}_{F,\mathbf{X}}(\mathbf{I}_\ell \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{u}_{n+1,n+1})) \\
&= \hat{\mathbf{s}}_{\mathbf{x}_L}^T \mathbf{A}_F + \mathbf{t}^T\mathbf{Y} + (\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_L}}^T, \mathbf{e}_{c_{t,\mathbf{x}_L}}^T)\mathbf{H}_{F,\mathbf{X}}(\mathbf{I}_\ell \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{u}_{n+1,n+1})) \\
&= \hat{\mathbf{s}}_{\mathbf{x}_L}^T \mathbf{A}_F + (\left\lceil \frac{q}{2} \right\rfloor C(\mathbf{x}_L) + \mathsf{PRF}_B(K_B, \mathbf{x}_L) + \left\lceil \frac{q}{2} \right\rfloor \mathsf{PRF}_H(K_H, \mathbf{x}_L) - \mathbf{e}_{\mathsf{fhe}}^T \mathbf{R}_f) \\
&+ (\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_L}}^T, \mathbf{e}_{c_{t,\mathbf{x}_L}}^T)\mathbf{H}_{F,\mathbf{X}}(\mathbf{I}_\ell \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{u}_{n+1,n+1})) \\
&= \hat{\mathbf{s}}_{\mathbf{x}_L}^T \mathbf{A}_F + (\left\lceil \frac{q}{2} \right\rfloor C(\mathbf{x}_L) + \mathsf{PRF}_B(K_B, \mathbf{x}_L) + \left\lceil \frac{q}{2} \right\rfloor \mathsf{PRF}_H(K_H, \mathbf{x}_L)) + \mathbf{e}_{c_{F,\mathbf{x}_L}}^T \qquad (1)
\end{aligned}$$

, where $\mathbf{e}_{c_{F,\mathbf{x}_L}}^T := -\mathbf{e}_{\mathsf{fhe}}^T \mathbf{R}_f + (\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_L}}^T, \mathbf{e}_{c_{t,\mathbf{x}_L}}^T)\mathbf{H}_{F,\mathbf{X}}(\mathbf{I}_\ell \otimes \mathbf{G}_{n+1}^{-1}(\mathbf{u}_{n+1,n+1}))$. Here, $\hat{\mathbf{s}}_{\mathbf{x}_L}^T \mathbf{u}_{n+1,n+1} = -1$ holds since the last element of $\hat{\mathbf{s}}_{\mathbf{x}_L}$ is -1.

By Lemma 4, it holds that

$$\begin{aligned}
\left\|\mathbf{R}_f^T\right\|_\infty &\leq (m+2)^{\mathsf{dep}} \lceil \log_2 q \rceil \max_{\ell \in [L]} \left\|\mathbf{R}_\ell^T\right\|_\infty \\
&= m(m+2)^{\mathsf{dep}} \lceil \log_2 q \rceil \\
&\leq (m+2)^{\mathsf{dep}+1} \lceil \log_2 q \rceil.
\end{aligned}$$

Therefore, Lemma 5 follows that the following upper bound of $\left\|\mathbf{e}_{c_{F,\mathbf{x}_L}}\right\|_\infty$:

$$\begin{aligned}
\left\|\mathbf{e}_{c_{F,\mathbf{x}_L}}\right\|_\infty &\leq m \left\|\left((\mathbf{e}_{c_{\mathsf{att},\mathbf{x}_L}}^T, \mathbf{e}_{c_{t,\mathbf{x}_L}}^T)\mathbf{H}_{F,\mathbf{X}}\right)^T\right\|_\infty + \left\|\mathbf{R}_f^T \mathbf{e}_{\mathsf{fhe}}\right\|_\infty \\
&\leq mB_c \left\|\mathbf{H}_{F,\mathbf{X}}^T\right\|_\infty + \sigma_E \sqrt{\lambda} \left\|\mathbf{R}_f\right\|_\infty \\
&\leq mB_c \left((n+1)\ell \lceil \log_2 q \rceil^2 (m+2)^{\mathsf{dep}'+1}\right) \\
&+ \sigma_E \sqrt{\lambda} (m+2)^{\mathsf{dep}+1} \lceil \log_2 q \rceil \\
&\leq \left((n+1)\ell B_c + \sigma_E \sqrt{\lambda}\right) \lceil \log_2 q \rceil^2 (m+2)^{\mathsf{dep}'+2}, \qquad (2)
\end{aligned}$$

where $\mathsf{dep}' := \mathsf{dep} \cdot \mathcal{O}(\log_2 m \log_2 \log_2 q) + \mathcal{O}(\log^2 \lceil \log_2 q \rceil)$. In the following, let the last R.H.S. 2 be defined as B_F.

A preimage of $\mathbf{A}_F$ is produced by using $\mathbf{v}_{F,\mathbf{x}_L}^T$ as follows:

$$\begin{aligned}
\mathbf{v}_{F,\mathbf{x}_L}^T &= \mathbf{p}_{\mathbf{x}_L}^T \mathbf{K}_F \\
&= \hat{\mathbf{s}}_{\mathbf{x}_L}^T \mathbf{A}_F + \mathbf{e}_{v_{F,\mathbf{x}_L}}^T,
\end{aligned}$$

where $\left\|\mathbf{e}_{F,\mathbf{x}_L}\right\|_\infty \leq \left\|\mathbf{K}_F^T\right\|_\infty \left\|\mathbf{e}_{p_{\mathbf{x}_L}}\right\|_\infty \leq \sigma_E \gamma_B^{2L+1} m_B^{L+\frac{1}{2}} \lambda^{L+1}$ holds.

As $\mathbf{r}_{H,\mathbf{x}_L} = \mathsf{PRF}_H(K_H, \mathbf{x})$ holds from the correctness of PRO, we can show that $\mathbf{z}_{\mathbf{x}_L}^T \overset{c}{\approx} \lceil \frac{q}{2} \rfloor C(\mathbf{x}_L)$ as follows.

$$\begin{aligned}
\mathbf{z}_{\mathbf{x}_L}^T &= \mathbf{c}_{F,\mathbf{x}_L}^T - \mathbf{v}_{F,\mathbf{x}_L}^T - \left\lceil \frac{q}{2} \right\rfloor \mathbf{r}_{H,\mathbf{x}_L} \\
&= \hat{\mathbf{s}}_{\mathbf{x}_L}^T \mathbf{A}_F + (\left\lceil \frac{q}{2} \right\rfloor C(\mathbf{x}_L) + \mathsf{PRF}_B(K_B, \mathbf{x}_L) + \left\lceil \frac{q}{2} \right\rfloor \mathsf{PRF}_H(K_H, \mathbf{x}_L)) \\
&+ \mathbf{e}_{c_{F,\mathbf{x}_L}} - \hat{\mathbf{s}}_{\mathbf{x}_L}^T \mathbf{A}_F - \mathbf{e}_{v_{F,\mathbf{x}_L}} - \left\lceil \frac{q}{2} \right\rfloor \mathbf{r}_{H,\mathbf{x}_L} \\
&= \left\lceil \frac{q}{2} \right\rfloor C(\mathbf{x}_L) + \mathsf{PRF}_B(K_B, \mathbf{x}_L) + \mathbf{e}_{c_{F,\mathbf{x}_L}} - \mathbf{e}_{v_{F,\mathbf{x}_L}}
\end{aligned}$$

The norm $\left\| \mathbf{e}_{c_{F,\mathbf{x}_L}} - \mathbf{e}_{v_{F,\mathbf{x}_L}} \right\|_\infty$ is upper-bounded by $B := B_F + \sigma_E \gamma_B^{2L+1} m_B^{L+\frac{1}{2}} \lambda^{L+1}$. Since the output of PRF_B is upper-bounded by $[-\frac{q}{4} + B, \frac{q}{4} - B]$, it holds that

$$\left\| \mathsf{PRF}_B(K_B, \mathbf{x}_L) + \mathbf{e}_{c_{F,\mathbf{x}_L}} - \mathbf{e}_{v_{F,\mathbf{x}_L}} \right\|_\infty \leq \frac{q}{4}.$$

Hence, for every $i \in [\ell]$, y_i is 0—i.e., $z_i \in [-\frac{q}{4}, \frac{q}{4})$—if $C(\mathbf{x}_L)[j] = 0$ and 1 otherwise. This completes the proof. □

4.4 Security Proof

Before starting the security proof of our iO construction, we define our new lattice assumption called all-product LWE assumption in Assumption 3. In a nutshell, the all-product LWE assumption considers the following variant of the LWE instances

$$\hat{\mathbf{s}}_{\mathbf{x}_i}^T (\mathbf{A}_i - \left(\mathbf{x}_i^T, \mathbf{0}_{L-i}\right) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{A,\mathbf{x}_i}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T \mathbf{B}_i + \mathbf{e}_{B,\mathbf{x}_i}^T,$$

where we recall that $\hat{\mathbf{s}}_{\mathbf{x}_i}^T := \left(\bar{\mathbf{s}}^T \prod_{j \in [i]} \bar{\mathbf{R}}_{x_{i,j}}, -1\right)$. The assumption claims that even if the adversary can see $\Sigma_{i \in [L]} 2^i$ patterns of them, each of which corresponds to $\mathbf{x}_i \in \{0,1\}^i$ for $i \in [L]$, these instances are pseudorandom.

Assumption 3 (All-Product LWE Assumption). Let $n, \{m_{B,i}\}_{i \in [0,L]}, q, L \in \mathbb{N}$ be parameters and λ be a security parameter. Let $\{\sigma_{A,i}\}_{i \in [L]}$ and $\{\sigma_{B,i}\}_{i \in [0,L]}$ be Gaussian parameters. We define the following two distributions:

$$\mathcal{D}_{\mathsf{APLWE},0} := \left(\begin{array}{l} \{\mathbf{A}_i\}_{i \in [L]}, \{\mathbf{B}_i\}_{i \in [0,L]}, \{\bar{\mathbf{R}}_b\}_{b \in \{0,1\}}, \\ \{\mathbf{c}_{A,\mathbf{x}_i} := \hat{\mathbf{s}}_{\mathbf{x}_i}^T (\mathbf{A}_i - \left(\mathbf{x}_i^T, \mathbf{0}_{L-i}\right) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{A,\mathbf{x}_i}^T\}_{\substack{i \in [L], \\ \mathbf{x}_i \in \{0,1\}^i}} \\ \{\mathbf{c}_{B,\mathbf{x}_i} := \hat{\mathbf{s}}_{\mathbf{x}_i}^T \mathbf{B}_i + \mathbf{e}_{B,\mathbf{x}_i}^T\}_{\substack{i \in [0,L], \\ \mathbf{x}_i \in \{0,1\}^i}} \end{array} \right) \tag{3}$$

$$\mathcal{D}_{\mathsf{APLWE},1} := \left(\begin{array}{l} \{\mathbf{A}_i\}_{i \in [L]}, \{\mathbf{B}_i\}_{i \in [0,L]}, \{\bar{\mathbf{R}}_b\}_{b \in \{0,1\}}, \\ \{\mathbf{c}_{A,\mathbf{x}_i} \leftarrow\$ \mathbb{Z}_q^{m_{A,i}}\}_{\substack{i \in [L], \\ \mathbf{x}_i \in \{0,1\}^i}} \\ \{\mathbf{c}_{B,\mathbf{x}_i} \leftarrow\$ \mathbb{Z}_q^{m_{B,i}}\}_{\substack{i \in [0,L], \\ \mathbf{x}_i \in \{0,1\}^i}} \end{array} \right), \tag{4}$$

where

$$\begin{aligned}
&\bar{\mathbf{s}} \leftarrow\$ \{0,1\}^n, \mathbf{A}_i \leftarrow\$ \mathbb{Z}_q^{(n+1) \times L(n+1)\lceil \log_2 q \rceil}, \mathbf{B}_i \leftarrow\$ \mathbb{Z}_q^{(n+1) \times m_{B,i}}, \\
&\bar{\mathbf{R}}_b \leftarrow\$ \{0,1\}^{n \times n}, \mathbf{e}_{A,\mathbf{x}_i} \leftarrow\$ \mathcal{D}_{\mathbb{Z},\sigma_{A,i}}^{L(n+1)\lceil \log_2 q \rceil}, \mathbf{e}_{B,\mathbf{x}_i} \leftarrow\$ \mathcal{D}_{\mathbb{Z},\sigma_{B,i}}^{m_{B,i}}, \\
&\hat{\mathbf{s}}_{\mathbf{x}_i}^T := \left(\bar{\mathbf{s}}^T \prod_{j \in [i]} \bar{\mathbf{R}}_{x_{i,j}}, -1\right).
\end{aligned}$$

For a function $\kappa = \kappa(\lambda)$, the all-product LWE assumption is said to hold, if for every non-uniform adversary $\mathcal{A}$ such that $\mathrm{Size}(\mathcal{A}) \leq \mathsf{poly}(\kappa)$, it holds that

$$\left|\Pr\left[\mathcal{A}\left(\mathcal{D}_{\mathsf{APLWE},0}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathcal{D}_{\mathsf{APLWE},1}\right) = 1\right]\right| \leq \mathsf{negl}(\kappa).$$

We first show that if all preimages in the obfuscated circuit are multiplied by the corresponding LWE instances, then the resulting distribution is indistinguishable from the distribution that is independent of the secret keys $\mathbf{s}$, $\mathbf{t}$ and the original circuit C in Lemma 6. This lemma is later used in Theorem 2 to claim that the preconditions required in Lemma 3 hold.

Lemma 6. Given the prarameters defined for our iO construction in Subsection 4.2, we define the following two distributions for a circuit $C : \{0,1\}^L \to \{0,1\}^\ell$:

$$\mathcal{D}_{C,\mathsf{real}} := \left(\begin{array}{l} \{\mathbf{R}_b\}_{b\in\{0,1\}}, h, \{\mathbf{A}_{\mathsf{att},i}\}_{i\in[0,L]}, \{\mathbf{A}_{t,t}\}_{i\in[0,L]}, \\ \mathbf{A}_{\mathsf{fhe}} := \begin{pmatrix} \bar{\mathbf{A}}_{\mathsf{fhe}} \\ \bar{\mathbf{t}}^T\bar{\mathbf{A}}_{\mathsf{fhe}} + \mathbf{e}_{\mathsf{fhe}}^T \end{pmatrix}, \mathbf{X} := \mathbf{A}_{\mathsf{fhe}}\mathbf{R}_{\mathsf{fhe}} - (C, K_B, K_H) \otimes \mathbf{G}_{n+1}, \\ \mathbf{c}_{\mathsf{att},\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T(\mathbf{A}_{\mathsf{att},0} - (1, \mathsf{bits}(\mathbf{X}), \mathbf{0}_L) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{\mathsf{att},\epsilon}}^T, \\ \mathbf{c}_{t,\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T(\mathbf{A}_{t,0} - \mathbf{t}^T \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{t,\epsilon}}^T, \\ \mathbf{p}_\epsilon^T := (\hat{\mathbf{s}}_\epsilon^T, \hat{\mathbf{s}}_\epsilon^T)\mathbf{B}_{0,\star} + \mathbf{e}_{\mathbf{p}_\epsilon}^T, \{\mathbf{q}_{\mathbf{x}_i}^T := (\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T)\mathbf{B}_{i,x_i} + \mathbf{e}_{q_{\mathbf{x}_i}}^T\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}, \\ \{\mathbf{p}_{\mathbf{x}_i}^T := (\hat{\mathbf{s}}_{\mathbf{x}_i}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T)\mathbf{B}_{i,\star} + \mathbf{e}_{p_{\mathbf{x}_i}}^T\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}, \\ \left\{\begin{array}{r} \mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T := -\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T\mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,x_i} \\ + \hat{\mathbf{s}}_{\mathbf{x}_i}^T(\mathbf{A}_{\mathsf{att},i} - (\mathbf{0}_{L_{\mathsf{fhe}}+i}, x_i, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{v_{\mathsf{att},\mathbf{x}_i}}^T \end{array}\right\}_{\substack{i\in[L],\\ \mathbf{x}_i\in\{0,1\}^i}} \\ \{\mathbf{v}_{t,\mathbf{x}_i}^T := -\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T\mathbf{A}_{t,i-1}\mathbf{T}_{t,i,x_i} + \hat{\mathbf{s}}_{\mathbf{x}_i}^T\mathbf{A}_{t,i} + \mathbf{e}_{v_{t,\mathbf{x}_i}}^T\}_{\substack{i\in[L],\\ \mathbf{x}_i\in\{0,1\}^i}}, \\ \{\mathbf{v}_{F,\mathbf{x}_L}^T := \hat{\mathbf{s}}_{\mathbf{x}_L}^T\mathbf{A}_F + \mathbf{e}_{v_{F,\mathbf{x}_L}}^T\}_{\mathbf{x}_L\in\{0,1\}^L}, \{C(\mathbf{x}_L)\}_{\mathbf{x}_L\in\{0,1\}^L} \end{array}\right),$$

$$\mathcal{D}_{C,\mathsf{sim}} := \left(\begin{array}{l} \{\mathbf{R}_b\}_{b\in\{0,1\}}, \mathbf{A}_{\mathsf{fhe}}, h, \{\mathbf{A}_{\mathsf{att},i}\}_{i\in[0,L]}, \{\mathbf{A}_{t,t}\}_{i\in[0,L]}, \mathbf{A}_{\mathsf{fhe}} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times m}, \\ \mathbf{X} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times(|C|+2\lambda)m}, \mathbf{c}_{\mathsf{att},\epsilon}^T \leftarrow\$ \mathbb{Z}_q^{1\times L'm}, \mathbf{c}_{t,\epsilon}^T \leftarrow\$ \mathbb{Z}_q^{1\times(n+1)m}, \\ \mathbf{p}_\epsilon^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}, \{\mathbf{q}_{\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}, \{\mathbf{p}_{\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}, \\ \{\mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times L'm}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}, \{\mathbf{v}_{t,\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times(n+1)m}\}_{\substack{i\in[L],\\ \mathbf{x}_i\in\{0,1\}^i}}, \\ \{\mathbf{v}_{F,\mathbf{x}_L}^T \leftarrow\$ \mathbb{Z}_q^{1\times\ell}\}_{\mathbf{x}_L\in\{0,1\}^L}, \{C(\mathbf{x}_L)\}_{\mathbf{x}_L\in\{0,1\}^L} \end{array}\right),$$

where $\hat{\mathbf{s}}_{\mathbf{x}_i}, \mathbf{R}_b, \bar{\mathbf{A}}_{\mathsf{fhe}}, \bar{\mathbf{t}}^T, h, \{\mathbf{A}_{\mathsf{att},i}, \mathbf{A}_{t,i}\}_{i\in[0,L]}, \mathbf{A}_F, \{\mathbf{B}_{i,b}\}_{i\in[0,L],b\in\{0,1,\star\}}$, and $\{\mathbf{T}_{\mathsf{att},i,b}, \mathbf{T}_{t,i,b}\}_{i\in[L],b\in\{0,1\}}$ are defined in the same manner as the construction in Subsection 4.2, and $\mathbf{A}_F$ dependes on the provided C. Here, the errors are sampled as follows:

$$\begin{array}{l} \mathbf{e}_{c_{\mathsf{att},\epsilon}} \leftarrow\$ \mathcal{D}_{\mathbb{Z},\sigma_E}^{L'm}, \mathbf{e}_{c_{t,\epsilon}} \leftarrow\$ \mathcal{D}_{\mathbb{Z},\sigma_E}^{(n+1)m}, \mathbf{e}_{\mathbf{p}_\epsilon} \leftarrow\$ \mathcal{D}_{\mathbb{Z},\sigma_E}^{m_B}, \mathbf{e}_{\mathsf{fhe}}^T \leftarrow\$ \mathcal{D}_{\mathbb{Z},\sigma_E}^{m} \\ \{\mathbf{e}_{q_{\mathbf{x}_i}} \leftarrow\$ \mathcal{D}_{\mathbb{Z},\chi}^{m_B}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}\{\mathbf{e}_{p_{\mathbf{x}_i}} \leftarrow\$ \mathcal{D}_{\mathbb{Z},\chi}^{m_B}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i} \\ \{\mathbf{e}_{v_{\mathsf{att},\mathbf{x}_i}} \leftarrow\$ \mathcal{D}_{\mathbb{Z},\chi_i}^{L'm}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}\{\mathbf{e}_{v_{t,\mathbf{x}_i}} \leftarrow\$ \mathcal{D}_{\mathbb{Z},\chi_i}^{(n+1)m}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i} \\ \{\mathbf{e}_{v_{F,\mathbf{x}_i}} \leftarrow\$ \mathcal{D}_{\mathbb{Z},\chi_F}^{\ell}\}_{\mathbf{x}_L\in\{0,1\}^L}, \end{array}$$

for the Gaussian parameter σ_E, $\chi := \sqrt{2}\sigma_B$ defined in Subsection 4.2 and some other Gaussian

parameters $\{\chi_i\}_{i\in[L]}, \chi_F$ that satisfy

$$\begin{aligned}
&B_{c_0} := \sigma_E\sqrt{\lambda}\\
&B_{c_i} := mB_{c_{i-1}} + \sigma_{\chi_i}\sqrt{\lambda}\\
&\chi_i \geq (mB_{c_{i-1}})\lambda^{\omega(1)}\\
&B_F := \left(mB_{c_L}(n+1)\ell\lceil\log_2 q\rceil + \sigma_E\sqrt{\lambda}\right)\lceil\log_2 q\rceil(m+2)^{\mathsf{dep}'+1}\\
&\chi_F \geq B_F\lambda^{\omega(1)}.
\end{aligned}$$

Assuming the LWE assumption in Definition 1 and all-product LWE assumption in Definition 3, there is a function $\kappa = \kappa(\lambda)$ such that for every non-uniform adversary $\mathcal{A}$ and any circuit C provided by $\mathcal{A}$, it holds that

$$\left|\Pr\left[\mathcal{A}\left(\mathcal{D}_{C,\mathsf{real}}\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathcal{D}_{C,\mathsf{sim}}\right) = 1\right]\right| \leq \mathsf{negl}(\kappa).$$

Proof. We define the following sequence of hybrids.

Hyb_0. This is same as $\mathcal{D}_{C,\mathsf{real}}$.

For every $i \in [L]$ and $\mathbf{x}_i \in \{0,1\}^i$, $\mathsf{Hyb}_{i,\mathbf{x}_i}$ is defined as follows.

$\mathsf{Hyb}_{i,\mathbf{x}_i}$. This is same as the previous hybrid—namely, $\mathsf{Hyb}_{i-1,\mathbf{1}_{i-1}}$ if $\mathbf{x}_i = \mathbf{0}_i$ and $\mathsf{Hyb}_{i,\mathbf{x}_i-1}$ otherwise—except that for every $i \in [0,L]$ and $\mathbf{x}_i \in \{0,1\}^i$, $\mathbf{v}_{\mathsf{att},\mathbf{x}_i}$ and $\mathbf{v}_{t,\mathbf{x}_i}$ are replaced as follows, respectively.

$$\begin{aligned}
\mathbf{v}^T_{\mathsf{att},\mathbf{x}_i} &= -\mathbf{c}_{\mathsf{att},\mathbf{x}_{i-1}}\mathbf{T}_{\mathsf{att},i,x_i} + \hat{\mathbf{s}}^T_{\mathbf{x}_i}\left(\mathbf{A}_{\mathsf{att},i} - (1,\mathsf{bits}(\mathbf{X}),\mathbf{x}_i,\mathbf{0}_{L-i})\otimes\mathbf{G}_{n+1}\right) + \mathbf{e}^T_{v_{\mathsf{att},\mathbf{x}_i}}\\
\mathbf{v}^T_{t,\mathbf{x}_i} &= -\mathbf{c}_{t,\mathbf{x}_{i-1}}\mathbf{T}_{t,i,x_i} + \hat{\mathbf{s}}^T_{\mathbf{x}_i}\left(\mathbf{A}_{\mathsf{att},i} - \mathbf{t}^T\otimes\mathbf{G}_{n+1}\right) + \mathbf{e}^T_{v_{t,\mathbf{x}_i}}
\end{aligned}$$

Hyb_{L+1}. This is same as $\mathsf{Hyb}_{L,\mathbf{1}_L}$ except that for every $\mathbf{x}_L \in \{0,1\}^L$, $\mathbf{v}^T_{F,\mathbf{x}_L}$ is replaced with

$$\mathbf{v}^T_{F,\mathbf{x}_L} = \mathbf{c}^T_{F,\mathbf{x}_L} - \left(\left\lceil\frac{q}{2}\right\rfloor C(\mathbf{x}_L) + \mathsf{PRF}_B(K_B,\mathbf{x}_L) + \left\lceil\frac{q}{2}\right\rfloor\mathsf{PRF}_H(K_H,\mathbf{x}_L)\right) + \mathbf{e}^T_{v_{F,\mathbf{x}_L}}.$$

Hyb_{L+2}. This is same as Hyb_{L+1} except that we sample

$$\begin{aligned}
&\mathbf{c}^T_{\mathsf{att},\epsilon} \leftarrow\$ \mathbb{Z}_q^{1\times L'm}, \mathbf{c}^T_{t,\epsilon} \leftarrow\$ \mathbb{Z}_q^{1\times(n+1)m}, \mathbf{p}^T_\epsilon \leftarrow\$ \mathbb{Z}_q^{1\times m_B},\\
&\{\mathbf{q}^T_{\mathbf{x}_i} \leftarrow\$ \mathbb{Z}_q^{1\times m_B}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}, \{\mathbf{p}^T_{\mathbf{x}_i} \leftarrow\$ \mathbb{Z}_q^{1\times m_B}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i},\\
&\{\mathbf{v}^T_{\mathsf{att},\mathbf{x}_i} \leftarrow\$ \mathbb{Z}_q^{1\times L'm}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i},\\
&\{\mathbf{v}^T_{t,\mathbf{x}_i} \leftarrow\$ \mathbb{Z}_q^{1\times(n+1)m}\}_{\substack{i\in[L],\\ \mathbf{x}_i\in\{0,1\}^i}},
\end{aligned}$$

Hyb_{L+3}. This is same as Hyb_{L+2} except that $\mathbf{A}_{\mathsf{fhe}}$ is replaced with

$$\mathbf{A}_{\mathsf{fhe}} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times m}.$$

Hyb_{L+4}. This is same as Hyb_{L+3} except that $\mathbf{X}$ is replaced with

$$\mathbf{X} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times(|C|+2\lambda)m}.$$

Hyb_{L+5}. This is same as Hyb_{L+4} except that for every $\mathbf{x}_L \in \{0,1\}^L$, $\mathbf{v}^T_{F,\mathbf{x}_L}$ is replcaed with

$$\mathbf{v}^T_{F,\mathbf{x}_L} = \mathbf{c}^T_{F,\mathbf{x}_L} - \left\lceil\frac{q}{2}\right\rfloor(C(\mathbf{x}_L) + \mathsf{PRF}_H(K_H,\mathbf{x}_L)) + \mathbf{r}_{B,\mathbf{x}_L} + \mathbf{e}^T_{v_{F,\mathbf{x}_L}},$$

where $\mathbf{r}_{B,\mathbf{x}_L} \leftarrow\$ [-\frac{q}{4}+B, \frac{q}{4}-B]^{1\times\ell}$.

Hyb_{L+6}. This is same as Hyb_{L+5} except that for every $\mathbf{x}_L \in \{0,1\}^L$, $\mathbf{r}_{B,\mathbf{x}_L}$ is replaced with

$$\mathbf{r}_{B,\mathbf{x}_L} \leftarrow\!\$\ [-\frac{q}{4}, \frac{q}{4}]^{1\times\ell}.$$

Hyb_{L+7}. This is same as Hyb_{L+6} except that for every $\mathbf{x}_L \in \{0,1\}^L$, $\mathsf{PRF}_H(K_H, \mathbf{x}_L)$ is replcaed with $\mathbf{r}_{H,\mathbf{x}_L} \leftarrow\!\$\ \{0,1\}^\ell$.

Hyb_{L+8}. This is same as Hyb_{L+8} except that for every $\mathbf{x}_L \in \{0,1\}^L$, $\mathbf{v}^T_{F,\mathbf{x}_L}$ is replaced with $\mathbf{v}_{F,\mathbf{x}_L} \leftarrow\!\$\ \mathbb{Z}_q^\ell$. We note that the last hybrid is identical to $\mathcal{D}_{C,\mathsf{sim}}$.

We prove that each consecutive hybrids are indistinguishable.

Indistinguishability between $\mathsf{Hyb}_{i,\mathbf{x}_i}$ and its previous hybrid. Since $\mathbf{c}^T_{\mathsf{att},\mathbf{x}_{i-1}}\mathbf{T}_{\mathsf{att},i,x_i} = \hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}\mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,x_i} - \hat{\mathbf{s}}^T_{\mathbf{x}_i}((1,\mathsf{bits}(\mathbf{X}),\mathbf{x}^T_{i-1},\mathbf{0}_{L-i+1})\otimes\mathbf{G}_{n+1}) + \mathbf{e}^T_{c_{\mathsf{att},\mathbf{x}_{i-1}}}\mathbf{T}_{\mathsf{att},i,x_i}$, it holds that

$$\begin{aligned}
\mathbf{v}^T_{\mathsf{att},\mathbf{x}_i} =& -\hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}\mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,x_i} + \hat{\mathbf{s}}^T_{\mathbf{x}_i}((1,\mathsf{bits}(\mathbf{X}),\mathbf{x}^T_{i-1},\mathbf{0}_{L-i+1})\otimes\mathbf{G}_{n+1})\\
&-\mathbf{e}^T_{c_{\mathsf{att},\mathbf{x}_{i-1}}}\mathbf{T}_{\mathsf{att},i,x_i} + \hat{\mathbf{s}}^T_{\mathbf{x}_i}\left(\mathbf{A}_{\mathsf{att},i} - (1,\mathsf{bits}(\mathbf{X}),\mathbf{x}^T_i,\mathbf{0}_{L-i})\otimes\mathbf{G}_{n+1}\right) + \mathbf{e}^T_{v_{\mathsf{att},\mathbf{x}_i}}\\
=& -\hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}\mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,x_i} + \hat{\mathbf{s}}^T_{\mathbf{x}_i}\left(\mathbf{A}_{\mathsf{att},i} - (\mathbf{0}_{L_{\mathsf{fhe}}+i},x_i,\mathbf{0}_{L-i})\otimes\mathbf{G}_{n+1}\right)\\
&-\mathbf{e}^T_{c_{\mathsf{att},\mathbf{x}_{i-1}}}\mathbf{T}_{\mathsf{att},i,x_i} + \mathbf{e}^T_{v_{\mathsf{att},\mathbf{x}_i}}.
\end{aligned}$$

Here, the norm $\left\|-\mathbf{e}^T_{c_{\mathsf{att},\mathbf{x}_{i-1}}}\mathbf{T}_{\mathsf{att},i,x_i}\right\|_\infty$ is upper-bounded by $mB_{c_{i-1}}$. Therefore, by the definition of χ_i, noise flooding (Lemma 2) follows

$$-\mathbf{e}^T_{c_{\mathsf{att},\mathbf{x}_{i-1}}}\mathbf{T}_{\mathsf{att},i,x_i} + \mathbf{e}^T_{v_{\mathsf{att},\mathbf{x}_i}} \stackrel{\mathrm{s}}{\approx} \mathbf{e}^T_{v_{\mathsf{att},\mathbf{x}_i}}.$$

Similarly, it holds that

$$\begin{aligned}
\mathbf{v}^T_{\mathsf{att},\mathbf{x}_i} =& -\hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}\mathbf{A}_{t,i-1}\mathbf{T}_{t,i,x_i} + \hat{\mathbf{s}}^T_{\mathbf{x}_i}\mathbf{A}_{t,i} - \mathbf{e}^T_{c_{t,\mathbf{x}_{i-1}}}\mathbf{T}_{t,i,x_i} + \mathbf{e}^T_{v_{t,\mathbf{x}_i}}\\
\stackrel{\mathrm{s}}{\approx}& -\hat{\mathbf{s}}^T_{\mathbf{x}_{i-1}}\mathbf{A}_{t,i-1}\mathbf{T}_{t,i,x_i} + \hat{\mathbf{s}}^T_{\mathbf{x}_i}\mathbf{A}_{t,i} + \mathbf{e}^T_{v_{t,\mathbf{x}_i}}.
\end{aligned}$$

Hence, $\mathsf{Hyb}_{i,\mathbf{x}_i}$ is statistically indistinguishable from the previous hybrid.

Indistinguishability between $\mathsf{Hyb}_{L,\mathbf{1}_L}$ and Hyb_{L+1}. By the definition of $\mathbf{c}^T_{F,\mathbf{x}_L}$ in Equation 1, it holds that

$$\begin{aligned}
\mathbf{v}^T_{F,\mathbf{x}_L} =& \mathbf{c}^T_{F,\mathbf{x}_L} - \left((\left\lceil\frac{q}{2}\right\rfloor C(\mathbf{x}_L) + \mathsf{PRF}_B(K_B,\mathbf{x}_L) + \left\lceil\frac{q}{2}\right\rfloor\mathsf{PRF}_H(K_H,\mathbf{x}_L))\right) + \mathbf{e}^T_{v_{F,\mathbf{x}_L}}\\
=& \hat{\mathbf{s}}^T_{\mathbf{x}_L}\mathbf{A}_F + \mathbf{e}^T_{c_{F,\mathbf{x}_L}} + \mathbf{e}^T_{v_{F,\mathbf{x}_L}}
\end{aligned}$$

In similar to Inequation 2, $\left\|\mathbf{e}_{c_{F,\mathbf{x}_L}}\right\|_\infty$ is upper-bounded by B_F:

$$\begin{aligned}
\left\|\mathbf{e}_{c_{F,\mathbf{x}_L}}\right\|_\infty \le& m\left\|\left((\mathbf{e}^T_{c_{\mathsf{att},\mathbf{x}_L}},\mathbf{e}^T_{c_{t,\mathbf{x}_L}})\mathbf{H}_{F,\mathbf{X}}\right)^T\right\|_\infty + \left\|\mathbf{R}^T_f\mathbf{e}_{\mathsf{fhe}}\right\|_\infty\\
\le& mB_{c_L}\left((n+1)\ell\lceil\log_2 q\rceil^2(m+2)^{\mathsf{dep}'+1}\right)\\
&+\sigma_E\sqrt{\lambda}(m+2)^{\mathsf{dep}+1}\lceil\log_2 q\rceil\\
\le& B_F.
\end{aligned}$$

Therefore, by the choise of χ_F, noise flooding (Lemma 2) follows

$$\mathbf{e}^T_{c_{F,\mathbf{x}_L}} + \mathbf{e}^T_{v_{F,\mathbf{x}_L}} \stackrel{\mathrm{s}}{\approx} \mathbf{e}^T_{v_{F,\mathbf{x}_L}}.$$

Hence, Hyb_{L+1} is statistically indistinguishable from $\mathsf{Hyb}_{L,\mathbf{1}_L}$.

Indistinguishability between Hyb_{L+1} and Hyb_{L+2}. We prove that if there is an adversary $\mathcal{A}$ that can distinguish between Hyb_{L+1} and Hyb_{L+2}, then we can construct an adversary $\mathcal{B}$ that can break the security of the all-product LWE assumption (Definition 3) with non-negligible advantage. We define the parameters of the all-product LWE assumption as below:

$$\begin{aligned}
&n := n, m_{B,0} := (1+n+L')m + 4m_B \\
&\{m_{B,i} := (2+L_{\mathsf{fhe}}+n)m + 6m_B\}_{i\in[1,L-1]}, \\
&m_{B,L} := 4m_B, q := q, L := L, \lambda := \lambda, \\
&\{\sigma_{A,i} := \sigma_E\}_{i\in[L]}, \sigma_{B,0} := \sigma_E, \{\sigma_{B,i} := \chi\}_{i\in[L]}
\end{aligned}$$

$\mathcal{B}$ internally invokes $\mathcal{A}$ as follows.

1. $\mathcal{A}$ provides $\mathcal{B}$ a circuit C.
2. $\mathcal{B}$ receives the following data from the challenger of the security game for the all-product LWE assumption.

$$\begin{pmatrix} \{\mathbf{A}_i\}_{i\in[L]}, \{\mathbf{B}_i\}_{i\in[0,L]}\{\bar{\mathbf{R}}_b\}_{b\in\{0,1\}}, \\ \{\mathbf{c}_{A,\mathbf{x}_i}\}_{\substack{i\in[L],\\ \mathbf{x}_i\in\{0,1\}^i}}, \{\mathbf{c}_{B,\mathbf{x}_i}\}_{\substack{i\in[0,L],\\ \mathbf{x}_i\in\{0,1\}^i}}, \end{pmatrix} \tag{5}$$

   In the following, we let $b_1 = 0$ if the challenger samples them from $\mathcal{D}_{\mathsf{APLWE},0}$ and $b_1 = 1$ otherwise.

3. $\mathcal{B}$ parses the provided data as follows.

   - For every $i \in [L]$, $\mathbf{A}_i \in \mathbb{Z}_q^{(n+1)\times Lm}$ is parsed as $\hat{\mathbf{A}}_{\mathsf{att},i,2} \in \mathbb{Z}_q^{(n+1)\times Lm}$.
   - $\mathbf{B}_0 \in \mathbb{Z}_q^{(n+1)\times((1+n+L')m+4m_B)}$ is parsed as
   $$\mathbf{B}_0 = (\hat{\mathbf{A}}_{\mathsf{att},0}, \hat{\mathbf{A}}_{t,0}, \hat{\mathbf{B}}_{0,\star,1}, \hat{\mathbf{B}}_{0,\star,2}, \hat{\mathbf{B}}_{1,0,1}, \hat{\mathbf{B}}_{1,1,1}),$$
   where $\hat{\mathbf{A}}_{\mathsf{att},0} \in \mathbb{Z}_q^{(n+1)\times L'm}$, $\hat{\mathbf{A}}_{t,0} \in \mathbb{Z}_q^{(n+1)\times(n+1)m}$, $\hat{\mathbf{B}}_{0,\star,1} \in \mathbb{Z}_q^{(n+1)\times m_B}$, $\hat{\mathbf{B}}_{0,\star,2} \in \mathbb{Z}_q^{(n+1)\times m_B}$, $\hat{\mathbf{B}}_{1,0,1} \in \mathbb{Z}_q^{(n+1)\times m_B}$, and $\hat{\mathbf{B}}_{1,1,1} \in \mathbb{Z}_q^{(n+1)\times m_B}$.
   - For every $i \in [1, L-1]$, $\mathbf{B}_i \in \mathbb{Z}_q^{(n+1)\times((2+L_{\mathsf{fhe}}+n)m+6m_B)}$ is parsed as
   $$\mathbf{B}_i = (\hat{\mathbf{A}}_{\mathsf{att},i,1}, \hat{\mathbf{A}}_{t,i}, \hat{\mathbf{B}}_{i,0,2}, \hat{\mathbf{B}}_{i,1,2}, \hat{\mathbf{B}}_{i,\star,1}, \hat{\mathbf{B}}_{i,\star,2}, \hat{\mathbf{B}}_{i+1,0,1}, \hat{\mathbf{B}}_{i+1,1,1}),$$
   where $\hat{\mathbf{A}}_{\mathsf{att},i,1} \in \mathbb{Z}_q^{(n+1)\times(1+L_{\mathsf{fhe}})m}$, $\hat{\mathbf{A}}_{t,i} \in \mathbb{Z}_q^{(n+1)\times(n+1)m}$, $\hat{\mathbf{B}}_{i,0,2} \in \mathbb{Z}_q^{(n+1)\times m_B}$, $\hat{\mathbf{B}}_{i,1,2} \in \mathbb{Z}_q^{(n+1)\times m_B}$, $\hat{\mathbf{B}}_{i,\star,1} \in \mathbb{Z}_q^{(n+1)\times m_B}$, $\hat{\mathbf{B}}_{i,\star,2} \in \mathbb{Z}_q^{(n+1)\times m_B}$, $\hat{\mathbf{B}}_{i+1,0,1} \in \mathbb{Z}_q^{(n+1)\times m_B}$, and $\hat{\mathbf{B}}_{i+1,1,1} \in \mathbb{Z}_q^{(n+1)\times m_B}$.
   - $\mathbf{B}_L \in \mathbb{Z}_q^{n\times 4m_B}$ is parsed as
   $$\mathbf{B}_L = (\hat{\mathbf{B}}_{L,0,2}, \hat{\mathbf{B}}_{L,1,2}, \hat{\mathbf{B}}_{L,\star,1}, \hat{\mathbf{B}}_{L,\star,2}),$$
   where $\hat{\mathbf{A}}_{\mathsf{att},L,1} \in \mathbb{Z}_q^{(n+1)\times(1+L_{\mathsf{fhe}})m}$, $\hat{\mathbf{A}}_{t,L} \in \mathbb{Z}_q^{(n+1)\times(n+1)m}$, $\hat{\mathbf{B}}_{L,0,2} \in \mathbb{Z}_q^{(n+1)\times m_B}$, $\hat{\mathbf{B}}_{L,1,2} \in \mathbb{Z}_q^{(n+1)\times m_B}$, $\hat{\mathbf{B}}_{L,\star,1} \in \mathbb{Z}_q^{(n+1)\times m_B}$, and $\hat{\mathbf{B}}_{L,\star,2} \in \mathbb{Z}_q^{(n+1)\times m_B}$.
   - For every $i \in [L]$ and $\mathbf{x}_i \in \{0,1\}^i$, $\mathbf{c}_{A,\mathbf{x}_i}^T \in \mathbb{Z}_q^{1\times Lm}$ is parsed as $\hat{\mathbf{v}}_{\mathsf{att},i,2}^T \in \mathbb{Z}_q^{1\times Lm}$.
   - $\mathbf{c}_{B,\epsilon}^T \in \mathbb{Z}_q^{1\times((1+n+L')m+4m_B)}$ is parsed as
   $$\mathbf{c}_{B,\epsilon}^T = (\mathbf{c}_{\mathsf{att},\epsilon}^T, \mathbf{c}_{t,\epsilon}^T, \hat{\mathbf{p}}_{\epsilon,1}^T, \hat{\mathbf{p}}_{\epsilon,2}^T, \hat{\mathbf{q}}_{0,1}^T, \hat{\mathbf{q}}_{1,1}^T),$$
   where $\mathbf{c}_{\mathsf{att},\epsilon}^T \in \mathbb{Z}_q^{1\times L'm}$, $\mathbf{c}_{t,\epsilon}^T \in \mathbb{Z}_q^{1\times(n+1)m}$, $\hat{\mathbf{p}}_{\epsilon,1}^T \in \mathbb{Z}_q^{1\times m_B}$, $\hat{\mathbf{p}}_{\epsilon,2}^T \in \mathbb{Z}_q^{1\times m_B}$, $\hat{\mathbf{q}}_{0,1}^T \in \mathbb{Z}_q^{1\times m_B}$, and $\hat{\mathbf{q}}_{1,1}^T \in \mathbb{Z}_q^{1\times m_B}$.

- For every $i \in [1, L-1]$ and $\mathbf{x}_i \in \{0,1\}^i$, $\mathbf{c}_{B,\mathbf{x}_i}^T \in \mathbb{Z}_q^{1\times((2+L_{\mathsf{fhe}}+n)m+6m_B)}$ is parsed as

$$\mathbf{c}_{B,\mathbf{x}_i}^T = (\hat{\mathbf{v}}_{\mathsf{att},\mathbf{x}_i,1}^T, \hat{\mathbf{v}}_{t,\mathbf{x}_i}^T, \hat{\mathbf{q}}_{\mathbf{x}_{i-1}0,2}^T, \hat{\mathbf{q}}_{\mathbf{x}_{i-1}1,2}^T, \hat{\mathbf{p}}_{\mathbf{x}_i,1}^T, \hat{\mathbf{p}}_{\mathbf{x}_i,2}^T, \hat{\mathbf{q}}_{\mathbf{x}_i0,1}^T, \hat{\mathbf{q}}_{\mathbf{x}_i1,1}^T),$$

  where $\hat{\mathbf{v}}_{\mathsf{att},\mathbf{x}_i,1}^T \in \mathbb{Z}_q^{1\times(1+L_{\mathsf{fhe}})m}$, $\hat{\mathbf{v}}_{t,\mathbf{x}_i}^T \in \mathbb{Z}_q^{1\times(n+1)m}$, $\hat{\mathbf{q}}_{\mathbf{x}_{i-1}0,2}^T \in \mathbb{Z}_q^{1\times m_B}$, $\hat{\mathbf{q}}_{\mathbf{x}_{i-1}1,2}^T \in \mathbb{Z}_q^{1\times m_B}$, $\hat{\mathbf{p}}_{\mathbf{x}_i,1}^T \in \mathbb{Z}_q^{1\times m_B}$, $\hat{\mathbf{p}}_{\mathbf{x}_i,2}^T \in \mathbb{Z}_q^{1\times m_B}$, $\hat{\mathbf{q}}_{\mathbf{x}_i0,1}^T \in \mathbb{Z}_q^{1\times m_B}$, and $\hat{\mathbf{q}}_{\mathbf{x}_i1,1}^T \in \mathbb{Z}_q^{1\times m_B}$.
- For every $\mathbf{x}_L \in \{0,1\}^L$, $\mathbf{c}_{B,\mathbf{x}_L}^T \in \mathbb{Z}_q^{1\times((2+L_{\mathsf{fhe}}+n)m+4m_B)}$ is parsed as

$$\mathbf{c}_{B,\mathbf{x}_L}^T = (\hat{\mathbf{q}}_{\mathbf{x}_{L-1}0,1}^T, \hat{\mathbf{q}}_{\mathbf{x}_{L-1}1,1}^T, \hat{\mathbf{p}}_{\mathbf{x}_L,1}^T, \hat{\mathbf{p}}_{\mathbf{x}_L,2}^T),$$

  where $\hat{\mathbf{q}}_{\mathbf{x}_{L-1}0,1}^T \in \mathbb{Z}_q^{1\times m_B}$, $\hat{\mathbf{q}}_{\mathbf{x}_{L-1}1,1}^T \in \mathbb{Z}_q^{1\times m_B}$, $\hat{\mathbf{p}}_{\mathbf{x}_L,1}^T \in \mathbb{Z}_q^{1\times m_B}$, and $\hat{\mathbf{p}}_{\mathbf{x}_L,2}^T \in \mathbb{Z}_q^{1\times m_B}$.

4. $\mathcal{B}$ sets

$$\mathbf{A}_{\mathsf{att},0} = \hat{\mathbf{A}}_{\mathsf{att},0} + (1, \mathsf{bits}(\mathbf{X}), \mathbf{0}_L) \otimes \mathbf{G}_{n+1}.$$

5. For every $i \in [L]$, $\mathcal{B}$ sets

$$\mathbf{A}_{\mathsf{att},i} = \left(\hat{\mathbf{A}}_{\mathsf{att},i,1} + (1, \mathsf{bits}(\mathbf{X})) \otimes \mathbf{G}_n, \quad \hat{\mathbf{A}}_{\mathsf{att},i,2}\right),$$
$$\mathbf{A}_{t,i} = \hat{\mathbf{A}}_{t,i} + \mathbf{t}^T \otimes \mathbf{G}_n.$$

6. For every $i \in [L-1]$ and $\mathbf{x}_i \in \{0,1\}^i$, $\mathcal{B}$ sets

$$\mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T = -\mathbf{c}_{\mathsf{att},\mathbf{x}_{i-1}}^T \mathbf{T}_{\mathsf{att},i,x_i} + \left(\hat{\mathbf{v}}_{\mathsf{att},\mathbf{x}_i,1}^T, \quad \hat{\mathbf{v}}_{\mathsf{att},\mathbf{x}_i,2}^T\right),$$
$$\mathbf{v}_{t,\mathbf{x}_i}^T = -\mathbf{c}_{t,\mathbf{x}_{i-1}}^T \mathbf{T}_{t,i,x_i} + \hat{\mathbf{v}}_{t,\mathbf{x}_i}^T,$$

   where we assume that additional errors are injected into $\mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T$ and $\mathbf{v}_{t,\mathbf{x}_i}^T$, so that the total errors follow $\mathcal{D}_{\mathbb{Z},\chi_i}^{L'm}$ and $\mathcal{D}_{\mathbb{Z},\chi_i}^{(n+1)m}$, respectively.

7. For every $i \in [0, L]$ and $b \in \{0, 1, \star\}$, $\mathcal{B}$ sets

$$\mathbf{B}_{i,b} = \begin{pmatrix} \hat{\mathbf{B}}_{i,b,1} \\ \hat{\mathbf{B}}_{i,b,2} \end{pmatrix}.$$

8. $\mathcal{B}$ sets

$$\mathbf{p}_\epsilon^T = \hat{\mathbf{p}}_{\epsilon,1}^T + \hat{\mathbf{p}}_{\epsilon,2}^T.$$

9. For every $i \in [0, L]$ and $\mathbf{x}_i \in \{0,1\}^i$, $\mathcal{B}$ sets

$$\mathbf{p}_{\mathbf{x}_i}^T = \hat{\mathbf{p}}_{\mathbf{x}_i,1}^T + \hat{\mathbf{p}}_{\mathbf{x}_i,2}^T, \mathbf{q}_{\mathbf{x}_i}^T = \hat{\mathbf{q}}_{\mathbf{x}_i,1}^T + \hat{\mathbf{q}}_{\mathbf{x}_i,2}^T.$$

10. $\mathcal{B}$ sets the other data in the same manner as Hyb_{L+1}.
11. $\mathcal{B}$ sends the above data to $\mathcal{A}$.
12. $\mathcal{A}$ outputs a bit b', and $\mathcal{B}$ forwards it to the challenger of the security game for the all-product LWE assumption.

We claim that if $b_1 = 0$, i.e., the challenger samples the data from $\mathcal{D}_{\mathsf{APLWE},0}$, then the data sent to $\mathcal{A}$ simulates that in Hyb_{L+1}; otherwise, it simulates the data in Hyb_{L+2}. In the former case, we can confirm the following facts.

- It holds that

$$\begin{aligned}\mathbf{c}_{\mathsf{att},\epsilon}^T &= \hat{\mathbf{s}}_\epsilon^T \hat{\mathbf{A}}_{\mathsf{att},0} + \mathbf{e}_{c_{\mathsf{att},\epsilon}}^T \\ &= \hat{\mathbf{s}}_\epsilon^T \left(\mathbf{A}_{\mathsf{att},0} - (1, \mathsf{bits}(\mathbf{X}), \mathbf{0}_L) \otimes \mathbf{G}_{n+1}\right) + \mathbf{e}_{c_{\mathsf{att},\epsilon}}^T.\end{aligned}$$

  Therefore, $\mathbf{c}_{\mathsf{att},\epsilon}^T$ in Hyb_{L+1} is simulated when $b_1 = 0$.

- It holds that

$$\begin{aligned}\mathbf{c}_{t,\epsilon}^T &= \hat{\mathbf{s}}_\epsilon^T \hat{\mathbf{A}}_{t,0} + \mathbf{e}_{c_{t,\epsilon}}^T \\ &= \hat{\mathbf{s}}_\epsilon^T \left(\mathbf{A}_{t,0} - \mathbf{t}^T \otimes \mathbf{G}_{n+1}\right) + \mathbf{e}_{c_{t,\epsilon}}^T.\end{aligned}$$

  Therefore, $\mathbf{c}_{t,\epsilon}^T$ in Hyb_{L+1} is simulated when $b_1 = 0$.

- For every $i \in [L]$ and $\mathbf{x}_i \in \{0,1\}^i$, it holds that

$$\begin{aligned}\mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T &= -\mathbf{c}_{\mathsf{att},\mathbf{x}_{i-1}}^T \mathbf{T}_{\mathsf{att},i,x_i} + \left(\hat{\mathbf{v}}_{\mathsf{att},\mathbf{x}_i,1}^T, \quad \hat{\mathbf{v}}_{\mathsf{att},\mathbf{x}_i,2}^T\right) \\ &= -\mathbf{c}_{\mathsf{att},\mathbf{x}_{i-1}}^T \mathbf{T}_{\mathsf{att},i,x_i} + \hat{\mathbf{s}}_{\mathbf{x}_i}^T \left(\hat{\mathbf{A}}_{\mathsf{att},i,1}, \quad \hat{\mathbf{A}}_{\mathsf{att},i,2} - (\mathbf{x}_i^T, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1}\right) + \mathbf{e}_{\mathsf{att},\mathbf{x}_i}^T \\ &= -\mathbf{c}_{\mathsf{att},\mathbf{x}_{i-1}}^T \mathbf{T}_{\mathsf{att},i,x_i} + \hat{\mathbf{s}}_{\mathbf{x}_i}^T \left(\mathbf{A}_{\mathsf{att},i} - (1, \mathsf{bits}(\mathbf{X}), \mathbf{x}_i^T, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1}\right) + \mathbf{e}_{\mathsf{att},\mathbf{x}_i}^T.\end{aligned}$$

  Therefore, $\mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T$ in Hyb_{L+1} is simulated when $b_1 = 0$.

- For every $i \in [L]$ and $\mathbf{x}_i \in \{0,1\}^i$, it holds that

$$\begin{aligned}\mathbf{v}_{t,\mathbf{x}_i}^T &= -\mathbf{c}_{t,\mathbf{x}_{i-1}}^T \mathbf{T}_{t,i,x_i} + \hat{\mathbf{v}}_{t,\mathbf{x}_i}^T \\ &= -\mathbf{c}_{t,\mathbf{x}_{i-1}}^T \mathbf{T}_{t,i,x_i} + \hat{\mathbf{s}}_{\mathbf{x}_i}^T \hat{\mathbf{A}}_{t,i} + \mathbf{e}_{t,\mathbf{x}_i}^T \\ &= -\mathbf{c}_{t,\mathbf{x}_{i-1}}^T \mathbf{T}_{t,i,x_i} + \hat{\mathbf{s}}_{\mathbf{x}_i}^T \left(\mathbf{A}_{t,i} - \mathbf{t}^T \otimes \mathbf{G}_{n+1}\right) + \mathbf{e}_{t,\mathbf{x}_i}^T.\end{aligned}$$

  Therefore, $\mathbf{v}_{t,\mathbf{x}_i}^T$ in Hyb_{L+1} is simulated when $b_1 = 0$.

- It holds that

$$\begin{aligned}\mathbf{p}_\epsilon^T &= \hat{\mathbf{p}}_{\epsilon,1}^T + \hat{\mathbf{p}}_{\epsilon,2}^T \\ &= \hat{\mathbf{s}}_\epsilon^T \bar{\mathbf{B}}_{0,\star,1} + \mathbf{e}_{p_{\epsilon,1}}^T + \hat{\mathbf{s}}_\epsilon^T \hat{\mathbf{B}}_{0,\star,2} + \mathbf{e}_{p_{\epsilon,2}}^T \\ &= (\hat{\mathbf{s}}_\epsilon^T, \hat{\mathbf{s}}_\epsilon^T) \mathbf{B}_{0,\star} + \mathbf{e}_{p_\epsilon}^T.\end{aligned}$$

  Therefore, $\mathbf{p}_\epsilon^T$ in Hyb_{L+1} is simulated when $b_1 = 0$.

- For every $i \in [L]$ and $\mathbf{x}_i \in \{0,1\}^i$, it holds that

$$\begin{aligned}\mathbf{p}_{\mathbf{x}_i}^T &= \hat{\mathbf{p}}_{\mathbf{x}_i,1}^T + \hat{\mathbf{p}}_{\mathbf{x}_i,2}^T \\ &= \hat{\mathbf{s}}_{\mathbf{x}_i}^T \bar{\mathbf{B}}_{i,\star,1} + \mathbf{e}_{p_{\mathbf{x}_i,1}}^T + \hat{\mathbf{s}}_{\mathbf{x}_i}^T \hat{\mathbf{B}}_{i,\star,2} + \mathbf{e}_{p_{\mathbf{x}_i,2}}^T \\ &= (\hat{\mathbf{s}}_{\mathbf{x}_i}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T) \mathbf{B}_{i,\star} + \mathbf{e}_{p_{\mathbf{x}_i}}^T. \\ \mathbf{q}_{\mathbf{x}_i}^T &= \hat{\mathbf{q}}_{\mathbf{x}_i,1}^T + \hat{\mathbf{q}}_{\mathbf{x}_i,2}^T \\ &= \hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T \bar{\mathbf{B}}_{i,x_i,1} + \mathbf{e}_{q_{\mathbf{x}_i,1}}^T + \hat{\mathbf{s}}_{\mathbf{x}_i}^T \hat{\mathbf{B}}_{i,x_i,2} + \mathbf{e}_{q_{\mathbf{x}_i,2}}^T \\ &= (\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T) \mathbf{B}_{i,x_i} + \mathbf{e}_{q_{\mathbf{x}_i}}^T.\end{aligned}$$

  Therefore, $\mathbf{p}_{\mathbf{x}_i}^T$ and $\mathbf{q}_{\mathbf{x}_i}^T$ in Hyb_{L+1} are simulated when $b_1 = 0$.

Hence, Hyb_{L+1} is simulated when $b_1 = 0$.

In the latter case—namely, $b_1 = 1$—we can confirm the following facts.

- The $\mathbf{c}^T_{\mathsf{att},\epsilon}$ and $\mathbf{c}^T_{\mathsf{att},\epsilon}$ are uniformly random vectors.
- For every $i \in [L]$ and $\mathbf{x}_i \in \{0,1\}^i$, the randomness of $(\hat{\mathbf{v}}^T_{\mathsf{att},\mathbf{x},1}, \hat{\mathbf{v}}^T_{\mathsf{att},\mathbf{x},2})$ implies that of $\mathbf{v}^T_{\mathsf{att},\mathbf{x}}$.
- For every $i \in [L]$ and $\mathbf{x}_i \in \{0,1\}^i$, the randomness of $\hat{\mathbf{c}}^T_{t,\mathbf{x}}$ implies that of $\mathbf{v}^T_{t,\mathbf{x}}$.
- The randomness of $(\hat{\mathbf{p}}^T_{\epsilon,1}, \hat{\mathbf{p}}^T_{\epsilon,2})$ implies that of $\mathbf{p}^T_\epsilon$.
- For every $i \in [L]$ and $\mathbf{x}_i \in \{0,1\}^i$, the randomness of $(\hat{\mathbf{p}}^T_{\mathbf{x},1}, \hat{\mathbf{p}}^T_{\mathbf{x},2})$ implies that of $\mathbf{p}^T_\mathbf{x}$.
- For every $i \in [L]$ and $\mathbf{x}_i \in \{0,1\}^i$, the randomness of $(\hat{\mathbf{q}}^T_{\mathbf{x},1}, \hat{\mathbf{q}}^T_{\mathbf{x},2})$ implies that of $\mathbf{q}^T_\mathbf{x}$.

Hence, Hyb_{L+2} is simulated when $b_1 = 1$.

From the above discussion, it follows that $\mathcal{B}$ wins the security game for the all-product LWE assumption with non-negligible probability. Therefore, by contradiction, we can concluded that Hyb_{L+1} and Hyb_{L+2} are computationally indistinguishable.

Indistinguishability between Hyb_{L+2} and Hyb_{L+3}. The only difference between Hyb_{L+2} and Hyb_{L+3} is the last row of $\mathbf{A}_{\mathsf{fhe}}$, which is $\mathbf{b}^T_{\mathsf{fhe}} := \bar{\mathbf{t}}^T \bar{\mathbf{A}}_{\mathsf{fhe}} + \mathbf{e}^T_{\mathsf{fhe}}$ and $\mathbf{b}^T_{\mathsf{fhe}} \leftarrow\!\!\$\ \mathbb{Z}_q^{1\times m}$ in Hyb_{L+2} and Hyb_{L+3}, respectively. To prove their indistinguishability, we show that if there is an adversary $\mathcal{A}$ that can distinguish between Hyb_{L+2} and Hyb_{L+3}, then we can construct an adversary $\mathcal{B}$ that can break the security of the LWE assumption. $\mathcal{B}$ internally invokes $\mathcal{A}$ as follows.

- $\mathcal{A}$ provides $\mathcal{B}$ a circuit C.
- $\mathcal{B}$ recieves $\bar{\mathbf{A}}_{\mathsf{fhe}} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{b}_{\mathsf{fhe}} \in \mathbb{Z}_q^m$ from the challenger of the security game for the LWE assumption. In the following, we let $b_2 = 0$ if $\mathbf{b}^T_{\mathsf{fhe}} = \bar{\mathbf{t}}^T \bar{\mathbf{A}}_{\mathsf{fhe}} + \mathbf{e}^T_{\mathsf{fhe}}$ and $b_2 = 1$ otherwise.
- $\mathcal{B}$ sets $\mathbf{A}_{\mathsf{fhe}} = \begin{pmatrix} \bar{\mathbf{A}}_{\mathsf{fhe}} \\ \mathbf{b}_{\mathsf{fhe}} \end{pmatrix}$.
- $\mathcal{B}$ sets the other data in the same manner as Hyb_{L+2}, which is possible because there is no data except for $\mathbf{b}_{\mathsf{fhe}}$ that depends on $\hat{\mathbf{t}}$ in both hybrids.
- $\mathcal{B}$ sends the above data to $\mathcal{A}$.
- $\mathcal{A}$ outputs a bit b', and $\mathcal{B}$ forwards it to the challenger of the security game for the LWE assumption.

We can easily confirm that $\mathbf{A}_{\mathsf{fhe}}$ defined above simulates one in Hyb_{L+2} when $b_2 = 0$ and in Hyb_{L+3} when $b_2 = 1$. Therefore, $\mathcal{B}$ wins the security game for the LWE assumption with non-negligible probability. Hence, by contradiction, we can conclude that Hyb_{L+2} and Hyb_{L+3} are computationally indistinguishable.

Indistinguishability between Hyb_{L+3} and Hyb_{L+4}. The only difference between Hyb_{L+3} and Hyb_{L+4} is the distribution of $\mathbf{X}$, which is $\mathbf{X} = \mathbf{A}_{\mathsf{fhe}}\mathbf{R}_{\mathsf{fhe}} - (C, K_B, K_H) \otimes \mathbf{G}_{n+1}$ in Hyb_{L+2} and $\mathbf{X} \leftarrow\!\!\$\ \mathbb{Z}_q^{(n+1)\times(|C|+2\lambda)m}$ in Hyb_{L+3}. Since $\mathbf{A}_{\mathsf{fhe}} \in \mathbb{Z}_q^{(n+1)\times m}$ and $\mathbf{R}_{\mathsf{fhe}} \in \{0,1\}^{m\times(|C|+2\lambda)m}$ are uniformly random matrixes, the leftover hash lemma [HILL99, Reg09] implies that

$$\mathbf{A}_{\mathsf{fhe}}\mathbf{R}_{\mathsf{fhe}} \stackrel{\mathrm{s}}{\approx} \mathbf{U},$$

where $\mathbf{U} \leftarrow\!\!\$\ \mathbb{Z}_q^{(n+1)\times(|C|+2\lambda)m}$. Hence, Hyb_{L+3} and Hyb_{L+4} are statistically indistinguishable.

Indistinguishability between Hyb_{L+4} and Hyb_{L+5}. The only modification from Hyb_{L+4} to Hyb_{L+5} is that the output of $\mathsf{PRF}_B(K_B, \mathbf{x}_L)$ is replaced by the true randomness $\mathbf{r}_{B,\mathbf{x}_L}$ for every

$\mathbf{x}_L \in \{0,1\}^L$. Since K_B does not appear in the adversary's view in either hybrid, the security of PRF_B immidiately implies that Hyb_{L+4} and Hyb_{L+5} are indistinguishable.

Indistinguishability between Hyb_{L+5} and Hyb_{L+6}. The only difference between Hyb_{L+5} and Hyb_{L+6} is the range of $\mathbf{r}_{B,\mathbf{x}_L}$ for every $\mathbf{x}_L \in \{0,1\}^L$, which is

$$\begin{aligned}\mathcal{U}_1 =&[-\frac{q}{4}+B, \frac{q}{4}-B] \text{ in } \mathsf{Hyb}_{L+5}\\ \mathcal{U}_2 =&[-\frac{q}{4}, \frac{q}{4}] \text{ in } \mathsf{Hyb}_{L+6}.\end{aligned}$$

Since B is exponentially smaller than $\frac{q}{4}$ as defined in Subsection 4.2, the statistical distance between $\mathcal{U}_1$ and $\mathcal{U}_2$ is negligible. Therefore, Hyb_{L+5} and Hyb_{L+6} are statistically indistinguishable.

Indistinguishability between Hyb_{L+6} and Hyb_{L+7}. The only modification from Hyb_{L+6} to Hyb_{L+7} is that the output of $\mathsf{PRF}_H(K_H, \mathbf{x}_L)$, which is also the output of $\mathsf{O}(\mathsf{hEval}, h, \mathbf{x}_L)$, is replaced by the true randomness $\mathbf{r}_{H,\mathbf{x}_L} \leftarrow\!\!\$\ \{0,1\}^\ell$ for every $\mathbf{x}_L \in \{0,1\}^L$. Since K_H does not appear in the adversary's view in either hybrid, the security of PRF_H immidiately implies that Hyb_{L+6} and Hyb_{L+7} are indistinguishable.

Indistinguishability between Hyb_{L+7} and Hyb_{L+8}. We prove their indistinguishability by the same method present in Subsection 8.4 of [BDJ+24]. Let $\mathbf{z}^T_{q/2,\mathbf{x}_L} \in \{0,1\}^\ell$ be the highest-order bits of all entries of $\mathbf{z}^T_{\mathbf{x}_L}$. Since K_H does not appear in the adversary's view, we can leverage the programmability of the PRO. Therefore, by setting the output of $\mathsf{O}(\mathsf{hEval}, h, \mathbf{x}_L)$ to $C(\mathbf{x}_L) - \mathbf{z}^T_{q/2,\mathbf{x}_L}$, we can simulate both hybrids. Thus, we can conclude that Hyb_{L+7} is statistically indistinguishable Hyb_{L+8}.

As Hyb_{L+8} is identical to $\mathsf{Hyb}_{C,\mathsf{sim}}$, this completes the proof. □

Theorem 2. Suppose the LWE assumption in Definition 1 and the all-product LWE assumption in Definition 3, the construction in Subsection 4.2 satisfies indistinguishability (Definition 5).

Proof. **Proof sketch.** To prove that the obfuscated circuit can be simulated wihtout the circuit C that is randomly chosen either from C_0 or C_1, we apply Lemma 3 to $\mathcal{D}_{C,\mathsf{real}}$ in Lemma 6 $(3L+1)$ times for the input size L. Specifically, we first prove that the pseudorandomness of $\mathcal{D}_{C,\mathsf{real}}$ shown in Lemma 6 is preserved when $\{\mathbf{v}^T_{F,\mathbf{x}_L}\}_{\mathbf{x}_L \in \{0,1\}^L}$ are replaced with a preimage $\mathbf{K}_F$. Then for every $h \in \{L, \ldots, 1\}$, we show that the pseudorandomness of $\{\mathbf{p}^T_{\mathbf{x}_h}\}_{\mathbf{x}_i \in \{0,1\}^h}$ with presence of preimages

$$\mathbf{K}_F, \{\mathbf{M}_{i,b}, \mathbf{N}_{i,b}\}_{h<i\le L, b\in\{0,1\}}, \{\mathbf{K}_{i,b}\}_{h<i\le L, b\in\{0,1\}}$$

implies that $\{\mathbf{q}_{\mathbf{x}_h}\}_{\mathbf{x}_h\in\{0,1\}^h}$, $\{\mathbf{p}_{\mathbf{x}_h}\}_{\mathbf{x}_h\in\{0,1\}^h}$, and $\{\mathbf{v}_{\mathbf{x}_h}\}_{\mathbf{x}_h\in\{0,1\}^h}$ can be replaced with $\{\mathbf{M}_{h,b}\}_{b\in\{0,1\}}$, $\{\mathbf{N}_{h,b}\}_{b\in\{0,1\}}$, and $\{\mathbf{K}_{h,b}\}_{b\in\{0,1\}}$, respectively. By repeating this process until $h=1$, we can conclude that the obfuscated circuit does not depend on C except for its output $C(\mathbf{x}_L) = C_0(\mathbf{x}_L) = C_1(\mathbf{x}_L)$.

Pseudorandom after adding $\mathbf{K}_F$. Let Samp_F be a PPT algorithm that takes as input 1^λ and

outputs

$$\mathbf{S}_F := \begin{pmatrix} \hat{\mathbf{s}}_{\mathbf{0}_L}^T, & \hat{\mathbf{s}}_{\mathbf{0}_L}^T \\ \vdots & \vdots \\ \hat{\mathbf{s}}_{\mathbf{1}_L}^T, & \hat{\mathbf{s}}_{\mathbf{1}_L}^T \end{pmatrix} \in \mathbb{Z}_q^{2^L \times 2(n+1)}, \mathbf{P}_F := \begin{pmatrix} \mathbf{A}_F \\ \mathbf{0}_{(n+1)\times \ell} \end{pmatrix} \in \mathbb{Z}_q^{2(n+1)\times \ell},$$

$$\mathsf{aux}_{F,1} := \left(\begin{array}{l} \mathbf{A}_{\mathsf{fhe}} := \begin{pmatrix} \bar{\mathbf{A}}_{\mathsf{fhe}} \\ \bar{\mathbf{t}}^T \bar{\mathbf{A}}_{\mathsf{fhe}} + \mathbf{e}_{\mathsf{fhe}}^T \end{pmatrix}, \mathbf{X} := \mathbf{A}_{\mathsf{fhe}} \mathbf{R}_{\mathsf{fhe}} - (C, K_B, K_H) \otimes \mathbf{G}_{n+1}, \\ \mathbf{c}_{\mathsf{att},\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T (\mathbf{A}_{\mathsf{att},0} - (1, \mathsf{bits}(\mathbf{X}), \mathbf{0}_L) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{\mathsf{att},\epsilon}}^T, \\ \mathbf{c}_{t,\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T (\mathbf{A}_{t,0} - \mathbf{t}^T \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{t,\epsilon}}^T, \\ \mathbf{p}_\epsilon^T := (\hat{\mathbf{s}}_\epsilon^T, \hat{\mathbf{s}}_\epsilon^T) \mathbf{B}_{0,\star} + \mathbf{e}_{\mathbf{p}_\epsilon}^T, \{\mathbf{q}_{\mathbf{x}_i}^T := (\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T) \mathbf{B}_{i,x_i} + \mathbf{e}_{q_{\mathbf{x}_i}}^T\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}, \\ \{\mathbf{p}_{\mathbf{x}_i}^T := (\hat{\mathbf{s}}_{\mathbf{x}_i}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T) \mathbf{B}_{i,\star} + \mathbf{e}_{p_{\mathbf{x}_i}}^T\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}, \\ \left\{ \begin{array}{l} \mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T := -\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T \mathbf{A}_{\mathsf{att},i-1} \mathbf{T}_{\mathsf{att},i,x_i} \\ \qquad + \hat{\mathbf{s}}_{\mathbf{x}_i}^T (\mathbf{A}_{\mathsf{att},i} - (\mathbf{0}_{L_{\mathsf{fhe}}+i}, x_i, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{v_{\mathsf{att},\mathbf{x}_i}}^T \end{array} \right\}_{\substack{i\in[L], \\ \mathbf{x}_i\in\{0,1\}^i}}, \\ \{\mathbf{v}_{t,\mathbf{x}_i}^T := -\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T \mathbf{A}_{t,i-1} \mathbf{T}_{t,i,x_i} + \hat{\mathbf{s}}_{\mathbf{x}_i}^T \mathbf{A}_{t,i} + \mathbf{e}_{v_{t,\mathbf{x}_i}}^T\}_{\substack{i\in[L], \\ \mathbf{x}_i\in\{0,1\}^i}} \end{array} \right),$$

$$\mathsf{aux}_{F,2} := \left(\{\mathbf{R}_b\}_{b\in\{0,1\}}, h, \{\mathbf{A}_{\mathsf{att},i}\}_{i\in[0,L]}, \{\mathbf{A}_{t,t}\}_{i\in[0,L]}, \{C(\mathbf{x}_L)\}_{\mathbf{x}_L\in\{0,1\}^L} \right),$$

where the private-coin of Samp_F is

$$\begin{pmatrix} \bar{\mathbf{s}}, \bar{\mathbf{t}}, \mathbf{e}_{c_{\mathsf{att},\epsilon}}^T, \mathbf{e}_{c_{t,\epsilon}}^T, \mathbf{e}_{p_\epsilon}^T, \{\mathbf{e}_{p_{\mathbf{x}_i}}, \mathbf{e}_{q_{\mathbf{x}_i}}, \mathbf{e}_{v_{\mathsf{att},\mathbf{x}_i}}, \mathbf{e}_{v_{t,\mathbf{x}_i}}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}, \\ \{\mathbf{B}_{i,b}\}_{i\in[L],b\in\{0,1,\star\}} / \{\mathbf{B}_{L,\star}\} \end{pmatrix}.$$

We can easily confirm that $\mathsf{aux}_{F,2}$ allows constructing $\mathbf{P}_F$, ensuring the existence of $\mathsf{Reconstruct}$ algorithm.

To apply Lemma 3 with Samp_{K_F}, it suffices to prove that

$$\left(\mathbf{B}_{L,\star}, \mathbf{P}_F, \mathbf{S}_F \mathbf{B}_{L,\star} + \mathbf{E}_F, \mathbf{S}_F \mathbf{P}_F + \mathbf{E}'_F, \mathsf{aux}_{F,1}, \mathsf{aux}_{F,2}\right) \overset{\mathrm{c}}{\approx} \left(\mathbf{B}_{L,\star}, \mathbf{P}_F, \mathbf{C}_F, \mathbf{C}'_F, \widetilde{\mathsf{aux}_{F,1}}, \mathsf{aux}_{F,2}\right), \quad (6)$$

where $\mathbf{B}_{L,\star} \leftarrow\!\$\ \mathbb{Z}_q^{2(n+1)\times m_B}$, $\mathbf{E}_F \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi}^{2^L\times\ell}$, $\mathbf{E}'_F \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi_F}^{2^L\times\ell}$, $\mathbf{C}_F \leftarrow\!\$\ \mathbb{Z}_q^{2^L\times m_B}$, $\mathbf{C}'_F \leftarrow\!\$\ \mathbb{Z}_q^{2^L\times\ell}$ and

$$\widetilde{\mathsf{aux}_{F,1}} := \left(\begin{array}{l} \mathbf{A}_{\mathsf{fhe}} \leftarrow\!\$\ \mathbb{Z}_q^{(n+1)\times m}, \mathbf{X} \leftarrow\!\$\ \mathbb{Z}_q^{(n+1)\times(|C|+2\lambda)m}, \\ \mathbf{c}_{\mathsf{att},\epsilon}^T \leftarrow\!\$\ \mathbb{Z}_q^{1\times n'm}, \mathbf{c}_{t,\epsilon}^T \leftarrow\!\$\ \mathbb{Z}_q^{1\times(n+1)m}, \mathbf{p}_\epsilon^T \leftarrow\!\$\ \mathbb{Z}_q^{1\times m_B}, \\ \{\mathbf{q}_{\mathbf{x}_i}^T \leftarrow\!\$\ \mathbb{Z}_q^{1\times m_B}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}, \{\mathbf{p}_{\mathbf{x}_i}^T \leftarrow\!\$\ \mathbb{Z}_q^{1\times m_B}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}, \\ \{\mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T \leftarrow\!\$\ \mathbb{Z}_q^{1\times L'm}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i}, \{\mathbf{v}_{t,\mathbf{x}_i}^T \leftarrow\!\$\ \mathbb{Z}_q^{1\times(n+1)m}\}_{i\in[L],\mathbf{x}_i\in\{0,1\}^i} \end{array} \right).$$

Since the L.H.S and R.H.S distributions in Relation 6 are identitcal to $\mathcal{D}_{C,\mathsf{real}}$ and $\mathcal{D}_{C,\mathsf{sim}}$, respectively, Relation 6 follows immidiately from Lemma 6. Therefore, Lemma 3 guarantees the following indistinguishability:

$$\left(\mathbf{B}_{L,\star}, \mathbf{P}_F, \mathbf{S}_F \mathbf{B}_{L,\star} + \mathbf{E}_F, \mathbf{K}_F, \mathsf{aux}_{F,1}, \mathsf{aux}_{F,2}\right) \overset{\mathrm{c}}{\approx} \left(\mathbf{B}_{L,\star}, \mathbf{P}_F, \mathbf{C}_F, \mathbf{K}_F, \widetilde{\mathsf{aux}_{F,1}}, \mathsf{aux}_{F,2}\right). \quad (7)$$

Pseudorandom after adding $\mathbf{M}_{h,b}$, $\mathbf{N}_{h,b}$, and $\mathbf{K}_{h,b}$. For every $h \in \{L, \ldots, 1\}$, we define the

following distributions:

$$\mathbf{S}_{h,1} := \begin{pmatrix} \hat{\mathbf{s}}_{\mathbf{0}_{h-1}}^T, & \hat{\mathbf{s}}_{\mathbf{0}_{h-1}1}^T \\ \vdots & \vdots \\ \hat{\mathbf{s}}_{\mathbf{1}_{h-1}}^T, & \hat{\mathbf{s}}_{\mathbf{1}_{h-1}1}^T \end{pmatrix} \in \mathbb{Z}_q^{2^{h-1}\times 2(n+1)},$$

$$\mathbf{P}_{h,1} := \begin{pmatrix} \mathbf{U}_\star \mathbf{B}_{h,\star}, & -(\mathbf{A}_{\mathsf{att},h-1}\mathbf{T}_{\mathsf{att},h,1}, \mathbf{A}_{t,h-1}\mathbf{T}_{t,h,1}) \\ & (\mathbf{A}_{\mathsf{att},h} - (\mathbf{0}_{L_{\mathsf{fhe}}+h}^T, 1, \mathbf{0}_{L-h}^T) \otimes \mathbf{G}_{n+1}, \mathbf{A}_{t,h}) \end{pmatrix} \in \mathbb{Z}_q^{2(n+1)\times(m_B+(1+n+L')m)},$$

$$\mathsf{aux}_{h,1,1} := \left(\begin{array}{l} \mathbf{A}_{\mathsf{fhe}} := \begin{pmatrix} \bar{\mathbf{A}}_{\mathsf{fhe}} \\ \bar{\mathbf{t}}^T \bar{\mathbf{A}}_{\mathsf{fhe}} + \mathbf{e}_{\mathsf{fhe}}^T \end{pmatrix}, \mathbf{X} := \mathbf{A}_{\mathsf{fhe}}\mathbf{R}_{\mathsf{fhe}} - (C, K_B, K_H) \otimes \mathbf{G}_{n+1}, \\ \mathbf{c}_{\mathsf{att},\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T(\mathbf{A}_{\mathsf{att},0} - (1, \mathsf{bits}(\mathbf{X}), \mathbf{0}_L) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{\mathsf{att},\epsilon}}^T, \\ \mathbf{c}_{t,\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T(\mathbf{A}_{t,0} - \mathbf{t}^T \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{t,\epsilon}}^T, \mathbf{p}_\epsilon^T := (\hat{\mathbf{s}}_\epsilon^T, \hat{\mathbf{s}}_\epsilon^T)\mathbf{B}_{0,\star} + \mathbf{e}_{\mathbf{p}_\epsilon}^T, \\ \{\mathbf{q}_{\mathbf{x}_i}^T := (\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T)\mathbf{B}_{i,x_i} + \mathbf{e}_{q_{\mathbf{x}_i}}^T\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \\ \{\mathbf{q}_{\mathbf{x}_{h-1}0}^T := (\hat{\mathbf{s}}_{\mathbf{x}_{h-1}}^T, \hat{\mathbf{s}}_{\mathbf{x}_{h-1}0}^T)\mathbf{B}_{h,0} + \mathbf{e}_{q_{\mathbf{x}_{h-1}0}}^T\}_{\mathbf{x}_{h-1}\in\{0,1\}^{h-1}}, \\ \{\mathbf{p}_{\mathbf{x}_i}^T := (\hat{\mathbf{s}}_{\mathbf{x}_i}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T)\mathbf{B}_{i,\star} + \mathbf{e}_{p_{\mathbf{x}_i}}^T\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \\ \{\mathbf{p}_{\mathbf{x}_{h-1}0}^T := (\hat{\mathbf{s}}_{\mathbf{x}_{h-1}}^T, \hat{\mathbf{s}}_{\mathbf{x}_{h-1}0}^T)\mathbf{B}_{h,\star} + \mathbf{e}_{p_{\mathbf{x}_{h-1}0}}^T\}_{\mathbf{x}_{h-1}\in\{0,1\}^{h-1}}, \\ \left\{ \begin{array}{l} \mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T := -\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T \mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,x_i} \\ \qquad + \hat{\mathbf{s}}_{\mathbf{x}_i}^T(\mathbf{A}_{\mathsf{att},i} - (\mathbf{0}_{L_{\mathsf{fhe}}+i}, x_i, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{v_{\mathsf{att},\mathbf{x}_i}}^T \end{array} \right\}_{\substack{i\in[h-1],\\ \mathbf{x}_i\in\{0,1\}^i}}, \\ \left\{ \begin{array}{l} \mathbf{v}_{\mathsf{att},\mathbf{x}_{h-1}0}^T := -\hat{\mathbf{s}}_{\mathbf{x}_{h-1}}^T \mathbf{A}_{\mathsf{att},h-1}\mathbf{T}_{\mathsf{att},h,0} \\ \qquad + \hat{\mathbf{s}}_{\mathbf{x}_{h-1}0}^T(\mathbf{A}_{\mathsf{att},h} - (\mathbf{0}_{L_{\mathsf{fhe}}+h}^T, 0, \mathbf{0}_{L-h}^T) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{v_{\mathsf{att},\mathbf{x}_{h-1}0}}^T \end{array} \right\}_{xVech-1\in\{0,1\}^{h-1}}, \\ \{\mathbf{v}_{t,\mathbf{x}_i}^T := -\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T \mathbf{A}_{t,i-1}\mathbf{T}_{t,i,x_i} + \hat{\mathbf{s}}_{\mathbf{x}_i}^T \mathbf{A}_{t,i} + \mathbf{e}_{v_{t,\mathbf{x}_i}}^T\}_{\substack{i\in[h-1],\\ \mathbf{x}_i\in\{0,1\}^i}}, \\ \{\mathbf{v}_{t,\mathbf{x}_{h-1}0}^T := -\hat{\mathbf{s}}_{\mathbf{x}_{h-1}}^T \mathbf{A}_{t,h-1}\mathbf{T}_{t,h,0} + \hat{\mathbf{s}}_{\mathbf{x}_{h-1}0}^T \mathbf{A}_{t,h} + \mathbf{e}_{v_{t,\mathbf{x}_{h-1}0}}^T\}_{\mathbf{x}_{h-1}\in\{0,1\}^{h-1}} \end{array} \right)$$

$$\mathsf{aux}_{h,1,2} := \begin{pmatrix} \{\mathbf{R}_b\}_{b\in\{0,1\}}, h, \{\mathbf{A}_{\mathsf{att},i}\}_{i\in[0,L]}, \{\mathbf{A}_{t,t}\}_{i\in[0,L]}, \{C(\mathbf{x}_L)\}_{\mathbf{x}_L\in\{0,1\}^L}, \\ \mathbf{B}_{h,\star}, \{\mathbf{B}_{i,b}\}_{i\in[h+1,L],b\in\{0,1,\star\}}, \{\mathbf{M}_{i,b}\}_{i\in[h+1,L],b\in\{0,1\}}, \\ \{\mathbf{N}_{i,b}\}_{i\in[h+1,L],b\in\{0,1\}}, \{\mathbf{K}_{i,b}\}_{i\in[h+1,L],b\in\{0,1\}}, \mathbf{K}_F \end{pmatrix},$$

$$\widetilde{\mathsf{aux}_{h,1,1}} := \left(\begin{array}{l} \mathbf{A}_{\mathsf{fhe}} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times m}, \mathbf{X} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times(|C|+2\lambda)m}, \\ \mathbf{c}_{\mathsf{att},\epsilon}^T \leftarrow\$ \mathbb{Z}_q^{1\times L'm}, \mathbf{c}_{t,\epsilon}^T \leftarrow\$ \mathbb{Z}_q^{1\times(n+1)m}, \mathbf{p}_\epsilon^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}, \\ \{\mathbf{q}_{\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \{\mathbf{q}_{\mathbf{x}_{h-1}0}^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}\}_{\mathbf{x}_{h-1}\in\{0,1\}^{h-1}}, \\ \{\mathbf{p}_{\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \{\mathbf{p}_{\mathbf{x}_{h-1}0}^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}\}_{\mathbf{x}_{h-1}\in\{0,1\}^{h-1}}, \\ \{\mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times L'm}\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \{\mathbf{v}_{\mathsf{att},\mathbf{x}_{h-1}0}^T \leftarrow\$ \mathbb{Z}_q^{1\times L'm}\}_{\mathbf{x}_{h-1}\in\{0,1\}^{h-1}}, \\ \{\mathbf{v}_{t,\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times(n+1)m}\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \{\mathbf{v}_{t,\mathbf{x}_{h-1}0}^T \leftarrow\$ \mathbb{Z}_q^{1\times(n+1)m}\}_{\mathbf{x}_{h-1}\in\{0,1\}^{h-1}} \end{array} \right),$$

$$\mathbf{S}_{h,0} := \begin{pmatrix} \hat{\mathbf{s}}_{\mathbf{0}_{h-1}}^T, & \hat{\mathbf{s}}_{\mathbf{0}_{h-1}0}^T \\ \vdots & \vdots \\ \hat{\mathbf{s}}_{\mathbf{1}_{h-1}}^T, & \hat{\mathbf{s}}_{\mathbf{1}_{h-1}0}^T \end{pmatrix} \in \mathbb{Z}_q^{2^{h-1}\times 2(n+1)},$$

$$\mathbf{P}_{h,0} := \begin{pmatrix} \mathbf{U}_\star \mathbf{B}_{h,\star}, & -(\mathbf{A}_{\mathsf{att},h-1}\mathbf{T}_{\mathsf{att},h,0}, \mathbf{A}_{t,h-1}\mathbf{T}_{t,h,0}) \\ & (\mathbf{A}_{\mathsf{att},h} - (\mathbf{0}_{L_{\mathsf{fhe}}+h}^T, 0, \mathbf{0}_{L-h}^T) \otimes \mathbf{G}_{n+1}, \mathbf{A}_{t,h}) \end{pmatrix} \in \mathbb{Z}_q^{2(n+1)\times(m_B+(1+n+L')m)},$$

$$\mathsf{aux}_{h,0,1} := \left(\begin{array}{l} \mathbf{A}_{\mathsf{fhe}} := \begin{pmatrix} \bar{\mathbf{A}}_{\mathsf{fhe}} \\ \bar{\mathbf{t}}^T \bar{\mathbf{A}}_{\mathsf{fhe}} + \mathbf{e}_{\mathsf{fhe}}^T \end{pmatrix}, \mathbf{X} := \mathbf{A}_{\mathsf{fhe}}\mathbf{R}_{\mathsf{fhe}} - (C, K_B, K_H) \otimes \mathbf{G}_{n+1}, \\ \mathbf{c}_{\mathsf{att},\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T(\mathbf{A}_{\mathsf{att},0} - (1, \mathsf{bits}(\mathbf{X}), \mathbf{0}_L) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{\mathsf{att},\epsilon}}^T, \\ \mathbf{c}_{t,\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T(\mathbf{A}_{t,0} - \mathbf{t}^T \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{t,\epsilon}}^T, \mathbf{p}_\epsilon^T := (\hat{\mathbf{s}}_\epsilon^T, \hat{\mathbf{s}}_\epsilon^T)\mathbf{B}_{0,\star} + \mathbf{e}_{\mathbf{p}_\epsilon}^T, \\ \{\mathbf{q}_{\mathbf{x}_i}^T := (\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T)\mathbf{B}_{i,x_i} + \mathbf{e}_{q_{\mathbf{x}_i}}^T\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \\ \{\mathbf{q}_{\mathbf{x}_{h-1}1}^T := (\hat{\mathbf{s}}_{\mathbf{x}_{h-1}}^T, \hat{\mathbf{s}}_{\mathbf{x}_{h-1}1}^T)\mathbf{B}_{h,1} + \mathbf{e}_{q_{\mathbf{x}_{h-1}1}}^T\}_{\mathbf{x}_{h-1}\in\{0,1\}^{h-1}}, \\ \{\mathbf{p}_{\mathbf{x}_i}^T := (\hat{\mathbf{s}}_{\mathbf{x}_i}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T)\mathbf{B}_{i,\star} + \mathbf{e}_{p_{\mathbf{x}_i}}^T\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \\ \left\{ \begin{array}{rl} \mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T := & -\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T \mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,x_i} \\ & + \hat{\mathbf{s}}_{\mathbf{x}_i}^T(\mathbf{A}_{\mathsf{att},i} - (\mathbf{0}_{L_{\mathsf{fhe}}+i}, x_i, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{v_{\mathsf{att},\mathbf{x}_i}}^T \end{array} \right\}_{\substack{i\in[h-1],\\ \mathbf{x}_i\in\{0,1\}^i}}, \\ \{\mathbf{v}_{t,\mathbf{x}_i}^T := -\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T \mathbf{A}_{t,i-1}\mathbf{T}_{t,i,x_i} + \hat{\mathbf{s}}_{\mathbf{x}_i}^T \mathbf{A}_{t,i} + \mathbf{e}_{v_{t,\mathbf{x}_i}}^T\}_{\substack{i\in[h-1],\\ \mathbf{x}_i\in\{0,1\}^i}} \end{array} \right),$$

$$\mathsf{aux}_{h,0,2} := \begin{pmatrix} \{\mathbf{R}_b\}_{b\in\{0,1\}}, h, \{\mathbf{A}_{\mathsf{att},i}\}_{i\in[0,L]}, \{\mathbf{A}_{t,t}\}_{i\in[0,L]}, \{C(\mathbf{x}_L)\}_{\mathbf{x}_L\in\{0,1\}^L}, \\ \mathbf{B}_{h,\star}, \mathbf{B}_{h,1}, \mathbf{N}_{h,1}, \mathbf{K}_{h,1}, \{\mathbf{B}_{i,b}\}_{i\in[h+1,L],b\in\{0,1,\star\}}, \{\mathbf{M}_{i,b}\}_{i\in[h+1,L],b\in\{0,1\}}, \\ \{\mathbf{N}_{i,b}\}_{i\in[h+1,L],b\in\{0,1\}}, \{\mathbf{K}_{i,b}\}_{i\in[h+1,L],b\in\{0,1\}}, \mathbf{K}_F \end{pmatrix},$$

$$\widetilde{\mathsf{aux}_{h,0,1}} := \begin{pmatrix} \mathbf{A}_{\mathsf{fhe}} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times m}, \mathbf{X} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times(|C|+2\lambda)m}, \\ \mathbf{c}_{\mathsf{att},\epsilon}^T \leftarrow\$ \mathbb{Z}_q^{1\times L'm}, \mathbf{c}_{t,\epsilon}^T \leftarrow\$ \mathbb{Z}_q^{1\times(n+1)m}, \mathbf{p}_\epsilon^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}, \\ \{\mathbf{q}_{\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \{\mathbf{p}_{\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \\ \{\mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times L'm}\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \{\mathbf{v}_{t,\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times(n+1)m}\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i} \end{pmatrix}.$$

Similarly, for every $h \in \{L-1,\dots,1\}$ and $h=0$, we define the following distributions:

$$\mathbf{S}_{h,\star} := \begin{pmatrix} \hat{\mathbf{s}}_{\mathbf{0}_h}^T, & \hat{\mathbf{s}}_{\mathbf{0}_h}^T \\ \vdots & \vdots \\ \hat{\mathbf{s}}_{\mathbf{1}_h}^T, & \hat{\mathbf{s}}_{\mathbf{1}_h 1}^T \end{pmatrix} \in \mathbb{Z}_q^{2^h \times 2(n+1)},$$

$$\mathbf{P}_{h,\star} := \begin{pmatrix} \mathbf{U}_0\mathbf{B}_{h+1,0}, & \mathbf{U}_1\mathbf{B}_{h+1,1} \end{pmatrix} \in \mathbb{Z}_q^{2(n+1)\times 2m_B},$$

$$\mathsf{aux}_{h,\star,1} := \begin{pmatrix} \mathbf{A}_{\mathsf{fhe}} := \begin{pmatrix} \bar{\mathbf{A}}_{\mathsf{fhe}} \\ \bar{\mathbf{t}}^T\bar{\mathbf{A}}_{\mathsf{fhe}} + \mathbf{e}_{\mathsf{fhe}}^T \end{pmatrix}, \mathbf{X} := \mathbf{A}_{\mathsf{fhe}}\mathbf{R}_{\mathsf{fhe}} - (C, K_B, K_H) \otimes \mathbf{G}_{n+1}, \\ \mathbf{c}_{\mathsf{att},\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T(\mathbf{A}_{\mathsf{att},0} - (1, \mathsf{bits}(\mathbf{X}), \mathbf{0}_L) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{\mathsf{att},\epsilon}}^T, \\ \mathbf{c}_{t,\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T(\mathbf{A}_{t,0} - \mathbf{t}^T \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{t,\epsilon}}^T, \mathbf{p}_\epsilon^T := (\hat{\mathbf{s}}_\epsilon^T, \hat{\mathbf{s}}_\epsilon^T)\mathbf{B}_{0,\star} + \mathbf{e}_{\mathbf{p}_\epsilon}^T, \\ \{\mathbf{q}_{\mathbf{x}_i}^T := (\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T)\mathbf{B}_{i,x_i} + \mathbf{e}_{q_{\mathbf{x}_i}}^T\}_{i\in[h],\mathbf{x}_i\in\{0,1\}^i}, \\ \{\mathbf{p}_{\mathbf{x}_i}^T := (\hat{\mathbf{s}}_{\mathbf{x}_i}^T, \hat{\mathbf{s}}_{\mathbf{x}_i}^T)\mathbf{B}_{i,\star} + \mathbf{e}_{p_{\mathbf{x}_i}}^T\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \\ \left\{\begin{aligned} \mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T := &-\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T\mathbf{A}_{\mathsf{att},i-1}\mathbf{T}_{\mathsf{att},i,x_i} \\ &+ \hat{\mathbf{s}}_{\mathbf{x}_i}^T(\mathbf{A}_{\mathsf{att},i} - (\mathbf{0}_{L_{\mathsf{fhe}}+i}, x_i, \mathbf{0}_{L-i}) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{v_{\mathsf{att},\mathbf{x}_i}}^T \end{aligned}\right\}_{\substack{i\in[h],\\ \mathbf{x}_i\in\{0,1\}^i}}, \\ \{\mathbf{v}_{t,\mathbf{x}_i}^T := -\hat{\mathbf{s}}_{\mathbf{x}_{i-1}}^T\mathbf{A}_{t,i-1}\mathbf{T}_{t,i,x_i} + \hat{\mathbf{s}}_{\mathbf{x}_i}^T\mathbf{A}_{t,i} + \mathbf{e}_{v_{t,\mathbf{x}_i}}^T\}_{\substack{i\in[h],\\ \mathbf{x}_i\in\{0,1\}^i}} \end{pmatrix},$$

$$\mathsf{aux}_{h,\star,2} := \begin{pmatrix} \{\mathbf{R}_b\}_{b\in\{0,1\}}, h, \{\mathbf{A}_{\mathsf{att},i}\}_{i\in[0,L]}, \{\mathbf{A}_{t,t}\}_{i\in[0,L]}, \{C(\mathbf{x}_L)\}_{\mathbf{x}_L\in\{0,1\}^L}, \\ \mathbf{B}_{h,1}, \mathbf{N}_{h,1}, \mathbf{K}_{h,1}, \{\mathbf{B}_{i,b}\}_{i\in[h+1,L],b\in\{0,1,\star\}}, \{\mathbf{M}_{i,b}\}_{i\in[h+2,L],b\in\{0,1\}}, \\ \{\mathbf{N}_{i,b}\}_{i\in[h+1,L],b\in\{0,1\}}, \{\mathbf{K}_{i,b}\}_{i\in[h+1L],b\in\{0,1\}}, \mathbf{K}_F \end{pmatrix},$$

$$\widetilde{\mathsf{aux}_{h,\star,1}} := \begin{pmatrix} \mathbf{A}_{\mathsf{fhe}} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times m}, \mathbf{X} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times(|C|+2\lambda)m}, \\ \mathbf{c}_{\mathsf{att},\epsilon}^T \leftarrow\$ \mathbb{Z}_q^{1\times L'm}, \mathbf{c}_{t,\epsilon}^T \leftarrow\$ \mathbb{Z}_q^{1\times(n+1)m}, \mathbf{p}_\epsilon^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}, \\ \{\mathbf{q}_{\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}\}_{i\in[h],\mathbf{x}_i\in\{0,1\}^i}, \{\mathbf{p}_{\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \\ \{\mathbf{v}_{\mathsf{att},\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times L'm}\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \{\mathbf{v}_{t,\mathbf{x}_i}^T \leftarrow\$ \mathbb{Z}_q^{1\times(n+1)m}\}_{i\in[h-1],\mathbf{x}_i\in\{0,1\}^i}, \end{pmatrix},$$

$$\mathbf{S}_{0,\star} := \begin{pmatrix} \hat{\mathbf{s}}_\epsilon^T, & \hat{\mathbf{s}}_\epsilon^T \end{pmatrix} \in \mathbb{Z}_q^{2\times 2(n+1)},$$

$$\mathbf{P}_{0,\star} := \begin{pmatrix} \mathbf{U}_0\mathbf{B}_{0,0}, & \mathbf{U}_1\mathbf{B}_{0,1} \end{pmatrix} \in \mathbb{Z}_q^{2(n+1)\times 2m_B},$$

$$\mathsf{aux}_{0,\star,1} := \begin{pmatrix} \mathbf{A}_{\mathsf{fhe}} := \begin{pmatrix} \bar{\mathbf{A}}_{\mathsf{fhe}} \\ \bar{\mathbf{t}}^T\bar{\mathbf{A}}_{\mathsf{fhe}} + \mathbf{e}_{\mathsf{fhe}}^T \end{pmatrix}, \mathbf{X} := \mathbf{A}_{\mathsf{fhe}}\mathbf{R}_{\mathsf{fhe}} - (C, K_B, K_H) \otimes \mathbf{G}_{n+1}, \\ \mathbf{c}_{\mathsf{att},\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T(\mathbf{A}_{\mathsf{att},0} - (1, \mathsf{bits}(\mathbf{X}), \mathbf{0}_L) \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{\mathsf{att},\epsilon}}^T, \\ \mathbf{c}_{t,\epsilon}^T := \hat{\mathbf{s}}_\epsilon^T(\mathbf{A}_{t,0} - \mathbf{t}^T \otimes \mathbf{G}_{n+1}) + \mathbf{e}_{c_{t,\epsilon}}^T \end{pmatrix},$$

$$\mathsf{aux}_{0,\star,2} := \begin{pmatrix} \{\mathbf{R}_b\}_{b\in\{0,1\}}, h, \{\mathbf{A}_{\mathsf{att},i}\}_{i\in[0,L]}, \{\mathbf{A}_{t,t}\}_{i\in[0,L]}, \{C(\mathbf{x}_L)\}_{\mathbf{x}_L\in\{0,1\}^L}, \\ \{\mathbf{B}_{i,b}\}_{i\in[L],b\in\{0,1,\star\}}, \{\mathbf{M}_{i,b}\}_{i\in[2,L],b\in\{0,1\}}, \\ \{\mathbf{N}_{i,b}\}_{i\in[L],b\in\{0,1\}}, \{\mathbf{K}_{i,b}\}_{i\in[L],b\in\{0,1\}}, \mathbf{K}_F \end{pmatrix},$$

$$\widetilde{\mathsf{aux}_{0,\star,1}} := \begin{pmatrix} \mathbf{A}_{\mathsf{fhe}} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times m}, \mathbf{X} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times(|C|+2\lambda)m}, \\ \mathbf{c}_{\mathsf{att},\epsilon}^T \leftarrow\$ \mathbb{Z}_q^{1\times L'm}, \mathbf{c}_{t,\epsilon}^T \leftarrow\$ \mathbb{Z}_q^{1\times(n+1)m}, \mathbf{p}_\epsilon^T \leftarrow\$ \mathbb{Z}_q^{1\times m_B}, \end{pmatrix}.$$

We want to prove that for all $h \in \{L-1,\dots,0\}$, it holds that

$$\begin{aligned} &\left(\mathbf{B}_{h,\star}, \mathbf{P}_{h,\star}, \mathbf{S}_{h,\star}\mathbf{B}_{h,\star} + \mathbf{E}_{h,\star}, (\mathbf{M}_{h+1,0}, \mathbf{M}_{h+1,1}), \mathsf{aux}_{h,\star,1}, \mathsf{aux}_{h,\star,2}\right) \\ &\overset{c}{\approx} \left(\mathbf{B}_{h,\star}, \mathbf{P}_{h,\star}, \mathbf{C}_{h,\star}, (\mathbf{M}_{h+1,0}, \mathbf{M}_{h+1,1}), \widetilde{\mathsf{aux}_{h,\star,1}}, \mathsf{aux}_{h,\star,2}\right), \end{aligned} \tag{8}$$

where $\mathbf{B}_{h,\star} \leftarrow\!\$\ \mathbb{Z}_q^{2(n+1)\times m_B}$, $\mathbf{E}_{h,\star} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi}^{2^h\times m_B}$, and $\mathbf{C}_{h,\star} \leftarrow\!\$\ \mathbb{Z}_q^{2^h\times m_B}$.

Base case. For $h = L-1$, we first claim the follwoing indistinguishability:

$$\begin{aligned}&\left(\mathbf{B}_{L,1}, \mathbf{P}_{L,1}, \mathbf{S}_{L,1}\mathbf{B}_{L,1} + \mathbf{E}_{L,1}, \mathbf{S}_{L,1}\mathbf{P}_{L,1} + \mathbf{E}'_{L,1}, \mathsf{aux}_{L,1,1}, \mathsf{aux}_{L,1,2}\right)\\ &\overset{\mathrm{c}}{\approx} \left(\mathbf{B}_{L,1}, \mathbf{P}_{L,1}, \mathbf{C}_{L,1}, \mathbf{C}'_{L,1}, \widetilde{\mathsf{aux}_{L,1,1}}, \mathsf{aux}_{L,1,2}\right),\end{aligned} \tag{9}$$

where $\mathbf{B}_{L,1} \leftarrow\!\$\ \mathbb{Z}_q^{2(n+1)\times m_B}$, $\mathbf{E}_{L,1} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi}^{2^{L-1}\times m_B}$, $\mathbf{E}'_{L,1} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi_i}^{2^{L-1}\times(m_B+(1+n+L')m)}$, $\mathbf{C}_{L,1} \leftarrow\!\$\ \mathbb{Z}_q^{2^{L-1}\times m_B}$, and $\mathbf{C}'_{L,1} \leftarrow\!\$\ \mathbb{Z}_q^{2^{L-1}\times(m_B+(1+n+L')m)}$. The indistinguishability in Relation 9 follows immidiately from that in Relation 7. Indeed, the distributions in these relations are identical because $\mathbf{B}_{L,\star}$ and $\mathbf{K}_F$ are included in $\mathsf{aux}_{L,1,2}$, and $\mathbf{S}_F\mathbf{B}_{L,\star}$ and $\left(\mathbf{v}_{\mathsf{att},\mathbf{x}_{L-1}1}^T, \mathbf{v}_{t,\mathbf{x}_{L-1}1}^T\right)_{\mathbf{x}_{L-1}\in\{0,1\}^{L-1}}$ in $\mathsf{aux}_{F,1}$ are identical to the first m_B columns and the last $(1+n+L')m$ columns of $\mathbf{S}_{L,1}\mathbf{P}_{L,1}$, respectively. Additionally, $\mathbf{B}_{L,1}^{-1}(\mathbf{P}_{L,1}) = (\mathbf{N}_{L,1}, \mathbf{K}_{L,1})$ holds. Thus, by Lemma 3, the follwoing indistinguishability also holds:

$$\begin{aligned}&\left(\mathbf{B}_{L,1}, \mathbf{P}_{L,1}, \mathbf{S}_{L,1}\mathbf{B}_{L,1} + \mathbf{E}_{L,1}, (\mathbf{N}_{L,1}, \mathbf{K}_{L,1}), \mathsf{aux}_{L,1,1}, \mathsf{aux}_{L,1,2}\right)\\ &\overset{\mathrm{c}}{\approx} \left(\mathbf{B}_{L,1}, \mathbf{P}_{L,1}, \mathbf{C}_{L,1}, (\mathbf{N}_{L,1}, \mathbf{K}_{L,1}), \widetilde{\mathsf{aux}_{L,1,1}}, \mathsf{aux}_{L,1,2}\right).\end{aligned} \tag{10}$$

We next claim the following indistinguishability:

$$\begin{aligned}&\left(\mathbf{B}_{L,0}, \mathbf{P}_{L,0}, \mathbf{S}_{L,0}\mathbf{B}_{L,0} + \mathbf{E}_{L,0}, \mathbf{S}_{L,0}\mathbf{P}_{L,0} + \mathbf{E}'_{L,0}, \mathsf{aux}_{L,0,1}, \mathsf{aux}_{L,0,2}\right)\\ &\overset{\mathrm{c}}{\approx} \left(\mathbf{B}_{L,0}, \mathbf{P}_{L,0}, \mathbf{C}_{L,0}, \mathbf{C}'_{L,0}, \widetilde{\mathsf{aux}_{L,0,1}}, \mathsf{aux}_{L,0,2}\right),\end{aligned} \tag{11}$$

where $\mathbf{B}_{L,0} \leftarrow\!\$\ \mathbb{Z}_q^{2(n+1)\times m_B}$, $\mathbf{E}_{L,0} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi}^{2^{L-1}\times m_B}$, $\mathbf{E}'_{L,0} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi_i}^{2^{L-1}\times(m_B+(1+n+L')m)}$, $\mathbf{C}_{L,0} \leftarrow\!\$\ \mathbb{Z}_q^{2^{L-1}\times m_B}$, and $\mathbf{C}'_{L,0} \leftarrow\!\$\ \mathbb{Z}_q^{2^{L-1}\times(m_B+(1+n+L')m)}$. Notably, $\mathbf{S}_{L,0}\mathbf{P}_{L,0}+\mathbf{E}'_{L,0} = (\mathbf{p}_{\mathbf{x}_{h-1}0}^T, \mathbf{v}_{\mathsf{att},\mathbf{x}_{h-1}0}^T, \mathbf{v}_{t,\mathbf{x}_{h-1}0}^T)_{\mathbf{x}_{h-1}\in\{0,1\}^{h-1}}$ is included in $\mathsf{aux}_{h,1,1}$. Following the approach demonstrated above, we can derive the following indistinguishability:

$$\begin{aligned}&\left(\mathbf{B}_{L,0}, \mathbf{P}_{L,0}, \mathbf{S}_{L,0}\mathbf{B}_{L,0} + \mathbf{E}_{L,0}, (\mathbf{N}_{L,0}, \mathbf{K}_{L,0}), \mathsf{aux}_{L,0,1}, \mathsf{aux}_{L,0,2}\right)\\ &\overset{\mathrm{c}}{\approx} \left(\mathbf{B}_{L,0}, \mathbf{P}_{L,0}, \mathbf{C}_{L,0}, (\mathbf{N}_{L,0}, \mathbf{K}_{L,0}), \widetilde{\mathsf{aux}_{L,0,1}}, \mathsf{aux}_{L,0,2}\right).\end{aligned} \tag{12}$$

We finally claim the following indistinguishability:

$$\begin{aligned}&\left(\mathbf{B}_{L-1,\star}, \mathbf{P}_{L-1,\star}, \mathbf{S}_{L-1,\star}\mathbf{B}_{L-1,\star} + \mathbf{E}_{L-1,\star}, \mathbf{S}_{L-1,\star}\mathbf{P}_{L-1,\star} + \mathbf{E}'_{L-1,\star}, \mathsf{aux}_{L-1,\star,1}, \mathsf{aux}_{L-1,\star,2}\right)\\ &\overset{\mathrm{c}}{\approx} \left(\mathbf{B}_{L-1,\star}, \mathbf{P}_{L-1,\star}, \mathbf{C}_{L-1,\star}, \mathbf{C}'_{L-1,\star}, \widetilde{\mathsf{aux}_{L-1,\star,1}}, \mathsf{aux}_{L-1,\star,2}\right),\end{aligned} \tag{13}$$

where $\mathbf{B}_{L-1,\star} \leftarrow\!\$\ \mathbb{Z}_q^{2(n+1)\times m_B}$, $\mathbf{E}_{L-1,\star} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi}^{2^h\times m_B}$, $\mathbf{E}'_{L-1,\star} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi_i}^{2^h\times 2m_B}$, $\mathbf{C}_{L-1,\star} \leftarrow\!\$\ \mathbb{Z}_q^{2^h\times m_B}$, and $\mathbf{C}'_{L-1,\star} \leftarrow\!\$\ \mathbb{Z}_q^{2^h\times 2m_B}$. $\mathbf{S}_{L-1,\star}\mathbf{B}_{L-1,\star} + \mathbf{E}_{L-1,\star} = (\mathbf{p}_{\mathbf{x}_{L-1}})_{\mathbf{x}_{L-1}\in\{0,1\}^{L-1}}$ is included in $\mathsf{aux}_{L,0,1}$, and $\mathbf{S}_{L-1,\star}\mathbf{P}_{L-1,\star} + \mathbf{E}'_{L-1,\star}$—namely, $\mathbf{q}_{\mathbf{x}_{L-1}0}^T$ and $\mathbf{q}_{\mathbf{x}_{L-1}1}^T$—are identical to $\mathbf{S}_{L,0}\mathbf{B}_{L,0} + \mathbf{E}_{L,0}$ and $\mathbf{q}_{\mathbf{x}_{L-1}1}^T$ in $\mathsf{aux}_{L,0}$, respectively. Hence, it follows from Lemma 3 that the indistinguishability in Relation 8 holds for $h = L-1$.

Inductive step. Suppose that the indistinguishability in Relation 8 holds for h, we show that it also holds for $h-1$. We apply Lemma 3 three times in the same manner as the base case.

We first claim the following indistinguishability:

$$\begin{aligned}&\left(\mathbf{B}_{h,1}, \mathbf{P}_{h,1}, \mathbf{S}_{h,1}\mathbf{B}_{h,1} + \mathbf{E}_{h,1}, \mathbf{S}_{h,1}\mathbf{P}_{h,1} + \mathbf{E}'_{h,1}, \mathsf{aux}_{h,1,1}, \mathsf{aux}_{h,1,2}\right)\\ &\overset{\mathrm{c}}{\approx} \left(\mathbf{B}_{h,1}, \mathbf{P}_{h,1}, \mathbf{C}_{h,1}, \mathbf{C}'_{h,1}, \widetilde{\mathsf{aux}_{h,1,1}}, \mathsf{aux}_{h,1,2}\right),\end{aligned} \tag{14}$$

where $\mathbf{B}_{h,1} \leftarrow\!\$\ \mathbb{Z}_q^{2(n+1)\times m_B}$, $\mathbf{E}_{h,1} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi}^{2^{h-1}\times m_B}$, $\mathbf{E}'_{h,1} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi_i}^{2^{h-1}\times(m_B+(1+n+L')m)}$, $\mathbf{C}_{h,1} \leftarrow\!\$\ \mathbb{Z}_q^{2^{h-1}\times m_B}$, and $\mathbf{C}'_{h,1} \leftarrow\!\$\ \mathbb{Z}_q^{2^{h-1}\times(m_B+(1+n+L')m)}$. The indistinguishability in Relation 14 follows immidiately from that in Relation 8 for h. Thus, it holds that

$$\begin{aligned}&\left(\mathbf{B}_{h,1}, \mathbf{P}_{h,1}, \mathbf{S}_{h,1}\mathbf{B}_{h,1} + \mathbf{E}_{h,1}, (\mathbf{N}_{h,1}, \mathbf{K}_{h,1}), \mathsf{aux}_{h,1,1}, \mathsf{aux}_{h,1,2}\right)\\ &\overset{c}{\approx} \left(\mathbf{B}_{h,1}, \mathbf{P}_{h,1}, \mathbf{C}_{h,1}, (\mathbf{N}_{h,1}, \mathbf{K}_{h,1}), \widetilde{\mathsf{aux}_{h,1,1}}, \mathsf{aux}_{h,1,2}\right).\end{aligned} \tag{15}$$

We next claim the following indistinguishability:

$$\begin{aligned}&\left(\mathbf{B}_{h,0}, \mathbf{P}_{h,0}, \mathbf{S}_{h,0}\mathbf{B}_{h,0} + \mathbf{E}_{h,0}, \mathbf{S}_{h,0}\mathbf{P}_{h,0} + \mathbf{E}'_{h,0}, \mathsf{aux}_{h,0,1}, \mathsf{aux}_{h,0,2}\right)\\ &\overset{c}{\approx} \left(\mathbf{B}_{h,0}, \mathbf{P}_{h,0}, \mathbf{C}_{h,0}, \mathbf{C}'_{h,0}, \widetilde{\mathsf{aux}_{h,0,1}}, \mathsf{aux}_{h,0,2}\right),\end{aligned} \tag{16}$$

where $\mathbf{B}_{h,0} \leftarrow\!\$\ \mathbb{Z}_q^{2(n+1)\times m_B}$, $\mathbf{E}_{h,0} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi}^{2^{h-1}\times m_B}$, $\mathbf{E}'_{h,0} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi_i}^{2^{h-1}\times(m_B+(1+n+L')m)}$, $\mathbf{C}_{h,0} \leftarrow\!\$\ \mathbb{Z}_q^{2^{h-1}\times m_B}$, and $\mathbf{C}'_{h,0} \leftarrow\!\$\ \mathbb{Z}_q^{2^{h-1}\times(m_B+(1+n+L')m)}$.

Similarly, this holds by the indistinguishability in Relation 15. Thus, it holds that

$$\begin{aligned}&\left(\mathbf{B}_{h,0}, \mathbf{P}_{h,0}, \mathbf{S}_{h,0}\mathbf{B}_{h,0} + \mathbf{E}_{h,0}, (\mathbf{N}_{h,0}, \mathbf{K}_{h,0}), \mathsf{aux}_{h,0,1}, \mathsf{aux}_{h,0,2}\right)\\ &\overset{c}{\approx} \left(\mathbf{B}_{h,0}, \mathbf{P}_{h,0}, \mathbf{C}_{h,0}, (\mathbf{N}_{h,0}, \mathbf{K}_{h,0}), \widetilde{\mathsf{aux}_{h,0,1}}, \mathsf{aux}_{h,0,2}\right).\end{aligned} \tag{17}$$

Finally, we claim the following indistinguishability:

$$\begin{aligned}&\left(\mathbf{B}_{h-1,\star}, \mathbf{P}_{h-1,\star}, \mathbf{S}_{h-1,\star}\mathbf{B}_{h-1,\star} + \mathbf{E}_{h-1,\star}, \mathbf{S}_{h-1,\star}\mathbf{P}_{h-1,\star} + \mathbf{E}'_{h-2,\star}, \mathsf{aux}_{h-2,\star,1}, \mathsf{aux}_{h-2,\star,2}\right)\\ &\overset{c}{\approx} \left(\mathbf{B}_{h-1,\star}, \mathbf{P}_{h-1,\star}, \mathbf{C}_{h-1,\star}, \mathbf{C}'_{h-1,\star}, \widetilde{\mathsf{aux}_{h-1,\star,1}}, \mathsf{aux}_{h-1,\star,2}\right),\end{aligned} \tag{18}$$

where $\mathbf{B}_{h-1,\star} \leftarrow\!\$\ \mathbb{Z}_q^{2(n+1)\times m_B}$, $\mathbf{E}_{h-1,\star} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi}^{2^{h-1}\times m_B}$, $\mathbf{E}'_{h-1,\star} \leftarrow\!\$\ \mathcal{D}_{\mathbb{Z},\chi_i}^{2^{h-1}\times 2m_B}$, $\mathbf{C}_{h-1,\star} \leftarrow\!\$\ \mathbb{Z}_q^{2^{h-1}\times m_B}$, and $\mathbf{C}'_{h-1,\star} \leftarrow\!\$\ \mathbb{Z}_q^{2^{h-1}\times 2m_B}$. This indistinguishability follows directly from that in Relation 17 for h. Thus, by Lemma 3, the indistinguishability in Relation 8 also holds for $h-1$.

The above discussion derives the pseudorandomness of the L.H.S. distribution in Relation 8 for $h = 0$. Notably, this distribution is same as that of the obfuscated circuit $\widetilde{C} = \mathsf{Obf}(1^\lambda, C)$ except for the uniformly random matrixes $\{\mathbf{B}_{i.b}\}_{i\in[L],b\in\{0,1,\star\}}$, implying the pseudorandomness of $\mathsf{Obf}(1^\lambda, C)$. Since the pseudorandom version of $\mathsf{Obf}(1^\lambda, C)$ is independent of the choice of whether C_0 or C_1 is obfuscated, the theorem follows. □

5 Conclusion

Despite recent progress in constructing iO from reasonable cryptographic assumptions, it remains a theoretical rather than a practical cryptographic primitive due to its complexity and inefficiency. We observe that a common bottleneck in recently proposed iO schemes is their reliance on bootstrapping techniques from FE to iO. Specifically, they recursively invoke the FE encryption algorithm for each input bit as a function evaluated during decryption in the FE scheme. Consequently, the size and complexity of the FE encryption algorithm become a lower bound on the size of the functions that must be evaluated by the underlying FE scheme.

To address this bottleneck, we propose diamond iO, a straightforward construction of iO using lattice techniques. Our construction replaces the costly FE recursive encryption process for every input bit required in those bootstrapping techniques with simple matrix operations, by leveraging the FE scheme for pseudorandom functionalities proposed in [AKY24a]. We prove the security of our construction under LWE and evasive LWE, along with a new lattice assumption we introduce—called all-product LWE—in the PROM. A remaining challenge is to reduce this new assumption to standard ones such as LWE, thereby further advancing the goal of a practical and sound iO construction.

References

[AFH+20] Martin R Albrecht, Pooya Farshim, Shuai Han, Dennis Hofheinz, Enrique Larraia, and Kenneth G Paterson. Multilinear maps from obfuscation. *Journal of Cryptology*, 33(3):1080–1113, 2020.

[Agr19] Shweta Agrawal. Indistinguishability obfuscation without multilinear maps: new methods for bootstrapping and instantiation. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part I 38*, pages 191–225. Springer, 2019.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In *Annual Cryptology Conference*, pages 308–326. Springer, 2015.

[AJL+19] Prabhanjan Ananth, Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. Indistinguishability obfuscation without multilinear maps: new paradigms via low degree weak pseudorandomness and security amplification. In *Annual International Cryptology Conference*, pages 284–332. Springer, 2019.

[AKY24a] Shweta Agrawal, Simran Kumari, and Shota Yamada. Compact pseudorandom functional encryption from evasive lwe. *Cryptology ePrint Archive*, 2024.

[AKY24b] Shweta Agrawal, Simran Kumari, and Shota Yamada. Pseudorandom multi-input functional encryption and applications. *Cryptology ePrint Archive*, 2024.

[AS17] Prabhanjan Ananth and Amit Sahai. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 152–181. Springer, 2017.

[BDGM20a] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Candidate io from homomorphic encryption schemes. In *Advances in Cryptology - EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part I*, pages 79–109, Berlin, Heidelberg, 2020. Springer-Verlag.

[BDGM20b] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for io: Circular-secure lwe suffices. *Cryptology ePrint Archive*, 2020.

[BDJ+24] Pedro Branco, Nico Döttling, Abhishek Jain, Giulio Malavolta, Surya Mathialagan, Spencer Peters, and Vinod Vaikuntanathan. Pseudorandom obfuscation and applications. *Cryptology ePrint Archive*, 2024.

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits. In *Advances in Cryptology–EUROCRYPT 2014: 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings 33*, pages 533–556. Springer, 2014.

[BGI+01] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. In *Annual international cryptology conference*, pages 1–18. Springer, 2001.

[BGL+15] Nir Bitansky, Sanjam Garg, Huijia Lin, Rafael Pass, and Sidharth Telang. Succinct randomized encodings and their applications. In *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*, pages 439–448, 2015.

[BP15] Nir Bitansky and Omer Paneth. Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In *Theory of Cryptography: 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II 12*, pages 401–427. Springer, 2015.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *Theory of Cryptography: 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings 8*, pages 253–273. Springer, 2011.

[BTVW17] Zvika Brakerski, Rotem Tsabary, Vinod Vaikuntanathan, and Hoeteck Wee. Private constrained prfs (and more) from lwe. In *Theory of Cryptography Conference*, pages 264–302. Springer, 2017.

[BÜW24] Chris Brzuska, Akin Ünal, and Ivy KY Woo. Evasive lwe assumptions: Definitions, classes, and counterexamples. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 418–449. Springer, 2024.

[BV18] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. *Journal of the ACM (JACM)*, 65(6):1–37, 2018.

[BZ17] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. *Algorithmica*, 79:1233–1285, 2017.

[CDCG+18] David Bruce Cousins, Giovanni Di Crescenzo, Kamil Doruk Gür, Kevin King, Yuriy Polyakov, Kurt Rohloff, Gerard W Ryan, and Erkay Savas. Implementing conjunction obfuscation under entropic ring lwe. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 354–371. IEEE, 2018.

[DQV+21] Lalita Devadas, Willy Quach, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Succinct lwe sampling, random polynomials, and obfuscation. In *Theory of Cryptography: 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8–11, 2021, Proceedings, Part II 19*, pages 256–287. Springer, 2021.

[GGG+14] Shafi Goldwasser, S Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In *Advances in Cryptology–EUROCRYPT 2014: 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings 33*, pages 578–602. Springer, 2014.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *Theory of Cryptography: 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II 12*, pages 498–527. Springer, 2015.

[GGH+16] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM Journal on Computing*, 45(3):882–929, 2016.

[GJLS21] Romain Gay, Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from simple-to-state hard problems: New assumptions, new techniques, and simplification. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 97–126. Springer, 2021.

[GM18] Nicholas Genise and Daniele Micciancio. Faster gaussian sampling for trapdoor lattices with arbitrary modulus. In *Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part I 37*, pages 174–203. Springer, 2018.

[GP21] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 736–749, 2021.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206, 2008.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 75–92. Springer, 2013.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from lwe. In *Annual Cryptology Conference*, pages 503–523. Springer, 2015.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HJL21] Sam Hopkins, Aayush Jain, and Huijia Lin. Counterexamples to new circular security assumptions underlying io. In *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part II 41*, pages 673–700. Springer, 2021.

[HLL23] Yao-Ching Hsieh, Huijia Lin, and Ji Luo. Attribute-based encryption for circuits of unbounded depth from lattices: Garbled circuits of optimal size, laconic functional evaluation, and more. *Cryptology ePrint Archive*, 2023.

[HLL24] Yao-Ching Hsieh, Huijia Lin, and Ji Luo. A general framework for lattice-based abe using evasive inner-product functional encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 433–464. Springer, 2024.

[JLLS23] Aayush Jain, Huijia Lin, Paul Lou, and Amit Sahai. Polynomial-time cryptanalysis of the subspace flooding assumption for post-quantum i o. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 205–235. Springer, 2023.

[JLLW23] Aayush Jain, Huijia Lin, Ji Luo, and Daniel Wichs. The pseudorandom oracle model and ideal obfuscation. In *Annual International Cryptology Conference*, pages 233–262. Springer, 2023.

[JLMS19] Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. How to leverage hardness of constant-degree expanding polynomials over r r to build i o i o. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part I 38*, pages 251–281. Springer, 2019.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 60–73, 2021.

[JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from lpn over f p, dlin, and prgs in nc 0. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 670–699. Springer, 2022.

[Lin16] Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. 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 I 35*, pages 28–57. Springer, 2016.

[LPST16] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation with non-trivial efficiency. In *Public-Key Cryptography–PKC 2016*, pages 447–462. Springer, 2016.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 700–718. Springer, 2012.

[PST14] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In *Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I 34*, pages 500–517. Springer, 2014.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[RVV25] Seyoon Ragavan, Neekon Vafa, and Vinod Vaikuntanathan. Indistinguishability obfuscation from bilinear maps and lpn variants. In *Theory of Cryptography Conference*, pages 3–36. Springer, 2025.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 475–484, 2014.

[Tsa22] Rotem Tsabary. Candidate witness encryption from lattice techniques. In *Annual International Cryptology Conference*, pages 535–559. Springer, 2022.

[VWW22] Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-io from evasive lwe. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 195–221. Springer, 2022.

[Wee22] Hoeteck Wee. Optimal broadcast encryption and cp-abe from evasive lattice assumptions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 217–241. Springer, 2022.

[WW21] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious lwe sampling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 127–156. Springer, 2021.

[WWW22] Brent Waters, Hoeteck Wee, and David J Wu. Multi-authority abe from lattices without random oracles. In *Theory of Cryptography Conference*, pages 651–679. Springer, 2022.