

Tree Encodings IV: Depth-Unbounded Attribute-Based Encryption and Delay Encryption

Damiano Abram*

Giulio Malavolta[†]

Lawrence Roy[‡]

Abstract

Assuming the polynomial-time hardness of Decomposed LWE, a variant of the learning with errors (LWE) problem, we construct ciphertext-policy attribute-based encryption for depth-unbounded (but bounded-space) predicates. Previously, attribute-based encryption for depth-unbounded predicates was only known from an insecure version of evasive LWE, or by additionally assuming the cryptographic hardness of discrete logarithms, which makes such schemes quantum-insecure.

Adapting these techniques, we also obtain the first construction of lattice-based delay encryption [Burdges and De Feo, EUROCRYPT’21], additionally assuming the existence of *any* space-bounded sequential function. Finally, as a contribution of independent interest, we show how to instantiate the latter from a circular variant of decomposed LWE. This also gives the first delay encryption that does not require explicit sequentiality assumptions.

1 Introduction

Ajtai’s pioneering work [Ajt96] showed how to base cryptographic hardness on the worst-case difficulty of lattice problems and ushered in a revolution in modern cryptography. Shortly after, the work of Regev [Reg05] introduced the learning with errors (LWE) problem, which asks one to recover a secret vector from random linear equations modulo an integer, each perturbed by a small error. LWE combines rich algebraic structure with worst-case hardness guarantees, and is conjectured to remain hard even for quantum algorithms.

These properties enabled cryptographic tasks well beyond ordinary encryption. For example, lattices enabled computation over encrypted data through the introduction of fully-homomorphic encryption (FHE) [Gen09, BV11, GSW13], attribute-based encryption (ABE) [GVW13, BGG⁺14], predicate encryption [GVW15], delegation of computation [CJJ22] and even protocols in which a classical verifier checks the work of a quantum computer [Mah18], to mention a few.

A central ingredient in many of these developments is *homomorphic computation*: Algebraic operations on ciphertexts or other encodings implement corresponding operations on the values they represent. The small errors that provide security, however, also grow during evaluation. Controlling this growth often requires fixing a bound on the *depth* of the computation, namely the maximum number of gates along an input-to-output path in its circuit. In a depth-bounded scheme, this bound must be chosen when the cryptographic parameters are generated and often the resulting keys or ciphertexts grow with such a bound, which is undesirable from an efficiency standpoint.

*University of Edinburgh

[†]Bocconi University

[‡]IBM Research

This limitation is particularly restrictive for a program that repeatedly updates a small state: Its memory use can remain modest while the circuit obtained by writing out all of its steps has large depth. This motivates studying cryptographic schemes supporting the evaluation of Turing machines, an approach pioneered by Goldwasser et al. [GKP⁺13].

FHE illustrates well the progress and the remaining difficulty. *Leveled* fully homomorphic encryption supports any prescribed polynomial depth and can be based on LWE [BV11, BGV12] whereas Gentry’s *bootstrapping* technique [Gen09] refreshes a noisy ciphertext by homomorphically evaluating the decryption algorithm, using an encryption of the decryption key. However, it requires assuming *circular security* and, at present, it is open how to base the circular security needed for bootstrapping on ordinary LWE.

The analogous situation appears in the literature on ABE: Starting with the first lattice constructions for circuits [GVW13, BGG⁺14], a substantial line of work has improved security, expressiveness, and succinctness while retaining a prescribed circuit-depth bound [BV16, Wee22, CW23, HLL24, Wee24, Wee25]. Further work studies policies described by Turing machines and random-access programs, whose descriptions can be much smaller than their full computation circuits [DHM⁺24, AKY24, CW25, AMR25a]. Most of the aforementioned works are in the depth-bounded settings (akin to leveled FHE) with the notable exception of a recent result by Hsieh, Lin, and Luo [HLL23], who introduced a bootstrapping procedure for lattice encodings. However, proving security of the resulting ABE requires a version of the evasive LWE assumption [Wee22, VWV22]. This general approach requires care: Several forms of evasive LWE admit counterexamples [BÜW24, DJM⁺25], including the evasive circular assumption used for unbounded-depth ABE [AMYY25]. More recently, Agrawal, Modi, Yadav, and Yamada [AMYY26] constructed ABE for unbounded-depth circuits under a new *doubly circular* assumption. Their assumption couples circular LWE with a strong form of circular security of an ElGamal-type encryption scheme, based on groups.

To summarize, the *depth-unbounded* settings for homomorphic computations is far less understood and comes with unique challenges. Our objective is to place lattice-based depth-unbounded homomorphic computation on a more concrete foundation. We shall show how the tree-encoding framework from [AMR26a, AMR26b, AMR26c] allows us to make progress on this long-standing problem.

1.1 Our Results

In this work we construct new cryptographic primitives for *depth-unbounded* computation. All of our schemes assume the hardness of the *decomposed LWE* problem [AMR25a]. Decomposed LWE is a variant of the traditional LWE problem, where the coefficient matrix has a particular (publicly known) low-rank structure. The related succinct LWE assumption was introduced by Wee [Wee24], and the two assumptions are equivalent under appropriate parameter settings [AAC⁺26]. Together, they have enabled a growing range of cryptographic constructions [Wee24, Wee25, CHW25, WW25, AMR25a, LLL26, CW26].

Result I: Depth-unbounded ABE for Turing Machines. We construct *ciphertext-policy* ABE, where a secret key is associated with an attribute string x , and the sender places an access policy f in the ciphertext. Policies are polynomial-time Turing machine computations with a prescribed bound on their memory. Equivalently, a policy is specified by a transition circuit C_f acting on a d -bit state and an iteration count T ; its decision is a designated bit after T updates.

Here d captures the space bound, whereas T captures the running time; $|C_f|$ denotes the transition circuit's size.

Theorem 1.1 (Informal). *Assuming decomposed LWE , there exists selectively secure ciphertext-policy ABE for polynomial-time Turing machine computations with an a-priori fixed space bound. For a policy described by (C_f, T) , encryption runs in $\text{poly}(\lambda, d, |C_f|, \log T)$ time, and decryption runs in $\text{poly}(\lambda, d, |C_f|, T)$ time.*

This construction is described in detail in [Section 3](#). It relies on the iterated lattice encodings of [\[AMR26c\]](#) to control the growth of the error over a long computation. In a sense, this yields the complementary tradeoff compared to [\[BGG⁺14\]](#): We fix a bound on the available space but not the time, the opposite of [\[BGG⁺14\]](#).

A modification yields attribute-based laconic function evaluation [\[QWW18\]](#), in which a function is compressed into a short digest used to encrypt messages subject to that function. For the same space-bounded computations, hashing runs in $\text{poly}(\lambda, d, |C_f|, \log T)$ time, giving polynomial dependence on $\log T$ and improving on the corresponding hashing procedures of [\[QWW18, AMR25b, AMR25a\]](#) whose runtime is polynomial in T .

Result II: Delay Encryption. Delay encryption, introduced by Burdges and De Feo [\[BD21\]](#), associates a publicly computable decryption key with a *session identifier*. Anyone can quickly encrypt a message for a session, whereas deriving its decryption key requires a long computation. This improves the traditional notion of a time-lock puzzle [\[RSW96\]](#), where the delayed computation is tied to an individual puzzle.

We obtain delay encryption from decomposed LWE and an iterative *sequential function*. To our knowledge, this is the first construction of delay encryption that is based on lattice problems, whereas prior work [\[BD21\]](#) assumed the hardness of problems related to isogenies on elliptic curves. Previous lattice-based time-lock puzzles [\[AMZ24, BG25, AMR25a, ARS24\]](#) did not support the advanced functionality of delay encryption.¹

Theorem 1.2 (Informal). *Assume decomposed LWE and the existence of a space-bounded iterative sequential function. Then there exists a delay-encryption scheme with setup and encryption time polynomial in the security parameter and the logarithm of the iteration count T .*

The precise sequentiality assumption, security experiment, and reduction appear in [Section 4](#). Notice that, in addition to decomposed LWE , the construction requires the existence of a space-bounded sequential function. While candidate lattice-based sequentiality assumptions have been proposed [\[LM23\]](#), we currently lack a construction of the required space-bounded sequential function from a traditional cryptographic assumption. Our next result makes progress on this question.

Result III: A Sequential Function from Circular Decomposed LWE . Finally, we propose a lattice-based source of the bounded-space sequential computations used above. Here we assume the hardness of *small-secret circular decomposed LWE* [\[AMR25a\]](#), a circular version of the Decomposed LWE problem.

Theorem 1.3 (A hard sequential function, informal). *Assume small-secret circular decomposed LWE . For every polynomially-bounded T , there is an efficiently samplable input distribution and*

¹The work of [\[ARS24\]](#) proposes a delay encryption where encryption is low-depth and deriving the decryption key requires high depth, however, their ciphertext size grows proportionally to the depth parameter.

a function f computable by T iterations, with polynomial overhead per iteration and state size polynomial in λ , such that no polynomial-time adversary running for $T/O(\log \lambda)$ steps computes its output with non-negligible probability on a freshly sampled input.

The construction and quantitative lower bound appear in [Section 5](#). This improves upon the very recent work [\[AMR26c\]](#) which also proposed a sequential function, but with a large state $O(T)$. This comes at the cost of integrating the bootstrapping procedure of [\[HLL23\]](#) within the tree encodings, which results in an additional circularity assumption.

1.2 Technical Outline

Let $f : \{0, 1\}^d \rightarrow \{0, 1\}^d$ be a transition function on a d -bit state. Write $f^{(T)}(x)$ for the state obtained by applying f to x exactly T times, and let $g_i(x) = f^{(T)}(x)_i$ denote its i -th output bit. Our starting point is the iterated lattice encoding of [\[AMR26c\]](#). Its public parameters $\mathbf{pk}$ contain a matrix $\mathbf{Z}$ from the decomposed-LWE sampler and the gadget matrix $\mathbf{G} = \mathbf{I}_k \otimes (1, 2, \dots, q/2)$, where q is a power of two and $n = k \log q$ is the number of columns of $\mathbf{G}$.

The work [\[AMR26c\]](#) presents three algorithms, all given the public parameters $\mathbf{pk}$. The algorithm `ExpandP` takes x and computes an input-dependent matrix $\mathbf{P}_x$. The algorithm `EvalM` takes f, T and computes public matrices $\{\mathbf{M}_{g_i}\}_i$. Finally, `EvalH` evaluates the T steps of f on x and returns $f^{(T)}(x)$ together with matrices $\{\mathbf{H}_{g_i, x}\}_i$ satisfying

$$\mathbf{G}\mathbf{M}_{g_i}\mathbf{P}_x + \mathbf{Z}\mathbf{H}_{g_i, x} = g_i(x)\mathbf{G}. \quad (1)$$

The matrix $\mathbf{M}_{g_i}$ represents the computation independently of its input, while $\mathbf{H}_{g_i, x}$ connects this representation to the actual output on x . The matrices $\mathbf{M}_{g_i}$ are binary, and $\mathbf{P}_x$ and $\mathbf{H}_{g_i, x}$ have low norm. The algorithm `ExpandP` takes time polynomial in λ, d , and `EvalM` takes time polynomial in $\lambda, d, |f|$, and $\log T$, whereas `EvalH` performs the long computation and takes time polynomial in $\lambda, d, |f|$, and T . For fixed lattice parameters, the matrix dimensions stay fixed across iterations.

Attribute-Based Encryption. We first use these encodings to construct ciphertext-policy ABE, following Wee [\[Wee25\]](#). A policy g is one of the output predicates above, with $g(x) = 0$ meaning that attribute x is authorized. Setup samples a nearly uniform matrix $\mathbf{A}$ together with a trapdoor, and independent uniform matrices $\mathbf{B}, \mathbf{V}$. The trapdoor is secret information for sampling short solutions to linear equations. Setup publishes the matrices alongside $\mathbf{pk}$ and retains the trapdoor as the master secret key. Using it, key generation for x samples a short matrix $\mathbf{D}_x$ satisfying

$$[\mathbf{A} \mid \mathbf{Z} \mid \mathbf{B}\mathbf{P}_x]\mathbf{D}_x = \mathbf{V}. \quad (2)$$

To encrypt $\mu \in \{0, 1\}^n$ under g , sample a uniform secret $\mathbf{s} \in \mathbb{Z}_q^k$ and send g together with

$$\begin{aligned} \mathbf{c}_A^\top &\approx \mathbf{s}^\top \mathbf{A}, & \mathbf{c}_Z^\top &\approx \mathbf{s}^\top \mathbf{Z}, \\ \mathbf{c}_B^\top &\approx \mathbf{s}^\top (\mathbf{B} - \mathbf{G}\mathbf{M}_g), & \mathbf{c}_V^\top &\approx \mathbf{s}^\top \mathbf{V} + \frac{q}{2}\mu^\top. \end{aligned}$$

Decryption runs `EvalH` and uses [Equation \(1\)](#) to form $\hat{\mathbf{c}}_x^\top := \mathbf{c}_B^\top \mathbf{P}_x - \mathbf{c}_Z^\top \mathbf{H}_{g, x}$. This is approximately $\mathbf{s}^\top (\mathbf{B}\mathbf{P}_x - g(x)\mathbf{G})$. For an authorized attribute, $g(x) = 0$, so [Equation \(2\)](#) gives

$$\mathbf{c}_V^\top - [\mathbf{c}_A^\top \mid \mathbf{c}_Z^\top \mid \hat{\mathbf{c}}_x^\top]\mathbf{D}_x \approx \frac{q}{2}\mu^\top.$$

Rounding each coordinate to the nearer of 0 and $q/2$ recovers the message. Encryption needs only the compact matrix $\mathbf{M}_g$; decryption carries out the policy computation.

For selective security, the adversary commits to the challenge policy g^* before setup and may request keys only for attributes with $g^*(x) = 1$. The reduction chooses a random short binary matrix $\mathbf{U}$ and sets $\mathbf{B} = \mathbf{A}\mathbf{U} + \mathbf{G}\mathbf{M}_{g^*}$. This preserves the public-key distribution up to negligible statistical distance. Every allowed query now provides the following short solution:

$$[\mathbf{A} \mid \mathbf{Z} \mid \mathbf{B}\mathbf{P}_x] \begin{bmatrix} -\mathbf{U}\mathbf{P}_x \\ \mathbf{H}_{g^*,x} \\ \mathbf{I}_n \end{bmatrix} = \mathbf{G}\mathbf{M}_{g^*}\mathbf{P}_x + \mathbf{Z}\mathbf{H}_{g^*,x} = \mathbf{G}.$$

This is a gadget trapdoor in the sense of Micciancio and Peikert [MP12]. It permits sampling $\mathbf{D}_x$ with the same distribution as ordinary key generation, up to negligible statistical distance. For an authorized attribute the product would instead be zero, so this argument supplies keys precisely for the allowed queries. Meanwhile, $\mathbf{B} - \mathbf{G}\mathbf{M}_{g^*} = \mathbf{A}\mathbf{U}$ allows the reduction to simulate $\mathbf{c}_B$ from $\mathbf{c}_A$ and $\mathbf{U}$, adding fresh noise to mask the propagated error. Decomposed LWE makes the noisy samples for $\mathbf{Z}$, $\mathbf{A}$, and $\mathbf{V}$ indistinguishable from uniform; the last of these hides μ .

Delay Encryption. We use a similar idea, now assuming that f is a sequential function: computing $y = f^{(T)}(\alpha)$ for a fresh random input α cannot be done in time significantly less than T , even after preprocessing independent of α . The public-coin setup publishes the matrices $\mathbf{M}_{g_i}$ for the output bits of $f^{(T)}$. Running `EvalH` on a session identifier α produces the session key $\mathbf{dk}_\alpha = (y, \{\mathbf{H}_{g_i, \alpha}\}_i)$. We arrange that this same key decrypts every ciphertext for α .

To encrypt a d -bit message μ , we garble the constant circuit $C_\mu(v) = \mu$ [Yao86, BHR12]. Garbling assigns two independent random labels $L_{i,0}, L_{i,1} \in \{0,1\}^n$ to each input position i . One label per position allows evaluation on the corresponding input. We use the privacy guarantee that these selected labels and the garbled circuit reveal only its output, apart from a padded size. The sender encrypts both labels at each position without computing y . For each (i, b) , it independently samples a uniform matrix $\mathbf{U}_{i,b} \in \mathbb{Z}_q^{k \times n}$ and a uniform secret $\mathbf{s}_{i,b} \in \mathbb{Z}_q^k$, and publishes $\mathbf{U}_{i,b}$ together with

$$\begin{aligned} \mathbf{z}_{i,b}^\top &\approx \mathbf{s}_{i,b}^\top \mathbf{Z}, \\ \mathbf{c}_{i,b}^\top &\approx \mathbf{s}_{i,b}^\top \mathbf{G}(\mathbf{M}_{g_i} \mathbf{P}_\alpha - b \mathbf{I}_n) \mathbf{G}^{-1}(\mathbf{U}_{i,b}) + \frac{q}{2} L_{i,b}^\top. \end{aligned}$$

Writing $\mathbf{H}_i = \mathbf{H}_{g_i, \alpha}$, Equation (1) gives

$$\mathbf{c}_{i,b}^\top + \mathbf{z}_{i,b}^\top \mathbf{H}_i \mathbf{G}^{-1}(\mathbf{U}_{i,b}) \approx (y_i - b) \mathbf{s}_{i,b}^\top \mathbf{U}_{i,b} + \frac{q}{2} L_{i,b}^\top.$$

When $b = y_i$, rounding recovers L_{i,y_i} , and the garbled circuit returns $C_\mu(y) = \mu$. When $b \neq y_i$, the coefficient $y_i - b$ is 1 or -1 . Together with $\mathbf{z}_{i,b}$, the remaining term has the form of decomposed-LWE samples for $\mathbf{Z}$ and the independent uniform matrix $(y_i - b) \mathbf{U}_{i,b}$. Thus the incorrect-label ciphertexts can be replaced by uniform components. These hidden labels are what make the security proof work, even though the circuit being garbled is constant.

Next, we prove that the scheme satisfies one-wayness, namely that it is hard to return a uniformly random encrypted message in time much smaller than T . The Goldreich–Levin transformation [GL89] upgrades one-wayness to the standard message indistinguishability, as shown in [AMZ24]. As the first step in the proof, we syntactically rewrite the random message as $r \oplus y$, with r uniform. After hiding all incorrect labels, replace the constant circuit $C_{r \oplus y}$ by $D_r(v) = r \oplus v$. Both return $r \oplus y$ on the selected input y , so garbling privacy hides the change. Restore the incorrect-label ciphertexts using decomposed LWE. This final experiment can be generated knowing only r and α : The sender garbles D_r and encrypts both possible labels, without evaluating $f^{(T)}$. An inverter that recovers the output $\mu' = r \oplus f^{(T)}(\alpha)$ then reveals $f^{(T)}(\alpha) = r \oplus \mu'$. This contradicts sequentiality, accounting for the compact ciphertext-generation overhead.

A Lattice-Based Sequential Function. In [AMR26c], Abram, Malavolta and Roy show how to construct a function that, solely under the hardness of decomposed SIS, cannot be evaluated in low-depth. Our hope is to instantiate the delay encryption construction described above with such a function obtaining a construction of delay encryption that relies solely on the hardness of decomposed LWE. Unfortunately, this does not really work: The depth-hard function identified by [AMR26c] has input length that is proportional to the depth required to evaluate the function. This would make the delay encryption useless because, even if encryption is still low-depth and decryption requires high-depth computations, the size of the ciphertext is proportional to the sequential time required to decrypt. It is not even clear how to efficiently send such a huge ciphertext!

In this work, we modified the construction of [AMR26c] to derive a sequential function with small input. The approach follows the same blueprint as [AMR26c]: First, we build a hash function Hash that is correlation intractable [CGH98] for any relation

$$\mathcal{R}_{\mathcal{M}} := \left\{ \left((1^T, x), \mathcal{M}^{(T)}(x) \right) \mid T \in \mathbb{N}, x \in \{0, 1\}^* \right\}$$

where $\mathcal{M}$ is a Turing machine; then, we use a diagonalization argument from [AMR26c] to show that the function $\text{hk} \mapsto \text{Hash}(\text{hk}, (1^T, \text{hk}))$ cannot be computed within T steps by any Turing machine. Indeed, if such a Turing machine $\overline{\mathcal{M}}$ existed, we could break the intractability of the correlation of Hash with $\mathcal{R}_{\overline{\mathcal{M}}}$ simply by outputting $(1^T, \text{hk})$.

Next, we show how to build a correlation intractable hash function with the described properties from circular decomposed LWE. Fix any Turing machine $\mathcal{M}$. The high-level blueprint is the same as previous lattice-based CI hashing constructions [PS19]. We include decomposed LWE public parameters in the hash key. Given $\mathbf{z}^\top \approx \mathbf{s}^\top \mathbf{Z}$, using EvalH , one can derive an encoding

$$\mathbf{c}_{\mathcal{M}^{(T)}(x)}^\top \approx \mathbf{s}^\top (\mathbf{G}\mathbf{M}_{\mathcal{M}^{(T)}(x)} - \mathcal{M}^{(T)}(x)^\top \otimes \mathbf{G}),$$

where $\mathbf{M}_{\mathcal{M}^{(T)}(x)}$ is obtained via EvalM (we will explain the details of these computations below). By doing some restructuring, the encoding $\mathbf{c}_{\mathcal{M}^{(T)}(x)}$ can be converted into

$$\tilde{\mathbf{c}}_{\mathcal{M}^{(T)}(x)}^\top \approx \mathbf{s}^\top (\mathbf{G}\tilde{\mathbf{m}}_{\mathcal{M}^{(T)}(x)} - \mathbf{G}\mathcal{M}^{(T)}(x)),$$

for some efficiently computable vector $\tilde{\mathbf{m}}_{\mathcal{M}^{(T)}(x)}$.

Our idea is to set the hash of $(1^T, x)$ to be the bit representation of the vector $\mathbf{G}\tilde{\mathbf{m}}_{\mathcal{M}^{(T)}(x)}$. This means that any successful adversary against the correlation intractability property would provide us with T and x such that

$$\mathbf{G}\tilde{\mathbf{m}}_{\mathcal{M}^{(T)}(x)} = \mathbf{G}\mathcal{M}^{(T)}(x),$$

and hence $\tilde{\mathbf{c}}_{\mathcal{M}^{(T)}(x)} \approx \mathbf{0}$. This violates the hardness of decomposed LWE as it provides a way to distinguish the DLWE sample $\mathbf{z}$ from a random vector.

The Hash Function in Detail. Before we move on to explaining how to derive $\mathbf{M}_{\mathcal{M}^{(T)}(x)}$, we need to highlight some basic facts about the encodings of [AMR26c]. The first fact is that they allow multi-hop operations: Given $x, f_0(x), f_1(x), \mathbf{M}_{f_0}, \mathbf{M}_{f_1}, \mathbf{c}_{f_0}$ and $\mathbf{c}_{f_1}$ such that

$$\mathbf{c}_{f_0}^\top \approx \mathbf{s}^\top (\mathbf{G}\mathbf{M}_{f_0}\mathbf{P}_x - f_0(x)\mathbf{G}), \quad \mathbf{c}_{f_1}^\top \approx \mathbf{s}^\top (\mathbf{G}\mathbf{M}_{f_1}\mathbf{P}_x - f_1(x)\mathbf{G}),$$

and another function $g : \{0, 1\}^2 \rightarrow \{0, 1\}$, we can derive $\mathbf{c}_{g \circ (f_0, f_1)}$ such that

$$\mathbf{c}_{g \circ (f_0, f_1)}^\top \approx \mathbf{s}^\top (\mathbf{G}\mathbf{M}_{g \circ (f_0, f_1)}\mathbf{P}_x - g(f_0(x), f_1(x))\mathbf{G}),$$

where $\mathbf{M}_{g \circ (f_0, f_1)}$ can be efficiently derived from $\mathbf{M}_{f_0}$, $\mathbf{M}_{f_1}$ and g . This means that $\mathbf{M}_{f_0}$ and $\mathbf{M}_{f_1}$ can be viewed as commitments to the truth tables of f_0 and f_1 with the homomorphic property that they can be converted into a commitment to the truth table of $g \circ (f_0, f_1)$. This comes at the cost of an increase in the noise magnitude by a $\text{poly}(\lambda)$ multiplicative factor for each hop.

The second fact we highlight is that the encodings of [AMR26c] support *read* operations: Given $\alpha, \beta \in \{0, 1\}^d$ and encodings $(\mathbf{c}_i)_{i \in [d]}$ and $\mathbf{c}$ such that

$$\mathbf{c}_i^\top \approx \mathbf{s}^\top (\mathbf{G} \mathbf{M}_i \mathbf{P}_\alpha - \beta_i \mathbf{G}), \quad \mathbf{c}^\top \approx \mathbf{s}^\top (\mathbf{G} \mathbf{M} \mathbf{P}_\beta - x \mathbf{G})$$

we can efficiently derive an encoding

$$\tilde{\mathbf{c}}^\top \approx \mathbf{s}^\top (\widetilde{\mathbf{G} \mathbf{M} \mathbf{P}_\alpha} - x \mathbf{G}),$$

where $\widetilde{\mathbf{M}}$ only depends on $(\mathbf{M}_i)_{i \in [d]}$ and $\mathbf{M}$. This operation essentially allows us to convert commitments to functions f and g into a commitment to the function $f \circ g$.

Now, let f be the transition function of $\mathcal{M}$. We represent this as a d -layer circuit consisting only of NAND gates.² Since f is efficiently computable we can assume that each layer has fewer than 2^λ gates. Let $\phi_{i,L}, \phi_{i,R} : \{0, 1\}^\lambda \rightarrow \{0, 1\}^\lambda$ be routing maps for layer i . Specifically, let g be the α -th gate at level i : Its left input will come from the $\phi_{i,L}(\alpha)$ -th gate at level $i - 1$, whereas its right input will come from the $\phi_{i,R}(\alpha)$ -th gate at level $i - 1$. We augment the hash key with commitments to truth tables of $\phi_{i,L}$ and $\phi_{i,R}$, which can be computed using EvalM.³

Next, we describe how to derive $\mathbf{M}_{\mathcal{M}^{(T)}(x)}$. For any $y \in \{0, 1\}^{\leq 2^\lambda}$, let us use X_y to denote the function that maps $\alpha \in \{0, 1\}^\lambda$ to $y[\alpha]$ if this is well defined (otherwise we map it to 0). We first compute a commitment to the truth table of X_x . This can be done efficiently given x . At this point, we will homomorphically evaluate $\mathcal{M}$ on this commitment. Consider the evaluation of $f(x)$ and let x_i be the binary string that describes the value of the gates at the i -th layer. Using the homomorphic read operation and the commitments to $\phi_{1,L}$ and $\phi_{1,R}$ in the hash key, we will derive commitments to the truth tables of $X_x \circ \phi_{1,L}$ and $X_x \circ \phi_{1,R}$. By homomorphically evaluating a NAND on these, we obtain a commitment to the truth table of

$$\text{NAND}(X_x \circ \phi_{1,L}, X_x \circ \phi_{1,R}) \equiv X_{x_1}.$$

We can then repeat and derive a commitment to the truth table of $\text{NAND}(X_{x_1} \circ \phi_{2,L}, X_{x_1} \circ \phi_{2,R})$, which coincides with X_{x_2} . Repeating this again and again for each layer, we obtain a commitment to the truth table of $X_{f(x)}$. At this point nothing prevents us to go further and derive a commitment to the truth table of $X_{f^{(T)}(x)} = X_{\mathcal{M}^{(T)}(x)}$. By applying some minor restructuring to this, we can efficiently derive the matrix $\mathbf{M}_{\mathcal{M}^{(T)}(x)}$.

This gives us almost what we want except for a tiny but fundamental detail: The noise magnitude in $\tilde{\mathbf{c}}_{\mathcal{M}^{(T)}(x)}$ would grow exponentially in T (due to the piling up of homomorphic hops). This would force us to choose a modulus q that is $\Omega(T)$ bits long, and hence the hash key (which corresponds to the sequential function input) would grow proportionally. Luckily, this problem can be solved using the bootstrapping technique of Hsieh, Lin and Luo [HLL23] at the cost of an additional circularity.

²In this overview, we ignore the fact that we need also COPY gates.

³These encodings do not hide $\mathcal{M}$, which is actually important for the correlation intractability proof but, to keep things simple, we ignore this aspect here.

2 Preliminaries

2.1 Notation

The security parameter is λ , all logarithms are base 2, and $k \geq \lambda$. Unless stated otherwise, all dimension parameters are polynomially bounded in λ . All vectors are columns. Unless an integer representative is displayed explicitly, matrix equalities are over $\mathbb{Z}_q$, where we assume for convenience that q is a power of 2. We write $[a] = \{0, \dots, a-1\}$, $x \xleftarrow{\$} X$ for sampling, and $X \approx_c Y$ and $X \approx_s Y$ for computational and statistical indistinguishability. For an integer matrix $\mathbf{C}$, put

$$\|\mathbf{C}\|_{\infty, \infty} = \max_i \sum_j |C_{i,j}|, \quad \|\mathbf{C}\|_{\infty} = \max_{i,j} |C_{i,j}|,$$

and use centered representatives when a modular matrix occurs inside a norm. Thus $\|\mathbf{CD}\|_{\infty, \infty} \leq \|\mathbf{C}\|_{\infty, \infty} \|\mathbf{D}\|_{\infty, \infty}$. We write $\|\mathbf{C}\|_2$ for the spectral norm. Set $n = k \log q$ and define

$$\mathbf{g}^\top = (1, 2, \dots, 2^{\log q - 1}), \quad \mathbf{G} = \mathbf{I}_k \otimes \mathbf{g}^\top \in \mathbb{Z}_q^{k \times n}.$$

The map $\mathbf{G}^{-1}$ decomposes the representative in $\{0, \dots, q-1\}$ of each entry into its canonical $\log q$ -bit representation and is applied columnwise to matrices. Hence $\mathbf{GG}^{-1}(\mathbf{Y}) = \mathbf{Y}$, for every matrix $\mathbf{Y}$ with k rows. We use $\text{vec}(\mathbf{C})$ for columnwise vectorization.

2.2 Lattice Trapdoors and Gaussians

Let $\mathcal{D}_\eta$ denote the discrete Gaussian on $\mathbb{Z}$ that assigns mass proportional to $\exp(-\pi x^2/\eta^2)$ to $x \in \mathbb{Z}$; a vector or matrix superscript denotes the corresponding independent product distribution. For every $\eta \geq 1$, a polynomial number of samples all have absolute value at most $\eta\sqrt{\lambda}$ except with probability $\delta_{\text{tail}} := 2^{-\Omega(\lambda)}$ [Ban93]. We recall the following standard flooding fact.

Lemma 2.1 (Gaussian Flooding). *Let $G \xleftarrow{\$} \mathcal{D}_\eta$ be independent of Δ , and suppose that $B > 0$, $\Delta \in \mathbb{Z}$, and $|\Delta| \leq B$. If $\eta/B = 2^{\omega(\log \lambda)}$, then $G \approx_s G + \Delta$. The same conclusion holds jointly with arbitrary side information S , provided that G is independent of the pair (Δ, S) .*

We use the standard lattice-trapdoor and preimage-sampling algorithms of [MP12].

Lemma 2.2 (Gaussian Sampling). *For polynomially bounded dimensions and $a \geq 6k \log q$, there are probabilistic polynomial-time algorithms with the following guarantees. The algorithm*

- $\text{TrapGen}(1^k, 1^a, q) \mapsto (\mathbf{A}, \mathbf{T}_A)$

outputs $\mathbf{A} \in \mathbb{Z}_q^{k \times a}$ whose marginal distribution is statistically close to uniform, together with a short basis $\mathbf{T}_A \in \mathbb{Z}^{a \times a}$ for its q -ary kernel. Except with negligible probability, $\|\mathbf{T}_A\|_2 = O(\sqrt{a} k \log q)$. Moreover, there are polynomial-time algorithms

- $\text{SampleLeft}(\mathbf{A}, \mathbf{T}_A, \mathbf{B}, \mathbf{Y}, \tau) \mapsto \mathbf{D} \in \mathbb{Z}^{(a+r) \times s},$
- $\text{SampleRight}(\mathbf{A}, \mathbf{T}_B, \mathbf{B}, \mathbf{Y}, \tau) \mapsto \mathbf{D} \in \mathbb{Z}^{(a+r) \times s}$

such that, for $\mathbf{A} \in \mathbb{Z}_q^{k \times a}$, $\mathbf{B} \in \mathbb{Z}_q^{k \times r}$, and $\mathbf{Y} \in \mathbb{Z}_q^{k \times s}$, if

$$\frac{\tau}{\|\mathbf{T}_A\|_2} = \omega(\sqrt{\log \lambda}), \quad \frac{\tau}{\sqrt{\|\mathbf{T}_B\|_2^2 + 1}} = \omega(\sqrt{\log \lambda}), \quad \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{T}_B \\ \mathbf{I}_n \end{bmatrix} = \mathbf{G} \pmod{q},$$

then their output $\mathbf{D}$ is statistically close to the parameter- τ discrete Gaussian over

$$\left\{ \mathbf{D} \in \mathbb{Z}^{(a+r) \times s} : [\mathbf{A} \ \mathbf{B}] \mathbf{D} = \mathbf{Y} \pmod{q} \right\}.$$

Except with negligible probability, $\|\mathbf{D}\|_\infty \leq \tau\sqrt{\lambda}$ and $\|\mathbf{D}\|_{\infty,\infty} \leq s\tau\sqrt{\lambda}$.

The following is a consequence of the leftover hash lemma.

Lemma 2.3 (Leftover Hashing). *For every polynomially bounded positive integer p , let $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{k \times a}$, $\mathbf{U} \xleftarrow{\$} \{0,1\}^{a \times p}$, and $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{k \times p}$ be independent. Then if $a \geq k \log q + 2\lambda + 2 \log p$ it holds that*

$$(\mathbf{A}, \mathbf{AU}) \approx_s (\mathbf{A}, \mathbf{B}).$$

2.3 Lattice Encodings and Decomposed LWE

Set $t = 2mn$, the decomposed sampler of [AMR26b], on input $(1^k, 1^m, 1^t, q, \sigma)$, samples

$$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{tk \times m}, \quad \mathbf{B} \xleftarrow{\$} \mathcal{D}_\sigma^{m \times tn}.$$

Writing $\mathbf{A}_i$ for the i -th block of k rows and $\mathbf{u}_i$ for the i -th standard basis vector of $\mathbb{Z}^t$, it sets

$$\mathbf{Z}_i = \mathbf{A}_i \mathbf{B} - \mathbf{u}_i^\top \otimes \mathbf{G} \in \mathbb{Z}_q^{k \times tn}, \quad \mathbf{Z} = [\mathbf{Z}_0 \mid \cdots \mid \mathbf{Z}_{t-1}] \in \mathbb{Z}_q^{k \times nt^2}.$$

We write

$$(\mathbf{A}, \mathbf{B}, \mathbf{Z}) \xleftarrow{\$} \text{DSample}(1^k, 1^m, 1^t, q, \sigma)$$

for this distribution and abbreviate the right-hand side by DSample when the parameters are clear. The lattice-encoding setup $\text{Setup}(1^\lambda)$ runs this sampler with the parameters fixed as above and returns its output as $\mathbf{pk}$.

We recall the following extension of the decomposed-LWE assumption of [AMR25a], which is shown to be equivalent to the ordinary decomposed LWE [AMR26b].

Definition 2.4 (Extended Decomposed LWE). *Let r be polynomially bounded and let $\eta_z, \eta_c > 0$ be parameter functions. Extended decomposed LWE with error widths (η_z, η_c) postulates that the following distribution ensembles are computationally indistinguishable:*

$$(\mathbf{A}, \mathbf{B}, \mathbf{Z}, \mathbf{Q}, \mathbf{Z}^\top \mathbf{s} + \mathbf{e}_z, \mathbf{Q}^\top \mathbf{s} + \mathbf{e}_c) \approx_c (\mathbf{A}, \mathbf{B}, \mathbf{Z}, \mathbf{Q}, \mathbf{v}_Z, \mathbf{v}_Q),$$

where $\mathbf{Q} \xleftarrow{\$} \mathbb{Z}_q^{k \times r}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^k$, $\mathbf{e}_z \xleftarrow{\$} \mathcal{D}_{\eta_z}^{nt^2}$, $\mathbf{e}_c \xleftarrow{\$} \mathcal{D}_{\eta_c}^r$, $\mathbf{v}_Z \xleftarrow{\$} \mathbb{Z}_q^{nt^2}$, $\mathbf{v}_Q \xleftarrow{\$} \mathbb{Z}_q^r$, and $(\mathbf{A}, \mathbf{B}, \mathbf{Z}) \xleftarrow{\$} \text{DSample}$.

For a function f , define $f^{(0)}$ to be the identity and $f^{(T+1)} = f \circ f^{(T)}$. We call an arithmetic-circuit family *bounded* if, on every input, the absolute value carried by every wire is bounded by a polynomial in the security parameter. We recall the following consequence of [AMR26c].

Theorem 2.5 (Iterated Key Equation). *Let $f : \{0,1\}^d \rightarrow \{0,1\}^d$ range over a fixed polynomial-size bounded arithmetic-circuit family of depth δ , and let $T \geq 1$. There are deterministic algorithms with syntax*

- $\text{ExpandP}(\mathbf{pk}, x) \mapsto \mathbf{P}_x \in \mathbb{Z}_q^{mn^4 \times n},$
- $\text{EvalM}(\mathbf{pk}, f, T) \mapsto \{\mathbf{M}_{f_i} \in \{0,1\}^{n \times mn^4}\}_{i \in [d]},$

- $\text{EvalH}(\text{pk}, f, x, 1^T) \mapsto \left(f^{(T)}(x), \{\mathbf{H}_{f_i, x} \in \mathbb{Z}_q^{nt^2 \times n}\}_{i \in [d]} \right)$.

For every pk in the support of $\text{Setup}(1^\lambda)$, every $x \in \{0, 1\}^d$, and every $i \in [d]$, these matrices satisfy

$$\mathbf{G}\mathbf{M}_{f_i} \mathbf{P}_x + \mathbf{Z}\mathbf{H}_{f_i, x} = f^{(T)}(x)_i \mathbf{G}. \quad (3)$$

Moreover, except with negligible probability over $\text{pk} \xleftarrow{\$} \text{Setup}(1^\lambda)$, the following bounds hold simultaneously for every $x \in \{0, 1\}^d$ and $i \in [d]$:

$$\|\mathbf{H}_{f_i, x}\|_{\infty, \infty} \leq (\lambda\sigma)^{O(d\delta + d\log(d+1)\log(T+1))}, \quad (4)$$

$$\|\mathbf{P}_x\|_{\infty, \infty} \leq t^d. \quad (5)$$

The algorithm ExpandP runs in $\text{poly}(\lambda, d)$ time, EvalM runs in $\text{poly}(\lambda, |f|, d, \log(T+1))$ time, and EvalH runs in $\text{poly}(\lambda, |f|, d, T)$ time.

3 Depth-Unbounded Attribute-Based Encryption

We construct ciphertext-policy attribute-based encryption for iterated space-bounded computations. A policy is a predicate $f : \{0, 1\}^d \rightarrow \{0, 1\}$ and we adopt the convention that $f(x) = 0$ authorizes x and $f(x) = 1$ rejects it. Throughout this section, we consider policies f computed by the T -fold iteration of a transition circuit C_f on a d -bit state, for a polynomially bounded T . We assume that the circuit depth of C_f is bounded by $\delta_d = O(\log d)$ and that the absolute value carried by every wire of the arithmetized transition circuit is bounded by a fixed polynomial in λ .

Note that in the notation of [Section 2.3](#), $f(x)$ corresponds to the first bit of $C_f^{(T)}(x)$, and we adopt this convention in favor of a lighter notation.

3.1 Syntax and Selective Security

The algorithms have syntax

- $\text{ABESetup}(1^\lambda, 1^d) \mapsto (\text{mpk}, \text{msk})$,
- $\text{ABEKeyGen}(\text{mpk}, \text{msk}, x) \mapsto \text{sk}_x$ with $x \in \{0, 1\}^d$,
- $\text{ABEEnc}(\text{mpk}, (f, j, T), \mu) \mapsto \text{ct}_f$ with $\mu \in \{0, 1\}^n$,
- $\text{ABEDec}(\text{mpk}, \text{sk}_x, \text{ct}_f) \mapsto \mu'$.

For every f as described above, every x with $f(x) = 0$, and every $\mu \in \{0, 1\}^n$, correctness requires

$$\Pr[\text{ABEDec}(\text{mpk}, \text{sk}_x, \text{ABEEnc}(\text{mpk}, f, \mu)) \neq \mu] \leq \text{negl}(\lambda),$$

where $(\text{mpk}, \text{msk}) \xleftarrow{\$} \text{ABESetup}(1^\lambda, 1^d)$ and $\text{sk}_x \xleftarrow{\$} \text{ABEKeyGen}(\text{mpk}, \text{msk}, x)$.

Definition 3.1 (Selective Security). *Let $\mathcal{A}$ be a PPT adversary. Before setup it outputs a policy f^* . The challenger samples*

$$(\text{mpk}, \text{msk}) \xleftarrow{\$} \text{ABESetup}(1^\lambda, 1^d)$$

and gives mpk to $\mathcal{A}$. Before and after the challenge, $\mathcal{A}$ may query the oracle

$$\mathcal{O}_{f^*}(x) = \begin{cases} \text{ABEKeyGen}(\text{mpk}, \text{msk}, x), & f^*(x) = 1, \\ \perp, & f^*(x) = 0. \end{cases} \quad (6)$$

The adversary submits $\mu_0, \mu_1 \in \{0, 1\}^n$. For $b \xleftarrow{\$} \{0, 1\}$ the challenger returns

$$\text{ct}^* \xleftarrow{\$} \text{ABEEnc}(\text{mpk}, f^*, \mu_b).$$

Let b' be the adversary's output. A scheme is selectively secure if, for any PPT adversary, there exists a negligible function negl such that

$$\left| \Pr[b' = b] - \frac{1}{2} \right| \leq \text{negl}(\lambda).$$

3.2 Parameters

Recall that $n = k \log q$ and $t = 2mn$. Choose $\sigma \geq 1$ so that $\log \sigma = \text{poly}(\lambda, d)$ independently of q .

Throughout this section set

$$B := (\lambda\sigma)^{\lceil \log(\lambda+1) \rceil^2 (d\delta_d + d \log(d+1) \lceil \log(\lambda+1) \rceil^2)}. \quad (7)$$

Choose a to be the least integer satisfying

$$a \geq \max \{6k \log q, k \log q + 2\lambda + 2 \log(mn^4) + 4\}, \quad (8)$$

and set

$$\eta_e := \lambda, \quad (9)$$

$$\tau := 2^{\lceil \log(\lambda+1) \rceil^2} \max \left\{ \sqrt{a} k \log q, \sqrt{(\sqrt{an} mn^4 t^d + nt^2 B)^2 + 1} \right\}, \quad (10)$$

$$\eta_B := a\eta_e \sqrt{\lambda} 2^{\lceil \log(\lambda+1) \rceil^2}. \quad (11)$$

Finally, choose the least q satisfying

$$\eta_e \sqrt{\lambda} + \tau \sqrt{\lambda} \left((a + nt^2) \eta_e \sqrt{\lambda} + n \left(mn^4 t^d \eta_B \sqrt{\lambda} + nt^2 B \eta_e \sqrt{\lambda} \right) \right) < q/4. \quad (12)$$

Such a choice has $\log q = \text{poly}(\lambda, d)$. Indeed, all dimensions and trapdoor bounds are polynomial in $\log q$, t^d has polynomial bit length, and B is independent of q . Consequently, the logarithm of the left-hand side of Equation (12) is at most

$$\text{poly}(\lambda, d) + \text{poly}(\lambda, d) \log \log q.$$

The same upper bound applies to the logarithms of the fixed encoding requirements by the preceding choice of parameter functions. Taking $\log q$ to be a sufficiently large polynomial in λ, d makes these bounds smaller than $\log q - 2$, so a suitable power of two exists. The minimal choice of a above is then polynomially bounded.

3.3 Construction

ABESetup($1^\lambda, 1^d$): Sample

$$\text{pk} \xleftarrow{\$} \text{Setup}(1^\lambda), \quad (\mathbf{A}, \mathbf{T}_A) \xleftarrow{\$} \text{TrapGen}(1^k, 1^a, q),$$

and, independently,

$$\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{k \times mn^4}, \quad \mathbf{V} \xleftarrow{\$} \mathbb{Z}_q^{k \times n}.$$

The encoding parameters pk contain $\mathbf{G} \in \mathbb{Z}_q^{k \times n}$ and $\mathbf{Z} \in \mathbb{Z}_q^{k \times nt^2}$. Output

$$\text{mpk} = (\text{pk}, \mathbf{A}, \mathbf{B}, \mathbf{V}), \quad \text{msk} = \mathbf{T}_A.$$

ABEKeyGen($\text{mpk}, \text{msk}, x$): Compute $\mathbf{P}_x \leftarrow \text{ExpandP}(\text{pk}, x) \in \mathbb{Z}_q^{mn^4 \times n}$ and sample

$$\mathbf{D}_x \xleftarrow{\$} \text{SampleLeft}(\mathbf{A}, \mathbf{T}_A, [\mathbf{Z} \mid \mathbf{B}\mathbf{P}_x], \mathbf{V}, \tau).$$

Then

$$\mathbf{D}_x \in \mathbb{Z}^{(a+nt^2+n) \times n}, \quad [\mathbf{A} \mid \mathbf{Z} \mid \mathbf{B}\mathbf{P}_x] \mathbf{D}_x = \mathbf{V} \pmod{q}. \quad (13)$$

Output $\text{sk}_x = (x, \mathbf{D}_x)$.

ABEEnc(mpk, f, μ): Let $\mathbf{M}_f = \text{EvalM}(\text{pk}, f, T) \in \{0, 1\}^{n \times mn^4}$. Sample independently

$$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^k, \quad \mathbf{e}_A \xleftarrow{\$} \mathcal{D}_{\eta_e}^a, \quad \mathbf{e}_Z \xleftarrow{\$} \mathcal{D}_{\eta_e}^{nt^2}, \quad \mathbf{e}_B \xleftarrow{\$} \mathcal{D}_{\eta_B}^{mn^4}, \quad \mathbf{e}_V \xleftarrow{\$} \mathcal{D}_{\eta_e}^n,$$

and set

$$\mathbf{c}_A = \mathbf{A}^\top \mathbf{s} + \mathbf{e}_A, \quad \mathbf{c}_Z = \mathbf{Z}^\top \mathbf{s} + \mathbf{e}_Z, \quad (14)$$

$$\mathbf{c}_B = (\mathbf{B} - \mathbf{G}\mathbf{M}_f)^\top \mathbf{s} + \mathbf{e}_B, \quad \mathbf{c}_V = \mathbf{V}^\top \mathbf{s} + \mathbf{e}_V + \frac{q}{2}\mu. \quad (15)$$

Output $\text{ct}_f = (f, \mathbf{c}_A, \mathbf{c}_Z, \mathbf{c}_B, \mathbf{c}_V)$.

ABEDec($\text{mpk}, (x, \mathbf{D}_x), \text{ct}_f$): Compute $\mathbf{P}_x = \text{ExpandP}(\text{pk}, x)$, and let $\text{EvalH}(\text{pk}, f, x, 1^T) = \mathbf{H}_{f,x} \in \mathbb{Z}_q^{nt^2 \times n}$. Then compute

$$\hat{\mathbf{c}}_x := \mathbf{P}_x^\top \mathbf{c}_B - \mathbf{H}_{f,x}^\top \mathbf{c}_Z \in \mathbb{Z}_q^n. \quad (16)$$

Reduce

$$\mathbf{c}_V - \mathbf{D}_x^\top \begin{bmatrix} \mathbf{c}_A \\ \mathbf{c}_Z \\ \hat{\mathbf{c}}_x \end{bmatrix} \in \mathbb{Z}_q^n \quad (17)$$

coordinatewise to the nearer of 0 and $q/2$ and output the corresponding bit vector.

The dimensions of every public matrix, secret-key matrix, and cryptographic ciphertext component are independent of T . The policy descriptor contributes $|f| + O(\log d + \log(T + 1))$ bits to the ciphertext. For a policy f ,

$\mathbf{M}_f$ and ABEEnc are computable in $\text{poly}(\lambda, d, |f|, \log T)$ time,

$\mathbf{H}_{f,x}$ and ABEDec are computable in $\text{poly}(\lambda, d, |f|, T)$ time.

3.4 Analysis

We show that the scheme satisfies correctness.

Theorem 3.2 (Correctness). *For every fixed policy ensemble satisfying the bounds above, every x such that $f(x) = 0$, and every $\mu \in \{0, 1\}^n$, the scheme decrypts correctly except with negligible probability.*

Proof. Condition on the overwhelming-probability events on which the correction bounds of [Theorem 2.5](#) and the trapdoor-quality bound of [Lemma 2.2](#) hold. On the latter event, [Equation \(10\)](#) meets the left-sampling threshold. On these events, [Equation \(5\)](#) and the choice of B give

$$\|\mathbf{P}_x^\top\|_{\infty, \infty} \leq mn^4 t^d, \quad \|\mathbf{H}_{f,x}^\top\|_{\infty, \infty} \leq nt^2 B.$$

Indeed, transposition changes each row sum into a column sum, which is at most the number of rows times the largest entry; every entry is in turn bounded by the corresponding row-sum norm in [Equations \(4\) and \(5\)](#). The scalar form of [Theorem 2.5](#) gives

$$\mathbf{G}\mathbf{M}_f\mathbf{P}_x + \mathbf{Z}\mathbf{H}_{f,x} = f(x)\mathbf{G} \pmod{q}.$$

Consequently,

$$\hat{\mathbf{c}}_x = (\mathbf{B}\mathbf{P}_x - f(x)\mathbf{G})^\top \mathbf{s} + \mathbf{P}_x^\top \mathbf{e}_B - \mathbf{H}_{f,x}^\top \mathbf{e}_Z \pmod{q}.$$

For $f(x) = 0$, [Equation \(13\)](#) therefore implies that the value in [Equation \(17\)](#) is congruent to

$$\mathbf{c}_V - \mathbf{D}_x^\top \begin{bmatrix} \mathbf{c}_A \\ \mathbf{c}_Z \\ \hat{\mathbf{c}}_x \end{bmatrix} = \frac{q}{2}\mu + \mathbf{e}_V - \mathbf{D}_x^\top \begin{bmatrix} \mathbf{e}_A \\ \mathbf{e}_Z \\ \mathbf{P}_x^\top \mathbf{e}_B - \mathbf{H}_{f,x}^\top \mathbf{e}_Z \end{bmatrix} \pmod{q}.$$

Except with negligible probability, every entry of $\mathbf{D}_x$ has magnitude at most $\tau\sqrt{\lambda}$, and every encryption-error coordinate has magnitude at most its Gaussian width times $\sqrt{\lambda}$. Hence

$$\left\| \mathbf{e}_V - \mathbf{D}_x^\top \begin{bmatrix} \mathbf{e}_A \\ \mathbf{e}_Z \\ \mathbf{P}_x^\top \mathbf{e}_B - \mathbf{H}_{f,x}^\top \mathbf{e}_Z \end{bmatrix} \right\|_\infty \leq \eta_e \sqrt{\lambda} + \tau \sqrt{\lambda} \left((a + nt^2) \eta_e \sqrt{\lambda} + n \left(mn^4 t^d \eta_B \sqrt{\lambda} + nt^2 B \eta_e \sqrt{\lambda} \right) \right) < q/4$$

by [Equation \(12\)](#). Rounding therefore recovers every coordinate of μ . Accounting for the complementary events proves the claim. $\square$

We prove that the scheme satisfies selective security.

Theorem 3.3 (Selective security). *Assume extended decomposed LWE for extension width $a + n$, with both error widths equal to η_e , at the parameters fixed above. Then the construction is selectively secure in the sense of [Definition 3.1](#).*

Proof. Fix an adversary. The reduction runs it until it receives the committed policy f^* . Consider the following series of hybrids.

Hybrid $\mathcal{H}_0$: This is the experiment of [Definition 3.1](#).

Hybrid $\mathcal{H}_1$: For every allowed query x , replace the output of `SampleLeft` by an ideal parameter- τ discrete Gaussian over

$$\left\{ \mathbf{D} \in \mathbb{Z}^{(a+nt^2+n) \times n} : [\mathbf{A} \mid \mathbf{Z} \mid \mathbf{B}\mathbf{P}_x] \mathbf{D} = \mathbf{V} \pmod{q} \right\}. \quad (18)$$

If this set is empty, the experiment aborts and assigns its success indicator an independent uniform bit; the same convention is used in every subsequent hybrid. This changes the experiment only negligibly because the marginal of $\mathbf{A}$ is statistically close to uniform and the probability that a uniform matrix has such a set empty is negligible. Indeed,

$$\Pr[\text{rank}_{\mathbb{F}_2}(\mathbf{A} \bmod 2) < k] \leq (2^k - 1)2^{-a} < 2^{k-a} \leq 2^{-2\lambda-4}.$$

Outside this event, some $k \times k$ minor has odd determinant and is therefore invertible modulo q . Thus $\mathbf{A}$ is surjective over $\mathbb{Z}_q$, and every coset in Equation (18) is nonempty.

To conclude, Equation (10) meets the left-sampling threshold of Lemma 2.2 with overwhelming probability. A hybrid over the polynomial number of adaptive queries shows that the two key distributions are statistically close. Conditioned on the public matrices, the key oracle is now independent of $\mathbf{T}_A$, although it need not be efficient.

Hybrid $\mathcal{H}_2$: Discard $\mathbf{T}_A$ and replace $\mathbf{A}$ by a uniform matrix in $\mathbb{Z}_q^{k \times a}$. The change is statistically negligible by the near-uniform marginal guarantee in Lemma 2.2.

Hybrid $\mathcal{H}_3$: Let $\mathbf{M}_{f^*}$ be the j^* -th matrix returned by `EvalM(pk,  $f^*$ ,  $T^*$ )`. Sample $\mathbf{U} \xleftarrow{\$} \{0, 1\}^{a \times p}$ and set

$$\mathbf{B} = \mathbf{A}\mathbf{U} + \mathbf{G}\mathbf{M}_{f^*} \pmod{q}. \quad (19)$$

Condition on (pk, f^*) , so $\mathbf{G}\mathbf{M}_{f^*}$ is a fixed shift independent of $\mathbf{A}$. By Lemma 2.3 and Equation (8),

$$(\mathbf{A}, \mathbf{A}\mathbf{U} + \mathbf{G}\mathbf{M}_{f^*}) \approx_s (\mathbf{A}, \mathbf{B}).$$

Hence the entire adaptive view changes by a negligible amount.

Hybrid $\mathcal{H}_4$: For every allowed query x , compute $\mathbf{P}_x \leftarrow \text{ExpandP}(\text{pk}, x)$ and let $\mathbf{H}_{f^*, x}$ be the j^* -th correction matrix returned by `EvalH(pk,  $f^*$ ,  $x$ ,  $1^{T^*}$ )`. Set

$$\overline{\mathbf{R}}_x = \begin{bmatrix} -\mathbf{U}\mathbf{P}_x \\ \mathbf{H}_{f^*, x} \end{bmatrix} \in \mathbb{Z}^{(a+nt^2) \times n}. \quad (20)$$

Since every allowed query satisfies $f^*(x) = 1$,

$$\begin{aligned} [\mathbf{A} \mid \mathbf{Z} \mid \mathbf{B}\mathbf{P}_x] \begin{bmatrix} \overline{\mathbf{R}}_x \\ \mathbf{I}_n \end{bmatrix} &= -\mathbf{A}\mathbf{U}\mathbf{P}_x + \mathbf{Z}\mathbf{H}_{f^*, x} + (\mathbf{A}\mathbf{U} + \mathbf{G}\mathbf{M}_{f^*})\mathbf{P}_x \\ &= \mathbf{G}\mathbf{M}_{f^*}\mathbf{P}_x + \mathbf{Z}\mathbf{H}_{f^*, x} = \mathbf{G} \pmod{q}. \end{aligned} \quad (21)$$

Thus $\overline{\mathbf{R}}_x$ is a standard gadget trapdoor for $[\mathbf{A} \mid \mathbf{Z} \mid \mathbf{B}\mathbf{P}_x]$. Moreover,

$$\|\overline{\mathbf{R}}_x\|_2 \leq \sqrt{an} mn^4 t^d + nt B. \quad (22)$$

Indeed, every entry of $\mathbf{U}\mathbf{P}_x$ has magnitude at most $mn^4 t^d$, every entry of $\mathbf{H}_{f^*, x}$ has magnitude at most B , and the spectral norm is at most the Frobenius norm. This bound holds on the overwhelming-probability setup event of Theorem 2.5; the complementary event contributes at most its negligible probability to the statistical distance below.

Answer the query with $(x, \mathbf{D}_x)$, where

$$\mathbf{D}_x \xleftarrow{\$} \text{SampleRight}([\mathbf{A} \mid \mathbf{Z}], \bar{\mathbf{R}}_x, \mathbf{B}\mathbf{P}_x, \mathbf{V}, \tau).$$

Equations (10) and (22) meet the right-trapdoor threshold. By Lemma 2.2, its output is statistically close to the ideal Gaussian in $\mathcal{H}_3$. A hybrid over the adaptive queries has negligible cost.

Hybrid $\mathcal{H}_5$: For the challenge policy, Equation (19) gives $\mathbf{B} - \mathbf{G}\mathbf{M}_{f^*} = \mathbf{A}\mathbf{U}$. Replace $\mathbf{c}_B = (\mathbf{A}\mathbf{U})^\top \mathbf{s} + \mathbf{e}_B$ by

$$\mathbf{c}_B = \mathbf{U}^\top \mathbf{c}_A + \mathbf{e}_B, \quad \mathbf{e}_B \xleftarrow{\$} \mathcal{D}_{\eta_B}^{mn^4}, \quad (23)$$

where $\mathbf{e}_B$ is sampled afresh and independently. The Gaussian tail bound gives, except with negligible probability,

$$\|\mathbf{U}^\top \mathbf{e}_A\|_\infty \leq a\eta_e \sqrt{\lambda}.$$

Condition on any values of $\mathbf{U}$ and $\mathbf{e}_A$ satisfying this bound. The ratio between η_B and the shift is at least $2^{\lceil \log(\lambda+1) \rceil^2}$ by Equation (11). Applying Lemma 2.1 coordinatewise, taking a hybrid over the p coordinates, and then averaging over $\mathbf{U}$ and $\mathbf{e}_A$ shows that replacing $\mathbf{e}_B$ by $\mathbf{e}_B + \mathbf{U}^\top \mathbf{e}_A$ is statistically indistinguishable jointly with $\mathbf{U}$ and $\mathbf{e}_A$. Thus $\mathcal{H}_4$ and $\mathcal{H}_5$ are statistically indistinguishable.

Hybrid $\mathcal{H}_6$: We sample the ciphertext components uniformly at random. To argue indistinguishability, we apply extended decomposed LWE with extension matrix $[\mathbf{A} \mid \mathbf{V}] \in \mathbb{Z}_q^{k \times (a+n)}$. In $\mathcal{H}_5$,

$$\left(\mathbf{c}_Z, \begin{bmatrix} \mathbf{c}_A \\ \mathbf{c}_V - \frac{q}{2}\mu_b \end{bmatrix} \right) = \left(\mathbf{Z}^\top \mathbf{s} + \mathbf{e}_Z, \begin{bmatrix} \mathbf{A}^\top \mathbf{s} + \mathbf{e}_A \\ \mathbf{V}^\top \mathbf{s} + \mathbf{e}_V \end{bmatrix} \right) \quad (24)$$

is exactly the real distribution of Definition 2.4 with errors (η_e, η_e) . The reduction samples $\mathbf{U}$, programs $\mathbf{B}$ as in Equation (19), answers every key query using the bottom-identity trapdoor in Equation (20), and computes $\mathbf{c}_B$ by Equation (23). None of these operations uses the LWE secret.

Replace Equation (24) by independent uniform vectors. Then $\mathbf{c}_V$ is uniform after adding $q\mu_b/2$. The component $\mathbf{c}_B$ may remain correlated with the uniform $\mathbf{c}_A$, but its distribution is independent of b . Consequently the complete challenge view is independent of b .

The first five transitions have negligible statistical cost, and the last transition is computationally indistinguishable under the stated extended decomposed-LWE assumption. The advantage in $\mathcal{H}_6$ is zero, which proves the theorem. $\square$

3.5 Laconic Function Evaluation

The above construction can also be turned into an attribute-based laconic function evaluation (AB-LFE), as in [QWW18], which we sketch in the following.

The AB-LFE setup samples $\mathbf{pk} \xleftarrow{\$} \text{Setup}(1^\lambda)$, and its hash algorithm computes $\mathbf{M}_f$. To encrypt $\mu \in \{0, 1\}^n$ under x for an output $b \in \{0, 1\}$, compute $\mathbf{P}_x \leftarrow \text{ExpandP}(\mathbf{pk}, x)$, sample

$$\mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{k \times n}, \quad \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^k, \quad \mathbf{e} \xleftarrow{\$} \mathcal{D}_{\eta_z}^{nt^2}, \quad \hat{\mathbf{e}} \xleftarrow{\$} \mathcal{D}_{\eta_c}^n,$$

and return $\text{ct} = (x, b, \mathbf{U}, \mathbf{z}, \mathbf{c})$, where

$$\begin{aligned}\mathbf{z}^\top &= \mathbf{s}^\top \mathbf{Z} + \mathbf{e}^\top, \\ \mathbf{c}^\top &= \mathbf{s}^\top \mathbf{G}(\mathbf{M}_f \mathbf{P}_x - b \mathbf{I}_n) \mathbf{G}^{-1}(\mathbf{U}) + \hat{\mathbf{e}}^\top + \frac{q}{2} \mu^\top.\end{aligned}$$

Decryption computes $\mathbf{H}_{f,x}$ and rounds each coordinate of $\mathbf{c}^\top + \mathbf{z}^\top \mathbf{H}_{f,x} \mathbf{G}^{-1}(\mathbf{U})$ to the nearer of 0 and $q/2$. By Equation (3), this row vector equals

$$(f(x) - b) \mathbf{s}^\top \mathbf{U} + \frac{q}{2} \mu^\top + \hat{\mathbf{e}}^\top + \mathbf{e}^\top \mathbf{H}_{f,x} \mathbf{G}^{-1}(\mathbf{U}).$$

When $f(x) = b$, the first term vanishes so the decryption correctly recovers μ by rounding, since the other components are bounded. For security, if $f(x) \neq b$, then the first term makes the ciphertext pseudorandom, assuming the hardness of the extended decomposed LWE problem.

Finally, Theorem 2.5 gives compact setup, hashing, and encryption, while decryption runs in time polynomial in T .

4 Delay Encryption

For every security parameter, fix a polynomially bounded integer d , a public transition map $f : \{0, 1\}^d \rightarrow \{0, 1\}^d$ and an efficiently samplable distribution $\mathcal{D}$ over $\{0, 1\}^d$. The integer T is encoded in binary, and the public description satisfies

$$d + |f| \leq \text{poly}(\lambda, \log(T + 1)). \quad (25)$$

The transition map has a polynomial-size arithmetic circuit of depth $\delta = O(\log(d + 1))$, and every integer wire value on a Boolean input has polynomially bounded magnitude. We recall the definition of sequentiality.

Definition 4.1 (Sequentiality). *Let $T : \mathbb{N} \rightarrow \mathbb{N}$, the family f is T -sequential with respect to $\mathcal{D}$ if, for every pair of PPT algorithms $\mathcal{A} = (\mathcal{A}_{\text{pre}}, \mathcal{A}_{\text{on}})$ whose online running time is at most T , there exists a negligible function negl such that*

$$\Pr \left[\mathcal{A}_{\text{on}}(\text{st}, \alpha) = f^{(T)}(\alpha) : \begin{array}{l} \text{st} \xleftarrow{\$} \mathcal{A}_{\text{pre}}(1^\lambda, f, T), \\ \alpha \xleftarrow{\$} \mathcal{D} \end{array} \right] \leq \text{negl}(\lambda).$$

4.1 Garbled Circuits

We use a garbling scheme [Yao86, BHR12] with the following syntax:

- $\text{Garble}(1^\lambda, C, \{L_{i,b}\}_{i \in [d], b \in \{0,1\}}) \mapsto \tilde{C}$
- $\text{GEval}(\tilde{C}, \{L_{i,x_i}\}_{i \in [d]}) \mapsto y$

where all labels are independent and uniform in $\{0, 1\}^n$. Correctness requires

$$\Pr \left[\text{GEval}(\tilde{C}, \{L_{i,x_i}\}_{i \in [d]}) = C(x) \right] \geq 1 - \text{negl}(\lambda).$$

Definition 4.2 (Simulatability). *There exists a PPT simulator $\mathcal{S}$ such that for any circuit C and any input $x \in \{0, 1\}^d$ we have that*

$$(\tilde{C}, \{L_{i,x_i}\}_{i \in [d]}) \approx_c \mathcal{S}(1^\lambda, C(x), \{L_{i,x_i}\}_{i \in [d]})$$

where $L_{i,b} \xleftarrow{\$} \{0, 1\}^n$ and $\tilde{C} \xleftarrow{\$} \text{Garble}(1^\lambda, C, \{L_{i,b}\}_{i \in [d], b \in \{0,1\}})$.

4.2 Definitions

We recall the syntax of a delay encryption scheme. A scheme for d -bit messages consists of algorithms with the following syntax

- $\text{DESetup}(1^\lambda, 1^d, f, T) \mapsto \text{mpk}$,
- $\text{DEEnc}(\text{mpk}, \alpha, \mu) \mapsto \text{ct}$,
- $\text{DEExtract}(\text{mpk}, \alpha) \mapsto \text{dk}_\alpha$,
- $\text{DEDec}(\text{mpk}, \text{dk}_\alpha, \text{ct}) \mapsto \mu' \text{ or } \perp$,

where $\alpha, \mu \in \{0, 1\}^d$. Correctness requires that, for every α, μ ,

$$\Pr \left[\text{DEDec}(\text{mpk}, \text{dk}_\alpha, \text{ct}) = \mu : \begin{array}{l} \text{mpk} \xleftarrow{\$} \text{DESetup}(1^\lambda, 1^d, f, T), \\ \text{dk}_\alpha \xleftarrow{\$} \text{DEExtract}(\text{mpk}, \alpha), \\ \text{ct} \xleftarrow{\$} \text{DEEnc}(\text{mpk}, \alpha, \mu) \end{array} \right] \geq 1 - \text{negl}(\lambda).$$

Setup and encryption are compact if they run in $\text{poly}(\lambda, d, |f|, \log(T+1))$ time. Extraction may run in $\text{poly}(\lambda, d, |f|, T)$ time, and decryption is polynomial in its explicit inputs.

Next, we define the notion of one-wayness for delay encryption, modeling the fact that it should be hard to invert an encryption of a uniformly sampled message in time less than T . Note that the stronger notion of indistinguishability can be generically achieved by combining a scheme satisfying the definition below with the Goldreich-Levin theorem [GL89], as shown in [AMZ24].

Definition 4.3 (One-Wayness). *For a PPT adversary $\mathcal{A} = (\mathcal{A}_{\text{pre}}, \mathcal{A}_{\text{on}})$, sample*

$$\begin{array}{ll} \text{mpk} \xleftarrow{\$} \text{DESetup}(1^\lambda, 1^d, f, T), & \text{st} \leftarrow \mathcal{A}_{\text{pre}}(\text{mpk}), \\ \alpha \xleftarrow{\$} \mathcal{D}, & \mu \xleftarrow{\$} \{0, 1\}^d, \\ \text{ct} \xleftarrow{\$} \text{DEEnc}(\text{mpk}, \alpha, \mu), & \mu' \leftarrow \mathcal{A}_{\text{on}}(\text{st}, \alpha, \text{ct}). \end{array}$$

A delay encryption scheme is T -one-way with respect to $\mathcal{D}$ if for every adversary with online running time $\mathcal{A}_{\text{on}}$ bounded by T there exists a negligible function negl such that

$$\Pr[\mu' = \mu] \leq \text{negl}(\lambda).$$

4.3 Construction

Choose $\sigma \geq 1$ so that $\log \sigma = \text{poly}(\lambda, d, \log(T+1))$, independently of q . Let $B_H \geq 1$ be a deterministic bound furnished by Equation (4), chosen so that $\log B_H = \text{poly}(\lambda, d, \log(T+1))$ and, except with negligible probability over $\text{pk} \xleftarrow{\$} \text{Setup}(1^\lambda)$,

$$\|\mathbf{H}_{f_i, \alpha}\|_{\infty, \infty} \leq B_H \quad \text{for every } \alpha \in \{0, 1\}^d \text{ and } i \in [d]. \quad (26)$$

Set

$$\eta_z := \lambda, \quad B_{\text{dec}} := nt^2 \eta_z \sqrt{\lambda} B_H, \quad \eta_c := B_{\text{dec}} 2^{\lceil (\log(\lambda+1))^2 \rceil}. \quad (27)$$

Choose the power-of-two modulus q , subject to the lattice-encoding requirements of Section 2, so that

$$\eta_c \sqrt{\lambda} + B_{\text{dec}} < q/4. \quad (28)$$

This choice preserves compactness. Indeed, for $L = \log q$, all factors in the left-hand side of Equation (28) other than n and t have logarithm polynomial in $\lambda, d, \log(T+1)$, independently of L , while n and t are polynomial in L . Thus a sufficiently large $L = \text{poly}(\lambda, d, \log(T+1))$ satisfies Equation (28). The extended decomposed-LWE instance used below has extension width n and error widths (η_z, η_c) .

Our construction assumes a simulatable garbling scheme satisfying Definition 4.2.

DESetup($1^\lambda, 1^d, f, T$): Sample

$$\text{pk} \xleftarrow{\$} \text{Setup}(1^\lambda), \quad \text{EvalM}(\text{pk}, f, T) = \{\mathbf{M}_{f_i}\}_{i \in [d]}.$$

Return $\text{mpk} = (\text{pk}, f, T, \{\mathbf{M}_{f_i}\}_{i \in [d]})$.

DEEnc(mpk, α, μ): Compute $\mathbf{P}_\alpha = \text{ExpandP}(\text{pk}, \alpha)$, and sample mutually independent labels $L_{i,b} \xleftarrow{\$} \{0, 1\}^n$ for every $i \in [d]$ and $b \in \{0, 1\}$. Independently for every such pair, sample

$$\mathbf{U}_{i,b} \xleftarrow{\$} \mathbb{Z}_q^{k \times n}, \quad \mathbf{s}_{i,b} \xleftarrow{\$} \mathbb{Z}_q^k, \quad \mathbf{e}_{i,b} \xleftarrow{\$} \mathcal{D}_{\eta_z}^{nt^2}, \quad \hat{\mathbf{e}}_{i,b} \xleftarrow{\$} \mathcal{D}_{\eta_c}^n,$$

and set

$$\begin{aligned} \mathbf{z}_{i,b}^\top &:= \mathbf{s}_{i,b}^\top \mathbf{Z} + \mathbf{e}_{i,b}^\top, \\ \mathbf{c}_{i,b}^\top &:= \mathbf{s}_{i,b}^\top \mathbf{G}(\mathbf{M}_{f_i} \mathbf{P}_\alpha - b \mathbf{I}_n) \mathbf{G}^{-1}(\mathbf{U}_{i,b}) + \hat{\mathbf{e}}_{i,b}^\top + \frac{q}{2} L_{i,b}^\top, \\ \text{ct}_{i,b} &:= (\mathbf{U}_{i,b}, \mathbf{z}_{i,b}^\top, \mathbf{c}_{i,b}^\top). \end{aligned}$$

Let C_μ be the circuit that outputs μ on any input (with size padded appropriately). Compute

$$\tilde{C} \xleftarrow{\$} \text{Garble}(1^\lambda, C_\mu, \{L_{i,b}\}_{i \in [d], b \in \{0,1\}})$$

and return $\text{ct} = (\tilde{C}, \{\text{ct}_{i,b}\}_{i \in [d], b \in \{0,1\}})$.

DEExtract(mpk, α): Parse $\text{mpk} = (\text{pk}, f, T, \{\mathbf{M}_{f_i}\}_{i \in [d]})$, compute

$$\text{dk}_\alpha := \text{EvalH}(\text{pk}, f, \alpha, 1^T) = \left(f^{(T)}(\alpha), \{\mathbf{H}_{f_i, \alpha}\}_{i \in [d]} \right),$$

and return dk_α .

DEDec($\text{mpk}, \text{dk}_\alpha, \text{ct}$): For every $i \in [d]$, set $b = f^{(T)}(\alpha)_i$ and compute

$$\mathbf{c}_{i,b}^\top + \mathbf{z}_{i,b}^\top \mathbf{H}_{f_i, \alpha} \mathbf{G}^{-1}(\mathbf{U}_{i,b})$$

and round to the nearest multiple of $q/2$, obtaining $L_{i,b}$. Return $\text{GEval}(\tilde{C}, \{L_{i, (f^{(T)}(\alpha))_i}\}_{i \in [d]})$.

For $T = \text{poly}(\lambda)$, Theorem 2.5 shows that **DESetup** and **DEEnc** run in $\text{poly}(\lambda, d, |f|, \log(T+1))$ time, **DEExtract** runs in $\text{poly}(\lambda, d, |f|, T)$ time, and **DEDec** is polynomial in its explicit inputs. Indeed, setup invokes **EvalM**, encryption invokes **ExpandP**, and only extraction invokes **EvalH**; f, T are the fixed public parameters specified at the beginning of the section.

4.4 Analysis

We show that the scheme satisfies correctness.

Theorem 4.4 (Correctness). *For $T = \text{poly}(\lambda)$, the delay encryption scheme with the parameters of [Equations \(27\)](#) and [\(28\)](#) is correct.*

Proof. Fix $\alpha, \mu \in \{0, 1\}^d$ and $i \in [d]$, and set $b = f^{(T)}(\alpha)_i$. By [Equation \(3\)](#),

$$\mathbf{G}(\mathbf{M}_{f_i} \mathbf{P}_\alpha - b \mathbf{I}_n) + \mathbf{Z} \mathbf{H}_{f_i, \alpha} = \mathbf{0}.$$

Consequently, the row vector rounded by DEDec is

$$\mathbf{c}_{i,b}^\top + \mathbf{z}_{i,b}^\top \mathbf{H}_{f_i, \alpha} \mathbf{G}^{-1}(\mathbf{U}_{i,b}) = \frac{q}{2} L_{i,b}^\top + \hat{\mathbf{e}}_{i,b}^\top + \mathbf{e}_{i,b}^\top \mathbf{H}_{f_i, \alpha} \mathbf{G}^{-1}(\mathbf{U}_{i,b}).$$

Except with negligible probability, [Equation \(26\)](#) holds and all coordinates of $\mathbf{e}_{i,b}$ and $\hat{\mathbf{e}}_{i,b}$ have absolute value at most $\eta_z \sqrt{\lambda}$ and $\eta_c \sqrt{\lambda}$, respectively. Since every entry of $\mathbf{G}^{-1}(\mathbf{U}_{i,b})$ is a bit, each coordinate satisfies

$$\left\| \mathbf{e}_{i,b}^\top \mathbf{H}_{f_i, \alpha} \mathbf{G}^{-1}(\mathbf{U}_{i,b}) \right\|_\infty \leq nt^2 \eta_z \sqrt{\lambda} B_H = B_{\text{dec}}.$$

Thus the total error has absolute value less than $q/4$ in every coordinate by [Equation \(28\)](#), so decryption recovers $L_{i,b}$. A union bound over $i \in [d]$, followed by garbling correctness, gives

$$\text{GEval}\left(\tilde{C}, \{L_{i, (f^{(T)}(\alpha))_i}\}_{i \in [d]}\right) = C_\mu(f^{(T)}(\alpha)) = \mu$$

except with negligible probability. □

Before proving one-wayness, we show a useful helping lemma.

Lemma 4.5 (Pseudorandomness). *Assuming extended decomposed LWE with extension width n and error widths (η_z, η_c) , for all $i \in [d]$ and $b \neq f^{(T)}(\alpha)_i$, we have*

$$(\xi, \mathbf{c}_{i,b}) \approx_c (\xi, \mathbf{R})$$

where $\mathbf{R}$ is uniformly sampled and ξ contains the public parameters and the rest of the ciphertext.

Proof. The key equation gives

$$\mathbf{G}(\mathbf{M}_{f_i} \mathbf{P}_\alpha - b \mathbf{I}_n) = -\mathbf{Z} \mathbf{H}_{f_i, \alpha} + (f^{(T)}(\alpha)_i - b) \mathbf{G}. \quad (29)$$

We prove pseudorandomness with a reduction to extended decomposed LWE. The reduction interprets the public prefix $(\mathbf{A}, \mathbf{B}, \mathbf{Z})$ of its extended decomposed-LWE challenge as $\mathbf{pk}$, $\mathbf{Q}$ as the extension matrix and $\mathbf{v}_Z, \mathbf{v}_Q$ as the two challenge vectors. The reduction returns ξ as computed in the scheme together with

$$\left((f^{(T)}(\alpha)_i - b) \mathbf{Q}, \mathbf{v}_Z^\top, -\mathbf{v}_Z^\top \mathbf{H}_{f_i, \alpha} \mathbf{G}^{-1} \left((f^{(T)}(\alpha)_i - b) \mathbf{Q} \right) + \mathbf{v}_Q^\top + \frac{q}{2} L_{i,b}^\top \right). \quad (30)$$

Because $(f^{(T)}(\alpha))_i - b \in \{-1, 1\}$, the first component is uniform in $\mathbb{Z}_q^{k \times n}$. In the real extended decomposed-LWE distribution, rename the challenge secret $\mathbf{s}_{i,b}$. Since

$$\mathbf{G} \mathbf{G}^{-1} \left((f^{(T)}(\alpha)_i - b) \mathbf{Q} \right) = (f^{(T)}(\alpha)_i - b) \mathbf{Q} \quad \text{and} \quad (f^{(T)}(\alpha)_i - b)^2 = 1,$$

Equation (29) shows that the third component of Equation (30) is exactly

$$\mathbf{s}_{i,b}^\top \mathbf{G}(\mathbf{M}_{f_i} \mathbf{P}_\alpha - b \mathbf{I}_n) \mathbf{G}^{-1} \left(\left(f^{(T)}(\alpha)_i - b \right) \mathbf{Q} \right) + \mathbf{e}_c^\top + \frac{q}{2} L_{i,b}^\top - \mathbf{e}_z^\top \mathbf{H}_{f_i, \alpha} \mathbf{G}^{-1} \left(\left(f^{(T)}(\alpha)_i - b \right) \mathbf{Q} \right).$$

Condition on the simultaneous event on which Equation (26) holds and every coordinate of $\mathbf{e}_z$ has absolute value at most $\eta_z \sqrt{\lambda}$, which happens with overwhelming probability. On this event, the last row vector has infinity norm at most B_{dec} , by the same calculation as in the correctness proof. Since

$$\frac{\eta_c}{B_{\text{dec}}} = 2^{\lceil (\log(\lambda+1))^2 \rceil} = 2^{\omega(\log \lambda)},$$

by Lemma 2.1 the real distribution of Equation (30) is statistically indistinguishable from the actual ciphertext produced by the honest algorithm, jointly with ξ .

In the ideal extended decomposed-LWE distribution, $\mathbf{v}_Q$ is uniform independently of $\mathbf{Q}, \mathbf{v}_Z, \xi$. Conditioned on the first two components of Equation (30) and on ξ , its third component is therefore uniform and independent of $L_{i,b}$. This completes the proof. $\square$

Finally, we are ready to prove that the scheme is one-way.

Theorem 4.6 (One-Wayness). *Assuming extended decomposed LWE with extension width n and error widths (η_z, η_c) , the simulatable garbling property of Definition 4.2, and T -sequentiality of f , the scheme is one-way.*

Proof. Consider the following hybrids.

Hybrid $\mathcal{H}_0$: This is the experiment of Definition 4.3.

Hybrid $\mathcal{H}_1$: Sample $\alpha \xleftarrow{\$} \mathcal{D}$ and $\mu \xleftarrow{\$} \{0, 1\}^d$, and encrypt $\mu \oplus f^{(T)}(\alpha)$; the adversary succeeds if it outputs that value. Conditioned on α , the encrypted message remains uniform, so this hybrid is identically distributed to $\mathcal{H}_0$.

Hybrid $\mathcal{H}_2$: For every $i \in [d]$, replace $\text{ct}_{i, 1-f^{(T)}(\alpha)_i}$ by a uniformly-sampled component. A hybrid over $i \in [d]$ along with Lemma 4.5 proves computational indistinguishability.

Hybrid $\mathcal{H}_3$: Replace the garbling of $C_{\mu \oplus f^{(T)}(\alpha)}$ by a garbling of the circuit that on input x outputs $\mu \oplus x$, padding as appropriate. On the selected input $f^{(T)}(\alpha)$, both circuits output $\mu \oplus f^{(T)}(\alpha)$. By Definition 4.2, their garblings and selected labels are computationally indistinguishable.

Hybrid $\mathcal{H}_4$: Restore every uniformly-sampled ciphertext to an encryption of its label. The garbling and the label may again be included as auxiliary information in Lemma 4.5. A hybrid over $i \in [d]$ proves computational indistinguishability from $\mathcal{H}_3$.

Note that the final hybrid can be sampled without computing $f^{(T)}(\alpha)$. Construct an algorithm for Definition 4.1 as follows. During preprocessing, sample mpk , run $\mathcal{A}_{\text{pre}}(\text{mpk})$, and store (mpk, st) . Online, on input α , sample a random μ , generate the hybrid $\mathcal{H}_4$ ciphertext by garbling $x \mapsto \mu \oplus x$, run $\mathcal{A}_{\text{on}}(\text{st}, \alpha, \text{ct})$ to obtain μ' , and output $\mu \oplus \mu'$. Its online running time is at most T by the hypothesis, and

$$\mu' = \mu \oplus f^{(T)}(\alpha) \iff \mu \oplus \mu' = f^{(T)}(\alpha).$$

Sequentiality therefore makes the success probability in $\mathcal{H}_4$ negligible. The polynomial number of negligible hybrid transitions proves the claim. $\square$

5 A Sequential Function from Circular Decomposed LWE

5.1 More Background on Lattice Encodings

We recall some additional facts on lattice encodings from [AMR26a, AMR26c] that we only need below.

A length- d matrix-output read-once branching program $\text{BP} = \{\mathbf{M}_{i,b}\}_{i,b}$ on input $\alpha \in \{0,1\}^d$ computes $\text{BP}(\alpha) = \prod_{i \in [d]} \mathbf{M}_{i,\alpha_i}$. In [AMR26a] it is shown that one can efficiently compute $\mathbf{M}_{\text{BP}}$ and, for every α , matrices $\mathbf{H}_{\text{BP},\alpha}$ and $\mathbf{P}_\alpha$ such that

$$\mathbf{GCan}(\mathbf{M}_{\text{BP}})(\alpha) + \mathbf{Z}\mathbf{H}_{\text{BP},\alpha} = \mathbf{GBP}(\alpha), \quad (31)$$

with $\|\mathbf{H}_{\text{BP},\alpha}^\top\|_{\infty,\infty} \leq 2^{d \cdot \text{polylog}(\lambda)}$ and $\|\mathbf{P}_\alpha^\top\|_{\infty,\infty} \leq mn^4 t^d$. Here $\text{Can}(\mathbf{M}_{\text{BP}})(\alpha) := \mathbf{M}_{\text{BP}}\mathbf{P}_\alpha$ is the *canonical representation*, which is itself a read-once matrix branching program.

Given matrix-output programs $\{\text{BP}_i\}_{i \in [a]}$, for polynomially bounded a , a function $h : \{0,1\}^d \times \mathbb{Z}_q^a \rightarrow \mathbb{Z}_q$, $\alpha \in \{0,1\}^d$, and $\beta \in \mathbb{Z}_q^a$, the deterministic algorithms

- $\text{BPEvalM}(\text{pk}, h, \{\text{BP}_i\}_{i \in [a]}) \mapsto \mathbf{M}_h$,
- $\text{BPEvalH}(\text{pk}, h, \alpha, \beta, \{\text{BP}_i\}_{i \in [a]}) \mapsto \{\mathbf{H}_{h,\alpha,\beta}^{(i)} \in \mathbb{Z}_q^{n \times n}\}_{i \in [a]}, \mathbf{H}'_{h,\alpha,\beta} \in \mathbb{Z}_q^{nt^2 \times n}$

compute a public $\mathbf{M}_h$, independent of α, β , and public coefficient matrices satisfying [AMR26a]

$$\mathbf{GCan}(\mathbf{M}_h)(\alpha) - h(\alpha, \beta)\mathbf{G} = \sum_{i \in [a]} (\mathbf{GBP}_i(\alpha) - \beta_i \mathbf{G}) \mathbf{H}_{h,\alpha,\beta}^{(i)} + \mathbf{Z}\mathbf{H}'_{h,\alpha,\beta}. \quad (32)$$

When h is a unary or binary arithmetic gate, $a \leq 2$, and the transpose norms of the matrices are bounded by $2^{d \cdot \text{polylog}(\lambda)}$ times the largest absolute value among the gate's inputs, output, and constants. For the circuits considered in this Section, the norms of all matrices are at most $2^{d \cdot \text{polylog}(\lambda)}$.

Finally, let $f = (f_0, \dots, f_{d-1}) : \{0,1\}^d \rightarrow \{0,1\}^d$ and $g : \{0,1\}^d \rightarrow \mathbb{Z}_q$, represented by normalized matrices $\{\mathbf{M}_{f_i}\}_{i \in [d]}$ and $\mathbf{M}_g$. The homomorphic algorithms of [AMR26c] compute

- $\text{BPreadM}(\text{pk}, \{\mathbf{M}_{f_i}\}_{i \in [d]}, \mathbf{M}_g) \mapsto \mathbf{M}_{g \circ f}$,
- $\text{BPreadH}(\text{pk}, \alpha, \beta, \{\mathbf{M}_{f_i}\}_{i \in [d]}, \mathbf{M}_g) \mapsto \{\mathbf{H}_{g,f,\alpha,\beta}^{(i)} \in \mathbb{Z}_q^{n \times n}\}_{i \in [d]}, \mathbf{H}'_{g,f,\alpha,\beta} \in \mathbb{Z}_q^{nt^2 \times n}$

such that

$$\mathbf{GCan}(\mathbf{M}_{g \circ f})(\alpha) - \mathbf{GCan}(\mathbf{M}_g)(\beta) = \sum_{i \in [d]} (\mathbf{GCan}(\mathbf{M}_{f_i})(\alpha) - \beta_i \mathbf{G}) \mathbf{H}_{g,f,\alpha,\beta}^{(i)} + \mathbf{Z}\mathbf{H}'_{g,f,\alpha,\beta}. \quad (33)$$

The construction and its norm analysis give every coefficient matrix in Equation (33) a transpose norm at most $(\lambda\sigma)^{O(d \log(d+1))} \leq 2^{d \cdot \text{polylog}(\lambda)}$.

We recall the definition of the small-secret circular decomposed LWE assumption, as stated in [AMR25a]. Below Bits stacks the bit decomposition of the input columnwise.

Definition 5.1 (Small-Secret Circular Decomposed LWE). *Let r be polynomially bounded and let $\sigma > 0$. Small-Secret Circular Decomposed LWE postulates that the following distribution ensembles are computationally indistinguishable:*

$$\left(\mathbf{A}, \mathbf{B}, \mathbf{Z}, \mathbf{Q}, \mathbf{Z}^\top \mathbf{s} + \mathbf{e}_z, \mathbf{Q}^\top \mathbf{s} + \mathbf{e}_c, \left[\begin{array}{c} \mathbf{M} \\ \mathbf{r}^\top \mathbf{M} + \mathbf{e}_m^\top \end{array} \right] + \text{Bits}(\mathbf{s})^\top \otimes \mathbf{G} \right) \approx_c (\mathbf{A}, \mathbf{B}, \mathbf{Z}, \mathbf{Q}, \mathbf{v}_Z, \mathbf{v}_Q, \mathbf{v}_M),$$

where $\mathbf{Q} \xleftarrow{\$} \mathbb{Z}_q^{k \times r}$, $\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{(k-1) \times n^2}$, $\mathbf{r} \xleftarrow{\$} \mathcal{D}_\sigma^{k-1}$, $\mathbf{s}^\top = (\mathbf{r}^\top, -1)$, $\mathbf{e}_z \xleftarrow{\$} \mathcal{D}_\sigma^{nt^2}$, $\mathbf{e}_c \xleftarrow{\$} \mathcal{D}_\sigma^r$, $\mathbf{e}_m \xleftarrow{\$} \mathcal{D}_\sigma^{n^2}$, $\mathbf{v}_Z \xleftarrow{\$} \mathbb{Z}_q^{nt^2}$, $\mathbf{v}_Q \xleftarrow{\$} \mathbb{Z}_q^r$, $\mathbf{v}_M \xleftarrow{\$} \mathbb{Z}_q^{k \times n^2}$, and $(\mathbf{A}, \mathbf{B}, \mathbf{Z}) \xleftarrow{\$} \text{DSample}$.

5.2 Bootstrapping Lattice Encodings

We recall the bootstrapping procedure of Hsieh, Lin and Luo [HLL23]. Let $\mathbf{c}^\top := \mathbf{s}^\top \cdot (\mathbf{GM} - x\mathbf{G}) + \mathbf{e}^\top$ be an encoding of x under a *low norm* LWE secret $\mathbf{s}^\top = (\mathbf{r}^\top, -1)$, where $\mathbf{r}$ is sampled from $\mathcal{D}_\sigma^{k-1}$ and $\|\mathbf{e}\|_\infty, \|\mathbf{s}\|_\infty \leq B$ for some bound B such that B^3/q and $\sigma 2^{d \cdot \text{polylog}(\lambda)}/B$ are both negligible. Define the following circular ciphertext from [GSW13]

$$\mathbf{c}_{\text{circ}}^\top := \mathbf{s}^\top \cdot (\mathbf{GM}_{\text{circ}} - (1, \text{Bits}(\text{ct}_{\text{circ}})^\top) \otimes \mathbf{G}) + \mathbf{e}_{\text{circ}}^\top \quad \text{and} \quad \text{ct}_{\text{circ}} := \begin{bmatrix} \mathbf{M}_{\text{pk}} \\ \mathbf{r}^\top \mathbf{M}_{\text{pk}} + \mathbf{e}_{\text{pk}}^\top \end{bmatrix} + \text{Bits}(\mathbf{s})^\top \otimes \mathbf{G} \quad (34)$$

where $\mathbf{M}_{\text{circ}} \xleftarrow{\$} \{0, 1\}^{n \times (n^4 + n)}$, $\mathbf{M}_{\text{pk}} \xleftarrow{\$} \mathbb{Z}_q^{(k-1) \times n^2}$, $\mathbf{e}_{\text{circ}} \xleftarrow{\$} \mathcal{D}_\sigma^{n^4 + n}$, and $\mathbf{e}_{\text{pk}} \xleftarrow{\$} \mathcal{D}_\sigma^{n^2}$. The first circular block encodes 1. In the evaluations below, every constant and direct public input is supplied as its multiple of this block, followed by normalization. Thus setting all encoded inputs formally to zero makes the evaluated circuit zero. The work of [HLL23] defines algorithms with the following syntax:

- $\text{HLLRound}(\mathbf{c}) \mapsto \mathbf{c}_{\text{round}}$,
- $\text{HLLValM}(\mathbf{M}, \mathbf{M}_{\text{circ}}) \mapsto \mathbf{M}_{\text{boost}}$,
- $\text{HLLVal}(\mathbf{M}, \mathbf{M}_{\text{circ}}, \mathbf{c}, \mathbf{c}_{\text{circ}}, \text{ct}_{\text{circ}}) \mapsto \mathbf{c}_{\text{boost}}$.

We use HLLRound for the algorithm `RemoveNoise` of [HLL23, Construction 1], instantiated with $M = 2^{\lceil \log_2((Bq)^{1/4}) \rceil}$. [HLL23] shows that one can interpret the vector $\mathbf{c}_{\text{round}}^\top = \mathbf{m}_{\text{Pad}}^\top - x\mathbf{s}^\top \mathbf{G}$, where $\mathbf{m}_{\text{Pad}}$ depends on the public parameters and the secret $\mathbf{s}$. Moreover, the HLLVal algorithm computes a second vector $\mathbf{c}_{\text{Pad}}^\top = \mathbf{s}^\top \mathbf{GM}_{\text{boost}} - \mathbf{m}_{\text{Pad}}^\top + \mathbf{e}_{\text{boost}}^\top$ by homomorphically applying a function $g_{\text{Pad}, \mathbf{M}}$ to $\mathbf{c}_{\text{circ}}$, followed by a multiplication by the fixed binary recombination matrix $\mathbf{w} \in \{0, 1\}^{n^3 \times n}$ with $\|\mathbf{w}^\top\|_{\infty, \infty} = n$. Here $g_{\text{Pad}, \mathbf{M}}$ outputs the n^2 bits of a vector-valued GSW encryption of the pad; $\mathbf{w}$ selects the corresponding gadget columns to reconstruct its $k \times n$ ciphertext matrix. The output is

$$\mathbf{c}_{\text{boost}}^\top = \mathbf{c}_{\text{round}}^\top + \mathbf{c}_{\text{Pad}}^\top = \mathbf{s}^\top (\mathbf{GM}_{\text{boost}} - x\mathbf{G}) + \mathbf{e}_{\text{boost}}^\top \quad (35)$$

where $\|\mathbf{e}_{\text{boost}}\|_\infty \leq \sigma 2^{\delta \cdot O(\log \lambda)} \leq \sigma 2^{\text{polylog}(\lambda)} \ll B$ and $\delta = \text{polylog}(\lambda)$ is the circuit depth of $g_{\text{Pad}, \mathbf{M}}$.

The following lemma is a direct consequence of [HLL23] and gives a formal bound on the probability of a rounding error.

Lemma 5.2. *Suppose that polynomially many matrices $\mathbf{M}$, $\mathbf{H}$, and $\mathbf{H}'$ are efficiently computed from $(\text{pk}, \mathbf{Q})$ and data independent of $\mathbf{s}$ and the noisy samples, and satisfy*

$$\mathbf{GM} = \mathbf{ZH} + \mathbf{QH}' \pmod{q}, \quad \|[\mathbf{H}^\top \mid \mathbf{H}'^\top]\|_{\infty, \infty} \leq 2^{d \cdot \text{polylog}(\lambda)}.$$

Under the small-secret DLWE marginal of Definition 5.1, except with negligible probability, simultaneously for every such $\mathbf{M}$, every integer x , and every $\mathbf{e}$ satisfying $\|\mathbf{e}\|_\infty, |x| \|\mathbf{s}\|_\infty \leq B$, we have

$$\text{HLLRound}((\mathbf{s}^\top (\mathbf{GM} - x\mathbf{G}) + \mathbf{e}^\top)^\top)^\top = \underbrace{\text{HLLRound}((\mathbf{s}^\top \mathbf{GM})^\top)^\top}_{=:\mathbf{m}_{\text{Pad}}^\top} - x\mathbf{s}^\top \mathbf{G}.$$

Proof. On the Gaussian tail event,

$$[(\mathbf{Z}^\top \mathbf{s} + \mathbf{e}_z)^\top \mid (\mathbf{Q}^\top \mathbf{s} + \mathbf{e}_c)^\top] \begin{bmatrix} \mathbf{H} \\ \mathbf{H}' \end{bmatrix} = \mathbf{s}^\top \mathbf{G} \mathbf{M} + [\mathbf{e}_z^\top \mid \mathbf{e}_c^\top] \begin{bmatrix} \mathbf{H} \\ \mathbf{H}' \end{bmatrix} \pmod{q},$$

and

$$\left\| [\mathbf{e}_z^\top \mid \mathbf{e}_c^\top] \begin{bmatrix} \mathbf{H} \\ \mathbf{H}' \end{bmatrix} \right\|_\infty \leq \sigma \sqrt{\lambda} 2^{d \cdot \text{polylog}(\lambda)}.$$

The small-secret DLWE marginal of [Definition 5.1](#) allows the two noisy vectors in the first display to be replaced by uniform vectors independent of $\mathbf{H}$ and $\mathbf{H}'$. Applying the argument of [\[HLL23, Lemma 8\]](#) modulo q and modulo $2^{\lfloor \log_2((Bq)^{1/4}) \rfloor}$ bounds the probability that one of the two hypotheses of [\[HLL23, Theorem 11\]](#) fails by

$$\text{poly}(\lambda) \left(\frac{2B 2^{2 \lfloor \log_2((Bq)^{1/4}) \rfloor} + (2\sigma\sqrt{\lambda} + 1) 2^{d \cdot \text{polylog}(\lambda)}}{q} + \frac{2B + (2\sigma\sqrt{\lambda} + 1) 2^{d \cdot \text{polylog}(\lambda)}}{2^{\lfloor \log_2((Bq)^{1/4}) \rfloor}} \right) + \text{negl}(\lambda).$$

This is negligible because B^3/q and $\sigma 2^{d \cdot \text{polylog}(\lambda)}/B$ are negligible. A union bound applies to all the matrices. The claimed identity follows from [\[HLL23, Theorem 11\]](#). $\square$

The following lemma shows how to apply the bootstrapping procedure to the encodings of [\[AMR26a\]](#).

Lemma 5.3. *Let $f : \{0, 1\}^d \rightarrow \{0, 1\}$ and $\alpha \in \{0, 1\}^d$ and suppose that*

$$\mathbf{c}_{f,\alpha}^\top = \mathbf{s}^\top (\mathbf{G} \mathbf{B} \mathbf{P}_f(\alpha) - f(\alpha) \mathbf{G}) + \mathbf{e}^\top \quad \text{and} \quad \mathbf{z}^\top = \mathbf{s}^\top \cdot \mathbf{Z} + \mathbf{e}_{\text{DLWE}}^\top$$

for a read-once branching program $\mathbf{B} \mathbf{P}_f$, where $\|\mathbf{e}\|_\infty \leq B$, $\mathbf{e}_{\text{DLWE}} \stackrel{\$}{\leftarrow} \mathcal{D}_\sigma^{nt^2}$ and $\mathbf{G} \mathbf{B} \mathbf{P}_f(\alpha)$ satisfies the hypotheses of [Lemma 5.2](#). There is a deterministic polynomial-time algorithm that, on input $\text{pk}, \mathbf{B} \mathbf{P}_f, \alpha, \mathbf{c}_{f,\alpha}, \mathbf{z}, \mathbf{M}_{\text{circ}}, \mathbf{c}_{\text{circ}}$ and ct_{circ} as in [Equation \(34\)](#), computes a read-once matrix branching program $\tilde{\mathbf{B}} \mathbf{P}_f$ and an encoding $\tilde{\mathbf{c}}_{f,\alpha}$ such that

$$\tilde{\mathbf{c}}_{f,\alpha}^\top = \mathbf{s}^\top (\mathbf{G} \tilde{\mathbf{B}} \mathbf{P}_f(\alpha) - f(\alpha) \mathbf{G}) + \tilde{\mathbf{e}}^\top$$

with $\|\tilde{\mathbf{e}}\|_\infty \leq \sigma 2^{d \cdot \text{polylog}(\lambda)} \ll B$ except with negligible probability.

Proof. Let $\mathbf{B} \mathbf{P}_{\text{circ}}$ denote the $1+n^3$ trivial branching programs whose outputs are the $n \times n$ blocks of $\mathbf{M}_{\text{circ}}$. Define $g_{\text{Pad}, \mathbf{B} \mathbf{P}_f}(\alpha, \text{ct}) := g_{\text{Pad}, \mathbf{B} \mathbf{P}_f(\alpha)}(\text{ct})$. Apply the following computations componentwise to the n^2 outputs and concatenate their results:

$$\begin{aligned} \mathbf{M}_{\text{boost}} &:= \text{BPEvalM}(\text{pk}, g_{\text{Pad}, \mathbf{B} \mathbf{P}_f}, \mathbf{B} \mathbf{P}_{\text{circ}}), \\ (\mathbf{H}_{\text{boost}}, \mathbf{H}'_{\text{boost}}) &:= \text{BPEvalH}(\text{pk}, g_{\text{Pad}, \mathbf{B} \mathbf{P}_f}, \alpha, (1, \text{Bits}(\text{ct}_{\text{circ}})^\top), \mathbf{B} \mathbf{P}_{\text{circ}}), \end{aligned}$$

where $\mathbf{H}_{\text{boost}} \in \mathbb{Z}_q^{(n^4+n) \times n^3}$, and $\mathbf{H}'_{\text{boost}} \in \mathbb{Z}_q^{nt^2 \times n^3}$. Define $\tilde{\mathbf{B}} \mathbf{P}_f$ by $\tilde{\mathbf{B}} \mathbf{P}_f(\alpha) := \text{Can}(\mathbf{M}_{\text{boost}})(\alpha) \mathbf{w}$ and set

$$\tilde{\mathbf{c}}_{f,\alpha}^\top = (\mathbf{c}_{\text{circ}}^\top \cdot \mathbf{H}_{\text{boost}} + \mathbf{z}^\top \cdot \mathbf{H}'_{\text{boost}}) \cdot \mathbf{w} + \text{HLLRound}(\mathbf{c}_{f,\alpha})^\top.$$

By [Lemma 5.2](#),

$$\text{HLLRound}(\mathbf{c}_{f,\alpha})^\top = \mathbf{m}_{\text{Pad}, f, \alpha}^\top - f(\alpha) \mathbf{s}^\top \mathbf{G}.$$

Moreover, by Equation (32) and [HLL23, Theorem 12],

$$(\mathbf{c}_{\text{circ}}^\top \mathbf{H}_{\text{boost}} + \mathbf{z}^\top \mathbf{H}'_{\text{boost}}) \mathbf{w} = \mathbf{s}^\top \mathbf{G} \tilde{\mathbf{M}}_{f,\alpha} - \mathbf{m}_{\text{Pad},f,\alpha}^\top + \tilde{\mathbf{e}}_{f,\alpha}^\top,$$

where $\tilde{\mathbf{M}}_{f,\alpha} = \tilde{\mathbf{B}}\mathbf{P}_f(\alpha)$ and

$$\left\| \tilde{\mathbf{e}}_{f,\alpha}^\top - \mathbf{e}_{\text{circ}}^\top \mathbf{H}_{\text{boost}} \mathbf{w} - \mathbf{e}_{\text{DLWE}}^\top \mathbf{H}'_{\text{boost}} \mathbf{w} \right\|_\infty \leq \sigma 2^{\text{polylog}(\lambda)}.$$

The norm bounds from Section 5.1 and $\|\mathbf{w}^\top\|_{\infty,\infty} = n$ therefore give $\|\tilde{\mathbf{e}}_{f,\alpha}\|_\infty \leq \sigma 2^{d \cdot \text{polylog}(\lambda)}$. $\square$

By interleaving the bootstrapping procedure from Lemma 5.3 with the homomorphic evaluations from Equation (32), we obtain the new evaluation procedure.

Theorem 5.4 (Multi-Hop Evaluation with Bootstrapping). *Let $h : \{0, 1\}^d \times \mathbb{Z}_q^a \rightarrow \mathbb{Z}_q$ be a function described by a polynomial-size arithmetic circuit of arbitrary depth. There are deterministic algorithms with interfaces*

- $\text{BPEvalCircM}(\text{pk}, h, \{\mathbf{BP}_i\}_{i \in [a]}, \mathbf{M}_{\text{circ}}) \mapsto \mathbf{M}_h,$
- $\text{BPEvalCircC}(\text{pk}, h, \alpha, \beta, \text{ct}_{\text{circ}}, \{\mathbf{BP}_i\}_{i \in [a]}, \mathbf{M}_{\text{circ}}, \{\mathbf{c}_i\}_{i \in [a]}, \mathbf{c}_{\text{circ}}, \mathbf{z}) \mapsto \mathbf{c}_{h,\alpha}$

such that, if all wires of $h(\alpha, \beta)$ have magnitude at most $B/(\sigma \cdot 2^{d \cdot \text{polylog}(\lambda)})$ and

$$\|\mathbf{c}_i^\top - \mathbf{s}^\top (\mathbf{G} \cdot \mathbf{BP}_i(\alpha) - \beta_i \mathbf{G})\|_\infty \leq B, \quad (i \in [a]),$$

then, on the Gaussian tail event and provided every bootstrap satisfies the rounding identity above, it holds that

$$\|\mathbf{c}_{h,\alpha}^\top - \mathbf{s}^\top (\mathbf{G} \cdot \text{Can}(\mathbf{M}_h)(\alpha) - h(\alpha, \beta) \mathbf{G})\|_\infty \leq \sigma \cdot 2^{d \cdot \text{polylog}(\lambda)}.$$

When $a = 0$, we omit β , $\{\mathbf{BP}_i\}_{i \in [a]}$, and $\{\mathbf{c}_i\}_{i \in [a]}$ from the notation.

5.3 Hard Sequential Function

Let f be described by a circuit with d layers of NAND and copy gates, such that there are at most 2^κ gates in each layer. Let $\phi_L, \phi_R : [d] \times \{0, 1\}^\kappa \rightarrow \{0, 1\}^\kappa$ be the maps that send a pair (i, α) to the indexes of the left and right inputs of the α -th gate in layer i , respectively. For every $b \in [\kappa]$, let $\phi_{L,b}(i, \alpha), \phi_{R,b}(i, \alpha)$ be the functions that output the b -th bit of $\phi_L(i, \alpha)$ and $\phi_R(i, \alpha)$, respectively. Let $\phi_N : [d] \times \{0, 1\}^\kappa \rightarrow \{0, 1\}$ be the map that sends a pair (i, α) to 1 if and only if the α -th gate in the i -th layer is a NAND gate.

Our construction is described below.

KeyGen($1^\lambda, 1^\kappa, 1^d$): Sample $\text{pk} \xleftarrow{\$} \text{Setup}(1^\lambda)$ and independent uniform bit matrices

$$\begin{aligned} \mathbf{W}_{i,L,b}, \mathbf{W}_{i,R,b} &\xleftarrow{\$} \{0, 1\}^{n \times mn^4} & (i \in [d], b \in [\kappa]), \\ \mathbf{W}_{i,N} &\xleftarrow{\$} \{0, 1\}^{n \times mn^4} & (i \in [d]), \\ \mathbf{M}_{\text{circ}} &\xleftarrow{\$} \{0, 1\}^{n \times (n^4 + n)} \end{aligned}$$

together with $\mathbf{a} \xleftarrow{\$} \{0, 1\}^n$. Return all these values as hk .

$\text{Hash}(\text{hk}, 1^d, 1^T, x)$: Identify $\alpha \in \{0, 1\}^\kappa$ with the integer represented by its binary expansion, and define

$$X_x(\alpha) := \begin{cases} x_\alpha, & \alpha < |x|, \\ 0, & \text{otherwise.} \end{cases}$$

Equivalently, if j_b is the b -th bit of the κ -bit representation of j , then

$$X_x(\alpha) = \sum_{j \in [\min\{|x|, 2^\kappa\}]} x_j \prod_{b=0}^{\kappa-1} (j_b \alpha_b + (1 - j_b)(1 - \alpha_b)).$$

Compute

$$\mathbf{M}^{(0)} := \text{BPEvalCircM}(\text{pk}, X_x, \mathbf{M}_{\text{circ}}).$$

For $j = 0, \dots, dT - 1$, compute

$$\begin{aligned} \mathbf{M}_L^{(j)} &:= \text{BReadM}(\text{pk}, \{\mathbf{W}_{j \bmod d, L, b}\}_{b \in [\kappa]}, \mathbf{M}^{(j)}), \\ \mathbf{M}_R^{(j)} &:= \text{BReadM}(\text{pk}, \{\mathbf{W}_{j \bmod d, R, b}\}_{b \in [\kappa]}, \mathbf{M}^{(j)}), \\ \mathbf{M}^{(j+1)} &:= \text{BPEvalCircM}(\text{pk}, \text{Gate}, \{\text{Can}(\mathbf{W}_{j \bmod d, N}), \text{Can}(\mathbf{M}_L^{(j)}), \text{Can}(\mathbf{M}_R^{(j)})\}, \mathbf{M}_{\text{circ}}), \end{aligned}$$

where

$$\text{Gate}(\alpha, c, l, r) := (1 - c)l + c(1 - lr).$$

On Boolean inputs this equals $1 - lr$ when $c = 1$ and l when $c = 0$; in particular, it is computed by a constant-depth arithmetic circuit. If $\alpha_0, \dots, \alpha_{n-1}$ are the designated output-gate addresses, return

$$\mathbf{G}^{-1}(\mathbf{GM}^{(dT)}[\mathbf{P}_{\alpha_0} \mid \dots \mid \mathbf{P}_{\alpha_{n-1}}]\mathbf{v} + \mathbf{Ga}) \quad (36)$$

where

$$\mathbf{v} := \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_{n-1} \end{bmatrix} \in \{0, 1\}^{n^2}. \quad (37)$$

Here $\mathbf{e}_j \in \{0, 1\}^n$ is the j -th standard basis vector.

We prove that the function described above is correlation intractable.

Theorem 5.5 (Correlation Intractability for Sequential Functions). *Let $f : \{0, 1\}^{2^\kappa} \rightarrow \{0, 1\}^{2^\kappa}$ be described by a circuit of polynomial size S with d layers of NAND and COPY gates, where each layer has at most 2^κ gates. Fix $\alpha_0, \dots, \alpha_{n-1} \in \{0, 1\}^\kappa$, and define $\pi : \{0, 1\}^{2^\kappa} \rightarrow \{0, 1\}^n$ by*

$$\pi(x) := (x_{\alpha_0}, \dots, x_{\alpha_{n-1}}).$$

Suppose that $q = \sigma 2^{\kappa \cdot \text{polylog}(\lambda) + \omega(\log \lambda)}$, with the fixed polynomial large enough for the conditions on B above, using κ in place of d . Under the hardness of small-secret circular decomposed-LWE (Definition 5.1), for every non-uniform PPT adversary $\mathcal{A}$, there exists a negligible function negl such that, for every sufficiently large $\lambda \in \mathbb{N}$,

$$\Pr \left[\text{Hash}(\text{hk}, 1^d, 1^T, x) = \pi(f^{(T)}(X_x)) \mid \begin{array}{l} \text{hk} \xleftarrow{\$} \text{KeyGen}(1^\lambda, 1^\kappa, 1^d), \\ (1^T, x) \xleftarrow{\$} \mathcal{A}(1^\lambda, \text{hk}) \end{array} \right] \leq \text{negl}(\lambda).$$

Proof. Suppose $\mathcal{A}$ succeeds with non-negligible probability $\varepsilon(\lambda)$ in finding $T = \text{poly}(\lambda)$ and x such that

$$\text{Hash}(\text{hk}, 1^d, 1^T, x) = \pi(f^{(T)}(X_x)).$$

Since the Gaussian tail event fails with negligible probability, the conditional success probability remains non-negligible. Take the layered NAND-and-COPY conversion of f and, if necessary, append COPY layers so that there are exactly d layers. Use the routing functions defined above, using the shorthand $\phi_{i,L,b} := \phi_{L,b}(i, \cdot)$ and likewise for other functions. In the challenge of [Definition 5.1](#), use $r = n^4 + n + d(2\kappa + 1)mn^4 + 1$. Reserve the first $n^4 + n$ columns of $\mathbf{Q}$ for the circular encoding and partition its remaining columns as

$$\mathbf{U} = [\{\mathbf{U}_{i,L,b}, \mathbf{U}_{i,R,b}\}_{i,b} \mid \{\mathbf{U}_{i,N}\}_i \mid \mathbf{u}_\star],$$

where every block preceding the final one-column block $\mathbf{u}_\star$ has mn^4 columns. Let $\mathbf{w}_{i,L,b}$, $\mathbf{w}_{i,R,b}$, $\mathbf{w}_{i,N}$ and $w_\star$ be the corresponding LWE samples.

Let $\mathbf{z}$ be the challenge sample for $\mathbf{Z}$, and let ct_{circ} be its final component. Set

$$\mathbf{M}_{\text{circ}} := \mathbf{G}^{-1}(\mathbf{Q}_{:, [n^4+n]} + (1, \text{Bits}(\text{ct}_{\text{circ}})^\top) \otimes \mathbf{G}),$$

and let $\mathbf{c}_{\text{circ}}$ be the sample corresponding to these columns of $\mathbf{Q}$. In the real challenge this gives [Equation \(34\)](#). In the random challenge, since $\mathbf{G}^{-1}$ is a bijection, $\mathbf{M}_{\text{circ}}$ is uniform and independent of $(\text{pk}, \mathbf{s}, \text{ct}_{\text{circ}})$.

Each scalar routing function is computed on Boolean inputs by the formula

$$\phi(\alpha) = \sum_{\substack{j \in [2^\kappa] \\ \phi(j)=1}} \prod_{b=0}^{\kappa-1} (j_b \alpha_b + (1 - j_b)(1 - \alpha_b)),$$

where the routing functions are zero at unused gate positions, which copy address zero. The sum has at most S terms, including any appended copy gates. Thus balanced products and a balanced sum give size $O(\kappa(S + n))$ and depth $O(\kappa)$. For every $\phi \in \{\phi_{i,L,b}, \phi_{i,R,b}, \phi_{i,N}\}$, set

$$\mathbf{M}_\phi := \text{BPEvalCircM}(\text{pk}, \phi, \mathbf{M}_{\text{circ}}), \quad \mathbf{c}_{\phi,\alpha} := \text{BPEvalCircC}(\text{pk}, \phi, \alpha, \text{ct}_{\text{circ}}, \mathbf{M}_{\text{circ}}, \mathbf{c}_{\text{circ}}, \mathbf{z}).$$

Then [Theorem 5.4](#) gives

$$\|\mathbf{c}_{\phi,\alpha}^\top - \mathbf{s}^\top \cdot (\mathbf{G} \cdot \text{Can}(\mathbf{M}_\phi)(\alpha) - \phi(\alpha)\mathbf{G})\|_\infty \leq \sigma \cdot 2^{\kappa \cdot \text{polylog}(\lambda)}. \quad (38)$$

Program the corresponding hash-key matrices by

$$\begin{aligned} \mathbf{W}_{i,L,b} &:= \mathbf{G}^{-1}(\mathbf{G}\mathbf{M}_{\phi_{i,L,b}} + \mathbf{U}_{i,L,b}), \\ \mathbf{W}_{i,R,b} &:= \mathbf{G}^{-1}(\mathbf{G}\mathbf{M}_{\phi_{i,R,b}} + \mathbf{U}_{i,R,b}), \\ \mathbf{W}_{i,N} &:= \mathbf{G}^{-1}(\mathbf{G}\mathbf{M}_{\phi_{i,N}} + \mathbf{U}_{i,N}), \quad \mathbf{a} := \mathbf{G}^{-1}(\mathbf{u}_\star). \end{aligned}$$

All programmed matrices are jointly exactly uniform, with the distribution specified by [KeyGen](#), in both challenge distributions. Moreover, [Equation \(38\)](#) gives

$$\begin{aligned} \tilde{\mathbf{c}}_{\phi_{i,L,b},\alpha}^\top &:= \mathbf{c}_{\phi_{i,L,b},\alpha}^\top + \mathbf{w}_{i,L,b}^\top \cdot \mathbf{P}_\alpha = \mathbf{s}^\top (\mathbf{G}\mathbf{W}_{i,L,b}\mathbf{P}_\alpha - \phi_{i,L,b}(\alpha)\mathbf{G}) + \mathbf{e}_{i,L,b}^\top, & \|\mathbf{e}_{i,L,b}\|_\infty &\leq \sigma \cdot 2^{\kappa \cdot \text{polylog}(\lambda)} \\ \tilde{\mathbf{c}}_{\phi_{i,R,b},\alpha}^\top &:= \mathbf{c}_{\phi_{i,R,b},\alpha}^\top + \mathbf{w}_{i,R,b}^\top \cdot \mathbf{P}_\alpha = \mathbf{s}^\top (\mathbf{G}\mathbf{W}_{i,R,b}\mathbf{P}_\alpha - \phi_{i,R,b}(\alpha)\mathbf{G}) + \mathbf{e}_{i,R,b}^\top, & \|\mathbf{e}_{i,R,b}\|_\infty &\leq \sigma \cdot 2^{\kappa \cdot \text{polylog}(\lambda)} \end{aligned} \quad (39)$$

$$\tilde{\mathbf{c}}_{\phi_{i,N},\alpha}^\top := \mathbf{c}_{\phi_{i,N},\alpha}^\top + \mathbf{w}_{i,N}^\top \cdot \mathbf{P}_\alpha = \mathbf{s}^\top (\mathbf{G}\mathbf{W}_{i,N}\mathbf{P}_\alpha - \phi_{i,N}(\alpha)\mathbf{G}) + \mathbf{e}_{i,N}^\top, \quad \|\mathbf{e}_{i,N}\|_\infty \leq \sigma \cdot 2^{\kappa \cdot \text{polylog}(\lambda)}. \quad (40)$$

Give the resulting key to the adversary and let $(1^T, x)$ be its output.

The horizontal concatenation of $\mathbf{GM}_{\text{circ}}$ and all $\mathbf{GW}$ matrices is uniform and independent of $(\text{pk}, \text{s}, \text{ct}_{\text{circ}})$, and may therefore play the role of $\mathbf{Q}$ in [Lemma 5.2](#). Setting the encoded inputs to zero in [Equation \(32\)](#) expresses every matrix produced by a bootstrap using $\mathbf{GM}_{\text{circ}}$ and $\mathbf{Z}$. [Equation \(33\)](#), with $\beta = 0^\kappa$, expresses every matrix produced by a read using the $\mathbf{GW}$ matrices, the matrix produced by the preceding bootstrap, and $\mathbf{Z}$. Therefore the coefficient sums remain bounded by $2^{\kappa \cdot \text{polylog}(\lambda)}$ and do not accumulate with T . These coefficients and the required addresses depend only on the public matrices and the adversary's output. The T circuit copies contain at most TS gates, so only polynomially many bootstraps are used. Hence [Lemma 5.2](#) makes all required rounding identities hold simultaneously except with negligible probability.

Set $\mathbf{c}_\alpha^{(0)} := \text{BPEvalCircC}(\text{pk}, X_x, \alpha, \text{ct}_{\text{circ}}, \mathbf{M}_{\text{circ}}, \mathbf{c}_{\text{circ}}, \mathbf{z})$. We have

$$\|\mathbf{c}_\alpha^{(0)\top} - \mathbf{s}^\top(\mathbf{GM}^{(0)}\mathbf{P}_\alpha - X_x(\alpha)\mathbf{G})\|_\infty \leq \sigma 2^{\kappa \cdot \text{polylog}(\lambda)}.$$

Apply this invariant, [Equation \(39\)](#), [Equation \(40\)](#), and [Theorem 5.4](#) through the dT layers, using [Equation \(33\)](#) for each read. For every output address α_j , this yields an efficiently computable encoding $\mathbf{c}_{\alpha_j}^{(dT)}$ such that

$$\left\| \mathbf{c}_{\alpha_j}^{(dT)\top} - \mathbf{s}^\top(\mathbf{GM}^{(dT)}\mathbf{P}_{\alpha_j} - f^{(T)}(X_x)_{\alpha_j}\mathbf{G}) \right\|_\infty \leq \sigma \cdot 2^{\kappa \cdot \text{polylog}(\lambda)}. \quad (41)$$

Combining the equations with [Equation \(37\)](#) gives

$$\left\| \left[\mathbf{c}_{\alpha_0}^{(dT)\top} \mid \dots \mid \mathbf{c}_{\alpha_{n-1}}^{(dT)\top} \right] \mathbf{v} - \mathbf{s}^\top(\mathbf{GM}^{(dT)}[\mathbf{P}_{\alpha_0} \mid \dots \mid \mathbf{P}_{\alpha_{n-1}}]\mathbf{v} - \mathbf{G}\pi(f^{(T)}(X_x))) \right\|_\infty \leq \sigma 2^{\kappa \cdot \text{polylog}(\lambda)}. \quad (42)$$

By assumption, $\text{Hash}(\text{hk}, 1^d, 1^T, x) = \pi(f^{(T)}(X_x))$. Applying $\mathbf{G}$ to [Equation \(36\)](#) yields

$$\mathbf{GM}^{(dT)}[\mathbf{P}_{\alpha_0} \mid \dots \mid \mathbf{P}_{\alpha_{n-1}}]\mathbf{v} - \mathbf{G}\pi(f^{(T)}(X_x)) = -\mathbf{u}_\star.$$

We conclude that, if the challenger was provided with real LWE samples, it holds that

$$\left\| \left[\mathbf{c}_{\alpha_0}^{(dT)\top} \mid \dots \mid \mathbf{c}_{\alpha_{n-1}}^{(dT)\top} \right] \mathbf{v} + w^\star \right\|_\infty \leq \sigma 2^{\kappa \cdot \text{polylog}(\lambda)}.$$

The distinguisher accepts exactly when the displayed inequality holds. If instead the challenger provided random values, $w^\star$ is sampled uniformly at random over $\mathbb{Z}_q$ and independently of the key and all reconstructed encodings, since none of their computations uses $w^\star$. This means that the above relation holds with probability at most $(2\sigma 2^{\kappa \cdot \text{polylog}(\lambda)} + 1)/q = \text{negl}(\lambda)$. $\square$

Corollary 5.6. *Under the hardness of small-secret circular decomposed LWE ([Definition 5.1](#)), there exists a constant $d \in \mathbb{N}$ such that, for every polynomially bounded $T(\lambda) \geq 1$, there exists $\bar{L}(\lambda) = T(\lambda)/O(\log \lambda)$ such that for every probabilistic non-uniform Turing machine $\mathcal{M}$ and $L \leq \bar{L}(\lambda)$, we have*

$$\Pr \left[\mathcal{M}^L(\text{hk}) = \text{Hash}(\text{hk}, 1^d, 1^T, \text{hk}) \mid \text{hk} \xleftarrow{\$} \text{KeyGen}(1^\lambda, 1^\lambda, 1^d) \right] \leq \text{negl}(\lambda).$$

Proof. Suppose that a probabilistic non-uniform Turing machine $\mathcal{M}$ computes $x \mapsto \text{Hash}(x, 1^d, 1^T, x)$ in at most $L \leq \bar{L}$ steps with non-negligible probability $\varepsilon(\lambda)$ over its randomness and the choice of hk . By averaging, there is a deterministic non-uniform Turing machine $\mathcal{M}'$ with the same running time and non-negligible success probability $\varepsilon'(\lambda)$ over the sampled hash key.

A transition of $\mathcal{M}'$ on its polynomially many reachable tape and advice cells is computed by a depth- $O(\log \lambda)$ Boolean circuit. Including in the state which layer is evaluated next gives a polynomial-size circuit family $\{f_\lambda\}_{\lambda \in \mathbb{N}}$ with a constant number of NAND and COPY layers such that $\mathcal{M}'^{(L)}(x) = \pi(f_\lambda^{L'}(X_x))$, and $L' = L \cdot O(\log \lambda) \leq T$. By leaving the completed configuration unchanged during the remaining iterations, there is a polynomial-size circuit family $\{g_\lambda\}_{\lambda \in \mathbb{N}}$ with a constant number d of NAND and COPY layers such that $\mathcal{M}'^{(L)}(x) = \pi(g_\lambda^{(T)}(X_x))$.

Consequently,

$$\Pr \left[\text{Hash}(\text{hk}, 1^d, 1^T, \text{hk}) = \pi(g_\lambda^{(T)}(X_{\text{hk}})) \mid \text{hk} \xleftarrow{\$} \text{KeyGen}(1^\lambda, 1^\lambda, 1^d) \right] \geq \varepsilon'(\lambda),$$

contradicting [Theorem 5.5](#). □

Acknowledgements

The constructions and proofs were produced by humans. ChatGPT was used solely for writing assistance. The authors take full responsibility for all claims and proofs made in this paper.

D.A. was supported by a fellowship with the Simons Institute for the Theory of Computing. This work was supported by Input Output (iog.io) through their funding of the Edinburgh Blockchain Technology Lab. G.M. was supported by the European Research Council through an ERC Starting Grant (Grant agreement No. 101077455, ObfusQation) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA – 390781972. L.R. was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement number 101124977 (DECrypSIS).

References

- [AAC⁺26] Damiano Abram, Gal Arnon, Valerio Cini, Paul Lou, Giulio Malavolta, and Lawrence Roy. Decomposed LWE is equivalent to succinct LWE. *Cryptology ePrint Archive*, Paper 2026/1829, 2026. [2](#)
- [Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996. [1](#)
- [AKY24] Shweta Agrawal, Simran Kumari, and Shota Yamada. Attribute based encryption for turing machines from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 352–386. Springer, Cham, August 2024. [2](#)
- [AMR25a] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Key-homomorphic computations for RAM: Fully succinct randomised encodings and more. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part III*, volume 16002 of *LNCS*, pages 236–268. Springer, Cham, August 2025. [2](#), [3](#), [9](#), [21](#)
- [AMR25b] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Succinct oblivious tensor evaluation and applications: Adaptively-secure laconic function evaluation and trapdoor hashing for all circuits. In Michal Koucký and Nikhil Bansal, editors, *57th ACM STOC*, pages 1875–1886. ACM Press, June 2025. [3](#)

- [AMR26a] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Tree encodings I: How to authenticate a non-deterministic computation, 2026. Manuscript. [2](#), [21](#), [23](#)
- [AMR26b] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Tree encodings II: Correlation intractability for all batched relations, 2026. Manuscript. [2](#), [9](#)
- [AMR26c] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Tree encodings III: Time/space hardness from lattice problems, 2026. Manuscript. [2](#), [3](#), [4](#), [6](#), [7](#), [9](#), [21](#)
- [AMYY25] Shweta Agrawal, Anuja Modi, Anshu Yadav, and Shota Yamada. Zeroizing attacks against evasive and circular evasive LWE. In Benny Applebaum and Huijia (Rachel) Lin, editors, *TCC 2025, Part II*, volume 16269 of *LNCS*, pages 259–290. Springer, Cham, December 2025. [2](#)
- [AMYY26] Shweta Agrawal, Anuja Modi, Anshu Yadav, and Shota Yamada. ABE for unbounded depth circuits from the doubly circular assumption. Cryptology ePrint Archive, Paper 2026/1439, 2026. [2](#)
- [AMZ24] Shweta Agrawal, Giulio Malavolta, and Tianwei Zhang. Time-lock puzzles from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 425–456. Springer, Cham, August 2024. [3](#), [5](#), [17](#)
- [ARS24] Damiano Abram, Lawrence Roy, and Mark Simkin. Time-based cryptography from weaker assumptions: Randomness beacons, delay functions and more. Cryptology ePrint Archive, Report 2024/769, 2024. [3](#)
- [Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993. [8](#)
- [BD21] Jeffrey Burdges and Luca De Feo. Delay encryption. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 302–326. Springer, Cham, October 2021. [3](#)
- [BG25] Nir Bitansky and Rachit Garg. Succinct randomized encodings from laconic function evaluation, faster and simpler. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part VII*, volume 15607 of *LNCS*, pages 406–436. Springer, Cham, May 2025. [3](#)
- [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 Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Berlin, Heidelberg, May 2014. [1](#), [2](#), [3](#)
- [BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, January 2012. [2](#)
- [BHR12] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *ACM CCS 2012*, pages 784–796. ACM Press, October 2012. [5](#), [16](#)

- [BÜW24] Chris Brzuska, Akin Ünal, and Ivy K. Y. Woo. Evasive LWE assumptions: Definitions, classes, and counterexamples. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 418–449. Springer, Singapore, December 2024. [2](#)
- [BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, October 2011. [1](#), [2](#)
- [BV16] Zvika Brakerski and Vinod Vaikuntanathan. Circuit-ABE from LWE: Unbounded attributes and semi-adaptive security. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 363–384. Springer, Berlin, Heidelberg, August 2016. [2](#)
- [CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998. [6](#)
- [CHW25] Jeffrey Champion, Yao-Ching Hsieh, and David J. Wu. Registered ABE and adaptively-secure broadcast encryption from succinct LWE. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part III*, volume 16002 of *LNCS*, pages 3–34. Springer, Cham, August 2025. [2](#)
- [CJJ22] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for $\mathcal{P}$ from LWE. In *62nd FOCS*, pages 68–79. IEEE Computer Society Press, February 2022. [1](#)
- [CW23] Valerio Cini and Hoeteck Wee. ABE for circuits with poly (λ) -sized keys from LWE. In *64th FOCS*, pages 435–446. IEEE Computer Society Press, November 2023. [2](#)
- [CW25] Valerio Cini and Hoeteck Wee. Faster ABE for turing machines from circular evasive LWE. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part III*, volume 15603 of *LNCS*, pages 94–125. Springer, Cham, May 2025. [2](#)
- [CW26] Jeffrey Champion and David J. Wu. Distributed monotone-policy encryption for DNFs from lattices. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology – EUROCRYPT 2026, Part V*, volume 16545 of *Lecture Notes in Computer Science*, pages 334–365. Springer, 2026. [2](#)
- [DHM⁺24] Fangqi Dong, Zihan Hao, Ethan Mook, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and ABE for RAMs from (ring-)LWE. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 107–142. Springer, Cham, August 2024. [2](#)
- [DJM⁺25] Nico Döttling, Abhishek Jain, Giulio Malavolta, Surya Mathialagan, and Vinod Vaikuntanathan. Simple and general counterexamples for private-coin evasive LWE. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 73–92. Springer, Cham, August 2025. [2](#)
- [Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009. [1](#), [2](#)

- [GKP⁺13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. How to run Turing machines on encrypted data. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 536–553. Springer, Berlin, Heidelberg, August 2013. [2](#)
- [GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32. ACM Press, 1989. [5](#), [17](#)
- [GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Berlin, Heidelberg, August 2013. [1](#), [22](#)
- [GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013. [1](#), [2](#)
- [GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 503–523. Springer, Berlin, Heidelberg, August 2015. [1](#)
- [HLL23] Yao-Ching Hsieh, Huijia Lin, and Ji Luo. Attribute-based encryption for circuits of unbounded depth from lattices. In *64th FOCS*, pages 415–434. IEEE Computer Society Press, November 2023. [2](#), [4](#), [7](#), [22](#), [23](#), [24](#)
- [HLL24] Yao-Ching Hsieh, Huijia Lin, and Ji Luo. A general framework for lattice-based ABE using evasive inner-product functional encryption. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 433–464. Springer, Cham, May 2024. [2](#)
- [LLL26] Hanjun Li, Huijia Lin, and George Lu. Succinct garbled circuits with low-depth garbling algorithms. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology – EUROCRYPT 2026, Part III*, volume 16543 of *Lecture Notes in Computer Science*, pages 246–275. Springer, 2026. [2](#)
- [LM23] Russell W. F. Lai and Giulio Malavolta. Lattice-based timed cryptography. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 782–804. Springer, Cham, August 2023. [3](#)
- [Mah18] Urmila Mahadev. Classical verification of quantum computations. In Mikkel Thorup, editor, *59th FOCS*, pages 259–267. IEEE Computer Society Press, October 2018. [1](#)
- [MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Berlin, Heidelberg, April 2012. [5](#), [8](#)
- [PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 89–114. Springer, Cham, August 2019. [6](#)

- [QWW18] Willy Quach, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and applications. In Mikkel Thorup, editor, *59th FOCS*, pages 859–870. IEEE Computer Society Press, October 2018. [3](#), [15](#)
- [Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. [1](#)
- [RSW96] Ronald L. Rivest, Adi Shamir, and David A. Wagner. Time-lock puzzles and timed-release crypto. Technical Report MIT/LCS/TR-684, MIT Laboratory for Computer Science, 1996. [3](#)
- [VWW22] Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-IO from evasive LWE. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part I*, volume 13791 of *LNCS*, pages 195–221. Springer, Cham, December 2022. [2](#)
- [Wee22] Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 217–241. Springer, Cham, May / June 2022. [2](#)
- [Wee24] Hoeteck Wee. Circuit ABE with $\text{poly}(\text{depth}, \lambda)$ -sized ciphertexts and keys from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 178–209. Springer, Cham, August 2024. [2](#)
- [Wee25] Hoeteck Wee. Almost optimal KP and CP-ABE for circuits from succinct LWE. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part III*, volume 15603 of *LNCS*, pages 34–62. Springer, Cham, May 2025. [2](#), [4](#)
- [WW25] Hoeteck Wee and David J. Wu. Unbounded distributed broadcast encryption and registered ABE from succinct LWE. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part III*, volume 16002 of *LNCS*, pages 204–235. Springer, Cham, August 2025. [2](#)
- [Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986. [5](#), [16](#)