Learning with Alternating Moduli, Arora-Ge over Composite Moduli, and Weak PRFs

Yilei Chen* Liheng Ji† Wenjie Li‡

September 9, 2026

Abstract

In TCC 2018, Boneh, Ishai, Passelègue, Sahai, and Wu propose candidates of weak and strong PRFs by evaluating linear functions over coprime moduli alternatively. Such PRFs can be evaluated by low-depth circuits and are MPC-friendly. However, they have not been able to base the security of their PRFs on well-formed assumptions other than assuming that the PRF constructions themselves are secure.

In this paper, we formalize a new assumption called Learning with Alternating Moduli (LAM). We show that over certain large moduli, the LAM assumption is as hard as the Learning with Errors (LWE) assumption. For LAM over constant moduli, we do not know how to base its hardness on the LWE assumption. Instead, we provide

(i) polynomial-time attacks on LAM with constant prime-power moduli and certain constant non-prime-power moduli, and

(ii) evidence of the sub-exponential hardness of LAM with other moduli by analyzing the effect of typical attacks.

More specifically, we put forward two new attacks. The first attack is a recursive algorithm that solves LWE with certain constant composite moduli and error distributions. The algorithm extends the Arora-Ge algorithm for LWE from prime moduli to composite moduli, and it also solves LAM for certain parameters. The second attack is a polynomial-time attack that rules out the existence of weak PRFs in $\mathsf{NC}^0[p]$ for any prime p.

Based on our studies, we propose candidate weak PRFs in $\mathsf{NC}^0[p_1, p_2]$ for some distinct primes p_1, p_2 based on LAM over constant moduli, or the Learning with Rounding (LWR) assumption over constant moduli. Compared to the weak PRF candidates by Boneh et al., our weak PRF candidates live in the same complexity class while having the advantage of being based on well-formed assumptions.

*IIIS, Tsinghua University and Shanghai Qi Zhi Institute. chenyilei@mail.tsinghua.edu.cn. Supported by Shanghai Qi Zhi Institute Innovation Program.

†IIIS, Tsinghua University and Shanghai Qi Zhi Institute. jlh23@mails.tsinghua.edu.cn.

‡IIIS, Tsinghua University and Shanghai Qi Zhi Institute. liwj22@mails.tsinghua.edu.cn.

Contents

1 Introduction **1**

1.1 Main contributions . . . 2

1.2 A concrete example of Arora-Ge with prime power moduli . . . 8

2 Preliminary **10**

3 Hardness of Learning with Alternating Modulus **11**

3.1 Search to Decision Reduction for LAM with Binary Secrets . . . 12

3.2 Reduction from Search-LWE to Search-LAM . . . 14

3.3 Reduction from Decisional-LWE to Decisional-LAM . . . 17

3.4 Reduction from Search-LAM to Search-LWE . . . 20

4 Arora-Ge Algorithm for LWE with Composite Modulus **21**

4.1 The Algorithm . . . 21

4.2 The Solution Space of the Algorithm . . . 22

4.3 The Recursive Arora-Ge Algorithm . . . 29

5 Weak PRFs do not exist in $\mathsf{NC}^0[p]$ for prime p **32**

6 Candidate Weak PRFs **35**

6.1 Weak PRFs at Least as Secure as LWE . . . 35

6.2 Low-Depth Weak PRFs Candidates from LAM . . . 35

6.3 Low-Depth Weak PRF Candidates from LWR . . . 38

A Solving linear equations on a ring **41**

B $\mathsf{NC}[q]$, Polynomials and Our wPRFs **45**

B.1 Definitions . . . 45

B.2 The relationship between $\mathsf{NC}^0[q]$ and $\mathsf{CC}^0[q]$ . . . 46

B.3 How to Compute $\mathsf{NC}[2]$ Circuits Using Low-Degree Polynomials . . . 46

B.4 How to Compute $\mathsf{NC}[p]$ Circuits Using Low-Degree Polynomials . . . 47

B.5 The Explicit Depth of Our wPRFs . . . 49

B.6 The Depth of Our wPRFs in Other Settings. . . . 51

B.7 How to Reduce $\mathsf{NC}[q]$ for Composite q to Smaller Moduli . . . 52

C Other Omitted Proofs **54**

C.1 Proof of Lemma 13 . . . 54

C.2 Proof of Lemma 14 . . . 55

C.3 Proof of Lemma 15 . . . 56

C.4 Proof of Lemma 16 . . . 56

C.5 Proof of Lemma 33 . . . 58

C.6 Proof of Corollary 32 . . . 59

C.7 Proof of Lemma 40 . . . 61

D The output of Arora-Ge Algorithm when $q = p^\kappa$ **62**

1 Introduction

Weak pseudorandom functions with low circuit depth. The pseudorandom function (PRF) is one of the fundamental building blocks in cryptography. Although a generic transformation from PRG to PRF has been given by Goldreich, Goldwasser and Micali early in 1980s, cryptographers still show a fervent interest in building low-circuit-depth PRFs, as well as weak PRFs, which are PRFs that only allow the adversary to query on random inputs instead of arbitrary inputs. As shown in a recent survey by Boura et al [BCPR25], low-depth weak PRFs are very useful for building MPC protocols.

What is the simplest circuit class where secure weak PRFs exist? This is an intriguing question in cryptography, complexity, and learning theory. The seminal work of Linial, Mansour, and Nisan [LMN89] shows that weak PRFs in AC^0 have no more than quasipolynomial security. This implies that any weak PRF candidate that achieves subexponential security must be outside the class AC^0. (In this article, we say a primitive is T-secure if any distinguisher of circuit size bounded by T succeeds in breaking the primitive with probability less than $1/T$.) In the past few years, a variety of weak PRF candidates in circuit classes slightly beyond AC^0 (like $\mathsf{AC}^0[2]$, or larger ones), have been proposed and analyzed by, e.g., Akavia et al. [ABG+14], Applebaum and Raykov [AR16], Yu and Steinberger [YS16], Bogdanov and Rosen [BR17], Boneh et al. [BIP+18], Boyle et al. [BCG+20, BCG+21], and Ding et al. [DJK25].

The circuit class $\mathsf{NC}^0[q]$. In this work, we initiated the study of weak PRFs in a circuit class called $\mathsf{NC}^0[q]$, where q is an integer with $q \geq 2$. For any $c_1, ..., c_m \in \mathbb{N}^+$, we say a circuit is in $\mathsf{NC}^0[\mathsf{MOD}_{c_1}, ..., \mathsf{MOD}_{c_m}]$ if it consists of 2-fan-in OR and AND gates, unbounded-fan-in MOD_{c_1}, ..., MOD_{c_m} gates (on input $\mathbf{x} \in \{0,1\}^n$, MOD_c outputs 0 if the number of 1s in $\mathbf{x}$ is multiple of c, outputs 1 elsewhere), and NOT gates, and has $\mathsf{poly}(n)$ size and $O(1)$ depth. We abbreviate it as $\mathsf{NC}^0[c_1, ..., c_m]$ in the rest of the paper. Occasionally we use $\mathsf{NC}[c_1, ..., c_m]$, and the depth of the circuit will be explicitly given. Let us remark that for $q > 2$, $\mathsf{NC}^0[q]$ is equivalent to the well-studied complexity class $\mathsf{CC}^0[q]$ (the class of constant-depth and polynomial-size circuits consisting of unbounded fan-in MOD_q gates and NOT gate). See Appendix B.2 for further discussions.

Besides being a simpler class than $\mathsf{AC}^0[q]$, there is a more important reason to study $\mathsf{NC}^0[q]$ circuits. In MPC, there are many scenarios where addition is cheap (can be computed locally), while multiplication is expensive (requires communication between parties). The definition of $\mathsf{NC}^0[q]$ circuits fits in these scenarios better than $\mathsf{AC}^0[q]$ circuits do: We allow unbounded-fan-in MOD_q gates for addition, and 2-fan-in AND/OR gates for multiplication. However, the study of wPRFs in $\mathsf{NC}^0[q]$ classes has been long overlooked.

Alternating-moduli weak PRFs. To the best of our knowledge, the "alternating moduli" technique of Boneh et al. [BIP+18] has been the only known method to construct sub-exponentially secure weak PRFs in $\mathsf{NC}^0[2,3]$ circuits. Here we introduce a typical construction: let $\mathsf{map} : \{0,1\}^m \to \mathbb{Z}_3$ take $\mathbf{y} \in \{0,1\}^m$ to $\sum_{i=1}^m y_i \bmod 3$. The weak PRF construction, with key $k = \mathbf{A} \in \mathbb{Z}_2^{m \times n}$, is

$$F_{\mathbf{A}}(\mathbf{x}) := \mathsf{map}(\mathbf{A}\mathbf{x} \bmod 2). \tag{1}$$

That is, the evaluation algorithm of $F_{\mathbf{A}}$ interprets the result of $\mathbf{A}\mathbf{x} \bmod 2$ as m bits, and then sums them up modulo 3.

In the work of Boneh et al. [BIP+18], they provide some evidence suggesting that $F_{\mathbf{A}}$ is a secure weak PRF against certain attacks. Over the past few years, improved weak PRFs based on alternating moduli have been proposed [DGH+21, APRR24], together with some cryptanalytic attempts [CCKK21,JMN23,AR25], but all attacks run in exponential time in the security parameter. However, a solid reduction that bases the security of $F_{\mathbf{A}}$, or other alternating-moduli wPRFs, on any well-established assumption, is not known.

Arora-Ge algorithm. Another major focus of this paper is on the Arora-Ge algorithm [AG11], which is a polynomial time algorithm for solving the learning with errors (LWE) problem when the error support is constant size, given a sufficient amount of samples. However, the original work of Arora and Ge [AG11] and its follow-up work [ACF+15, STA20, Ste24, NMSÜ25] only analyze the behavior of the Arora-Ge algorithm for LWE with prime moduli. The generalization of this algorithm to composite moduli was unknown.

Previous discussions raise the following questions:

1. Can we base any learning problems over alternating moduli on established cryptographic assumptions? Or is it possible to construct alternating-moduli low-depth weak PRFs from simpler assumptions?
2. Do weak PRFs exist in circuit classes more restricted than $\mathsf{NC}^0[p_1, p_2]$, for distinct primes p_1, p_2?
3. What is the effect of the Arora-Ge algorithm on search LWE with composite moduli? To what extent can we solve LWE over composite moduli with constant-size error support?

1.1 Main contributions

Our main contributions are as follows.

1. We formalize the learning with alternating moduli (LAM) assumption and show that the LAM problem is as hard as the standard learning with errors (LWE) problem under certain parameters.
2. We analyze the behavior of the Arora-Ge algorithm [AG11] on LWE with composite moduli. Specifically, we prove that under certain parameters, applying Arora-Ge algorithm once recovers *part* of the secret of LWE.
3. Based on the analysis above, we further design a recursive algorithm that recovers the *entire* secret for LWE with composite moduli and certain error distributions. Our algorithm also applies to LAM and learning with rounding (LWR).
4. We show that weak PRFs do not exist in $\mathsf{NC}^0[p]$ for prime p.
5. Based on the studies above, we derive new weak PRF candidates in $\mathsf{NC}^0[p_1, p_2]$ for some distinct primes p_1, p_2, assuming the hardness of LAM or LWR with constant moduli. When constant moduli are replaced by certain large moduli, the security of weak PRFs can be based on LWE. This provides variants of the "dark matter" weak PRFs based on standard assumptions.

Before giving the details of our results, let us first recall the definitions of the LWE [Reg09] and LWR [BPR12] problems.

Definition 1 (Learning with errors)**.** *Let n, m, q be positive integers. Let χ be a distribution over $\mathbb{Z}_q$. Let $\mathbf{s} \in \mathbb{Z}_q^n$ be a secret vector. The **search LWE problem** $LWE_{n,m,q,\chi}$ requires the adversary to find the secret $\mathbf{s}$ given access to an oracle that outputs $(\mathbf{a}_i, (\langle \mathbf{s}, \mathbf{a}_i \rangle + e_i) \bmod q)$ on its i-th query, for $i = 1, \ldots, m$. Here, each $\mathbf{a}_i$ is a uniformly random vector in $\mathbb{Z}_q^n$, and each error term e_i is sampled from χ.*

*The **decisional LWE problem** $DLWE_{n,m,q,\chi}$ requires the adversary to distinguish whether we are given samples $(\mathbf{A}, \mathbf{y}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$ from the LWE distribution, i.e.,*

$$\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \quad \mathbf{y} = (\mathbf{A}^\top \mathbf{s} + \mathbf{e}) \bmod q$$

where $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi^m$; or from the uniformly random distribution over $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$.

Definition 2 (Learning with rounding)**.** *Let n, m, q_1, q_2 be positive integers such that $q_1 > q_2$. Let $\mathbf{s} \in \mathbb{Z}_{q_1}^n$ be a secret vector. The **search LWR problem** LWR_{n,m,q_1,q_2} requires the adversary to find the secret $\mathbf{s}$ given access to an oracle that outputs $(\mathbf{a}_i, \lfloor (q_2/q_1) \cdot (\langle \mathbf{s}, \mathbf{a}_i \rangle \bmod q_1) \rceil \bmod q_2)$ on its i-th query, for $i = 1, \ldots, m$. Here, each $\mathbf{a}_i$ is a uniformly random vector in $\mathbb{Z}_{q_1}^n$.*

*The **decisional LWR problem** $DLWR_{n,m,q_1,q_2}$ requires the adversary to distinguish whether we are given samples $(\mathbf{A}, \mathbf{y}) \in \mathbb{Z}_{q_1}^{n \times m} \times \mathbb{Z}_{q_2}^m$ from the LWR distribution, i.e.,*

$$\mathbf{A} \leftarrow \mathbb{Z}_{q_1}^{n \times m}, \quad \mathbf{y} = \lfloor (q_2/q_1) \cdot (\mathbf{A}^\top \mathbf{s} \bmod q_1) \rceil \bmod q_2$$

where $\mathbf{s} \in \mathbb{Z}_{q_1}^n$; or from the distribution $\mathcal{U}(\mathbb{Z}_{q_1}^{n \times m}) \times (\lfloor (q_2/q_1) \cdot \mathcal{U}(\mathbb{Z}_{q_1}^m) \rceil \bmod q_2)$, where $\mathcal{U}(S)$ represents the uniform distribution on S for any finite set S.

Learning with Alternating Moduli. Our first contribution is to formalize the LAM problem and show its connections to the LWE problem.

Definition 3 (Learning with Alternating Moduli)**.** *Let n, m, q_1, q_2 be positive integers such that $q_1 > q_2 \geq 2$, and $\gcd(q_1, q_2) = 1$. Let $\mathbf{s} \in \mathbb{Z}_{q_1}^n$ be a secret vector. The **search LAM problem** LAM_{n,m,q_1,q_2} requires the adversary to find the secret $\mathbf{s}$ given access to an oracle that outputs $(\mathbf{a}_i, (\langle \mathbf{s}, \mathbf{a}_i \rangle \bmod q_1) \bmod q_2)$ on its i-th query, for $i = 1, \ldots, m$. Here, each $\mathbf{a}_i$ is a uniformly random vector in $\mathbb{Z}_{q_1}^n$.*

*The **decisional LAM problem** $DLAM_{n,m,q_1,q_2}$ requires the adversary to distinguish whether we are given samples $(\mathbf{A}, \mathbf{y}) \in \mathbb{Z}_{q_1}^{n \times m} \times \mathbb{Z}_{q_2}^m$ obtained from querying the LAM distribution, i.e,*

$$\mathbf{A} \leftarrow \mathbb{Z}_{q_1}^{n \times m}, \quad \mathbf{y} = (\mathbf{A}^\top \mathbf{s} \bmod q_1) \bmod q_2$$

where $\mathbf{s} \in \mathbb{Z}_{q_1}^n$; or from the distribution $\mathcal{U}(\mathbb{Z}_{q_1}^{n \times m}) \times (\mathcal{U}(\mathbb{Z}_{q_1}^m) \bmod q_2)$.

Let us make a few remarks on Def. 3. First, for any $q \geq 2$, we define the output of any integer modulo q to be in $[q] := \{0, ..., q-1\}$ by default (some other papers use $(-q/2, ..., q/2] \cap \mathbb{Z}$). Second, we only consider coprime moduli q_1, q_2 since otherwise $\mathbf{s} \bmod \gcd(q_1, q_2)$ becomes trivially learnable.

In Section 3, we present several reductions between LAM and LWE. In particular, we show in Theorem 22 that under certain parameters, the LAM problem is as hard as LWE. The techniques used in our reductions are inspired by the reductions between LWR and LWE in [BPR12, BGM+16].

Theorem 4 (Theorem 22, informal). *Let n, m, q_1, q_2 and B be integers such that $q_1 > 2Bmq_2$, D_B be a B-bounded and balanced distribution. If there is a randomized $\mathsf{poly}(n)$-time algorithm Learn that solves search-LAM_{n,m,q_1,q_2} with probability ϵ, then there is a $\mathsf{poly}(n)$-time algorithm Learn' that solves search-LWE_{n,m,q_1,D_B} with probability $\Omega(\epsilon^2)$.*

We also provide reductions from search to decisional LAM, from LAM to LWE, and others, which shows that LWE and LAM are as hard as each other under certain parameter regimes. Please find more details in Section 3.

It should be noted that most of our reductions, including Theorem 22 as shown above, require q_1 to be at least a polynomial in n. To understand the hardness of LAM over constant modulus q_1, we need to investigate the Arora-Ge algorithm and other algorithms, as detailed in our next two contributions.

Arora-Ge over Composite Moduli. Recall that the Arora-Ge algorithm [AG11] is efficient for solving LWE with prime modulus and small support.

Lemma 5 (Section 3 in [AG11]). *Let q be a prime and n be an integer. Let $\mathcal{D}_{\mathsf{noise}}$ be an error distribution whose support is of size $D < q$, and $\Pr[e = 0, e \leftarrow \mathcal{D}_{\mathsf{noise}}] = \frac{1}{\delta}$ for some $\delta > 1$. Let N be $\binom{n+D}{D}$ and C be a sufficiently large constant. Let $m := CN\delta q \log q/(q - D)$. The Arora-Ge algorithm solves $\mathsf{LWE}_{n,m,q,\mathcal{D}_{\mathsf{noise}}}$ in time $\mathsf{poly}(m)$ with overwhelming probability.*

Let us make three remarks on Lemma 5. First, it applies as long as $D < q$, meaning that the size of possible errors can be as large as $q-1$. Second, in [AG11] the sample complexity was written as $m = CN\delta q \log q$. However, a careful examination of their proof reveals that in their application of the Schwartz-Zippel lemma to the proof, they simplified the lower bound $\geq 1 - D/q$ to $\geq 1/q$. By retaining the original bound $\geq 1 - D/q$ instead of performing this relaxation, one can directly recover the tighter sample complexity stated in Lemma 5. Third, since the Schwartz-Zippel lemma only applies to prime fields, their analysis does not apply to composite moduli. To the best of our knowledge, the behavior of the Arora-Ge algorithm over LWE with composite moduli has never been published before.

We start by analyzing the Arora-Ge algorithm over prime power rings. Our result is as follows.

Lemma 6 (Lemma 31, informal). *Let $q = p^\kappa$, where p is prime and κ is some positive integer. Let $d \in [1, q)$ be an integer. Let $\sigma \in (0, 1/d]$, and $\chi_{d,\sigma}$ be any distribution on $[d]$ such that for any $x^* \in [d]$, $\Pr_{x \leftarrow \chi_{d,\sigma}}[x = x^*] \geq \sigma$. Let n, m, N be positive integers such that $N = \binom{n+d}{n}$, $m > 10N(\log q)/\sigma$. Given an instance of $LWE_{n,m,q,\chi_{d,\sigma}}$ with secret $\mathbf{s}$, the Arora-Ge algorithm outputs $\mathbf{s} \bmod q/\gcd(d!, q)$ in time $\mathsf{poly}(m)$ with overwhelming probability.*

To prove Lemma 6, we have identified favorable linear algebraic properties of the Arora-Ge polynomials, which enable us to avoid Schwartz-Zippel and use a more direct analytic approach. A proof sketch (of a concrete example) is given in Subsection 1.2. This approach not only extends the Arora-Ge algorithm to prime-power rings, but also reduces sample complexity by a multiplicative factor of $q/(q-d)$ when q is prime. The factor of $q/(q-d)$ can be super-constant when both q and d are super-constant.

By Lemma 6, when $gcd(d!, q) = 1$ (including the case when q is prime), the algorithm *fully recovers* $\mathbf{s}$, which is consistent with the result in [AG11]; when $gcd(d!, q) = q$, the algorithm *gains*

no information about $\mathbf{s}$, since the polynomials built in the Arora-Ge algorithm will always be 0; when $1 < gcd(d!, q) < q$, the algorithm *obtains partial information* about $\mathbf{s}$. Actually, we can do even better in the last case. We design an algorithm that uses the recursive application of Arora-Ge plus CRT, which can *fully recover* $\mathbf{s}$ as long as $gcd(d!, q) < q$, i.e., $d! \bmod q \neq 0$.

Theorem 7 (Theorem 30, informal)**.** *Let $q = p_1^{\kappa_1} p_2^{\kappa_2} \dots p_\ell^{\kappa_\ell}$, where $p_1, \dots, p_\ell$ are distinct primes and $\kappa_1, \kappa_2, \dots, \kappa_\ell \in \mathbb{N}^+$. Let d be a positive integer such that $d! \bmod q \neq 0$. Let $\sigma \in (0, 1/d]$, and $\chi_{d,\sigma}$ be any distribution on $[d]$ such that for any $x^* \in [d]$, $\Pr_{x \leftarrow \chi_{d,\sigma}}[x = x^*] \geq \sigma$. Let n, m, N be positive integers such that $N = \binom{n+d}{n}$, $m > 10N(\log q)/\sigma$. There is an algorithm running in time $\mathsf{poly}(m)$ that solves $LWE_{n,m,q,\chi_{d,\sigma}}$ with overwhelming probability. In particular, when d is constant and σ is non-negligible in n, the algorithm runs in $\mathsf{poly}(n)$ time.*

Let us remark that our algorithm also applies to LAM and LWR. Concretely, since $\mathrm{LAM}_{n,m,q_1,q_2}$ and $\mathrm{LWR}_{n,m,q_1,q_2}$ can both be reduced to LWE over modulus q_1 and error support $[\lfloor q_1/q_2 \rfloor]$, the recursive Arora-Ge algorithm applies as long as $\lfloor q_1/q_2 \rfloor! \bmod q \neq 0$. To illustrate this result, we provide several examples in Table 1. The first column lists selected constant values of q_1, and the second column gives the smallest d for which Arora-Ge fails to apply to LWE over modulus q_1 and error support $[d]$ (i.e. the smallest d such that $d!/q_1 \in \mathbb{Z}$). The third and fourth columns show the largest q_2 values for which Arora-Ge does not apply to $\mathrm{LAM}_{n,m,q_1,q_2}$ and $\mathrm{LWR}_{n,m,q_1,q_2}$, respectively. Note that the LAM problem has an additional constraint $gcd(q_1, q_2) = 1$. Consequently, the largest secure q_2 value against the Arora-Ge attack is typically smaller for LAM than for LWR with the same q_1.

q_1	smallest d of LWE	largest q_2 of LAM	largest q_2 of LWR
4	4	1	1
8	4	1	2
16	6	1	2
32	8	3	4
9	6	1	1
27	9	2	3

q_1	smallest d of LWE	largest q_2 of LAM	largest q_2 of LWR
81	9	8	9
243	12	20	20
6	3	1	2
15	5	2	3
20	5	3	4
24	4	5	6

Table 1: Secure parameter choices of LWE, LAM and LWR with constant moduli under the Arora-Ge attack.

Before we move on to our next contribution, let us briefly discuss other related works of Arora-Ge. Among the works [ACF+15, STA20, Ste24, NMSÜ25] following the Arora-Ge algorithm [AG11], only two have developed optimized attacks on LWE with prime moduli without relying on any heuristic assumptions. The first is by Steiner [Ste24], who shows that for any LWE polynomial system (with $m > n$), there exists a Gröbner basis algorithm that takes exponential time and memory. The second is from Noval et al. [NMSÜ25], whose attack has a time complexity independent of the modulus q, but only works when $q = \Omega(\binom{n+d}{d} \cdot d)$. Since we mainly focus on polynomial-time attacks on constant-moduli LWE in this work, we do not dive deep into the composite-moduli generalization of these two optimizations.

Polynomial time attack for wPRF candidates in $\mathsf{NC}^0[p]$. Next, we prove in Section 5 that no weak PRF exists in $\mathsf{NC}^0[p]$ with prime p. The attack is done in two steps. First, we show that all circuits in $\mathsf{NC}^0[p]$ with prime p can be computed by a constant-degree polynomial over $\mathbb{Z}_p$. Second, we show that all constant degree polynomials can be distinguished from random by a simple linear algebraic attack.

Note that for circuits in $\mathsf{NC}^0[q]$ where q has at least two distinct prime factors, we do not know how to compute them by low-degree polynomials in general. In fact, we show (in Theorem 43) that it is impossible to compute *some* circuits in $\mathsf{NC}^0[q]$ by low-degree polynomials in $\mathbb{Z}_N$ for any $N \geq 2$, even if $N = q$.

Looking ahead, we will provide candidate weak PRFs based on LAM and LWR over some constant modulus q_1, which are computable in $\mathsf{NC}^0[q_1]$. So our attack implies that when q_1 is a prime power, the candidates are *not* weak PRFs. When q_1 is not a prime power, our attack does not apply.

Candidate weak PRFs from LAM. Next, we present simple constructions of weak PRFs based on the hardness of decisional LAM. For any coprime integers q_1, q_2 with $q_1 > q_2 > 1$, consider the function family $\mathcal{F} := \left\{F_{\mathbf{s}} : \mathbb{Z}_{q_1}^n \to \mathbb{Z}_{q_2}\right\}_{\mathbf{s}\in\mathbb{Z}_{q_1}^n}$,

$$F_{\mathbf{s}}(\mathbf{x}) := (\langle \mathbf{s}, \mathbf{x}\rangle \bmod q_1) \bmod q_2. \tag{2}$$

If q_1/q_2 is super-polynomial in n, then $\mathcal{F}$ is a weak PRF assuming DLAM holds, and DLAM is as hard as DLWE with super-polynomial modulus/noise ratio when q_1/q_2 is super-polynomial. If $q_1/q_2 \in \mathsf{poly}(n)$, then $\mathcal{F}$ is not yet a weak PRF family since the output distribution is not statistically close to uniformly random over $\mathbb{Z}_{q_2}$. Nevertheless, we can sum up (over $\mathbb{Z}_p$ for some prime p that divides q_1) $\ell = O(n)$ independent copies of $\mathcal{F}$ to make the output indistinguishable from random, and get the weak PRF family $\mathcal{G} := \{g_{\mathbf{S}} : \mathbb{Z}_{q_1}^n \to \mathbb{Z}_p\}_{\mathbf{S}\in\mathbb{Z}_{q_1}^{n\times\ell}}$ as follows:

$$g_{\mathbf{S}}(\mathbf{x}) := \left(\sum_{i=1}^{\ell} ((\langle \mathbf{s}_i, \mathbf{x}\rangle \bmod q_1) \bmod q_2)\right) \bmod p. \tag{3}$$

What if the moduli q_1, q_2 are constants? In this way, $F_{\mathbf{s}}, g_{\mathbf{S}}$ can be evaluated by $\mathsf{NC}^0[q_1]$ circuits, and their security is based on the hardness of LAM with constant modulus. However, in this case, the security reductions from standard LWE do not apply, so we need to be careful about the parameter choice. First of all, q_1 should not be any prime power p^k, as otherwise $\mathcal{G}$ lies in the circuit class $\mathsf{NC}^0[p]$ where weak PRFs do not exist. In addition, to make $F_{\mathbf{s}}$ secure against the Arora-Ge attack, we should guarantee that $\lfloor q_1/q_2 \rfloor! \bmod q_1 = 0$. For the remaining choices of q_1, q_2, we are not aware of any other polynomial-time algorithms for solving $\mathrm{LAM}_{n,m,q_1,q_2}$ when q_1 is a constant.

To safely instantiate $g_{\mathbf{S}}$ with concrete parameters, we pick the rows with non-prime-power q_1 in Table 1. For example, let $q_1 = 24$, $q_2 = 5$, $p = 2$, and then we have

$$g_{\mathbf{S}}(\mathbf{x}) := \left(\sum_{i=1}^{\ell} ((\langle \mathbf{s}_i, \mathbf{x}\rangle \bmod 24) \bmod 5)\right) \bmod 2. \tag{4}$$

is a weak PRF candidate in $\mathsf{NC}^0[2,3]$ based on the conjectured security of $\mathrm{LAM}_{n,m,24,5}$.

Attentive readers may have already noticed that the $\mathsf{NC}^0[2,3]$ weak PRF candidate of Boneh et al. ([BIP+18]; see Eqn. (1)) can be formulated as a special case of $g_{\mathbf{S}}$ by setting $q_1 = 6, q_2 = 2, p = 3$, for $\mathbf{x} \in \{0,1\}^n$:

$$g_{\mathbf{S}}(\mathbf{x}) := \left(\sum_{i=1}^{\ell} ((\langle \mathbf{s}_i, \mathbf{x} \rangle \bmod 6) \bmod 2) \right) \bmod 3. \tag{5}$$

However, we note that the security of this specific instantiation cannot be directly based on the LAM assumption, since we inherently need $\gcd(q_1, q_2) = 1$, but here $\gcd(6,2) = 2$. In general, we have not been able to provide an assumption to capture the construction in Eqn. (1). Although our construction and the construction of Boneh et al. both use alternating moduli, we think their underlying reasoning of security are different in some perspectives.

We also analyze the effect of other typical attacks, for example, linear cryptanalysis, and the BKW attack [BKW03]. In particular, the BKW algorithm gives a $2^{O(n/\log n)}$ time attack to the LAM problem, since the LAM distribution is biased. However, our wPRF construction $g_{\mathbf{S}}$ tackles this by summing sufficiently many LAM samples, making the bias exponentially small, therefore the BKW attack does not apply to our wPRF candidate.

Candidate weak PRFs from LWR. Our construction of candidate weak PRFs from LWR mimics that from LAM (Eqn. (3)). That is, for integers $q_1, q_2, p > 1$ such that q_1 is not a prime power, $(\lfloor q_1/q_2 \rfloor)! \bmod q_1 = 0$, and p is a prime such that $p \mid q_1$, we define the function family $\mathcal{L} := \left\{ L_{\mathbf{S}} : \mathbb{Z}_{q_1}^n \to \mathbb{Z}_p \right\}_{\mathbf{S} \in \mathbb{Z}_{q_1}^{n \times \ell}}$ as follows:

$$L_{\mathbf{S}}(\mathbf{x}) := \left(\sum_{i=1}^{\ell} \lfloor (q_2/q_1) \cdot (\langle \mathbf{s}_i, \mathbf{x} \rangle \bmod q_1) \rceil \bmod q_2 \right) \bmod p. \tag{6}$$

An advantage of LWR is that it does not require the condition $gcd(q_1, q_2) = 1$. Furthermore, when q_2 is a divisor of q_1, the LWR distribution is not biased, so there is no need to do a sum of the samples. This enables a simpler candidate construction $\mathcal{K} := \{K_{\mathbf{s}} : \mathbb{Z}_{q_1}^n \to \mathbb{Z}_{q_2}\}_{\mathbf{s} \in \mathbb{Z}_{q_1}}$ as follows:

$$K_{\mathbf{s}}(\mathbf{x}) := \lfloor (q_2/q_1) \cdot (\langle \mathbf{s}, \mathbf{x} \rangle \bmod q_1) \rceil \bmod q_2 \tag{7}$$

In Remark 6.4 in the Eprint version of [BIP+18], Boneh et al. make some comments on weak PRFs related to LWR with constant composite moduli. We extend their analysis by taking into account the Arora-Ge attack over composite moduli and the polynomial attack for $\mathsf{NC}^0[p]$ circuits.

Comparison with existing works. In Table 2 we list some low-depth weak PRF candidates with conjectured security at least subexponential in the security parameter. In each of [BIP+18, BCG+20, BCG+21], the authors provide some main candidates along with several variants. Here we only include their main candidates. See Table 1 in [BCG+21] for a more detailed survey of weak and strong PRF candidates with possibly quasipolynomial security and in larger circuit classes. Note that the best attack we are aware of against our weak PRF candidates runs in $2^{O(n)}$ time, but we put $2^{O(n/\log n)}$ in the table since their underlying assumptions (LAM and LWR with constant coprime moduli) suffer from $2^{O(n/\log n)}$ time attacks (e.g., the BKW algorithm [BKW03]).

Let us also mention that some constructions in [BIP+18, BCG+20, BCG+21] are given with explicit low depths (e.g., the depths are as low as 2 or 3). When using a $\mathsf{NC}^0[q_1]$ (or $\mathsf{AC}^0[q_1]$) circuit, the exact depth of the weak PRF candidates $g_{\mathbf{S}}$ in Eqn. (3) and $L_{\mathbf{S}}$ in Eqn. (6) is 3 (when q_1 is even and $p = 2$, the depth can be reduced to 2), and that of $K_{\mathbf{s}}$ in Eqn. (7) is 2. The depth depends on whether we are using MOD_{q_1} gates or MOD_p gates for smaller prime p. For example, if $q_1 = p_1p_2$ and we compute the weak PRF candidates in $\mathsf{NC}^0[p_1, p_2]$, the depth would increase by 1.

Reference	Circuit Class	Hardness	Assumption
[BIP+18, Sec 3.1]	$\mathsf{NC}^0[p_1, p_2] = \mathsf{CC}^0[p_1, p_2]$	$2^{O(n)}$	Heuristic
[BIP+18, Sec 6.5]	$\mathsf{NC}^0[2, 3] = \mathsf{CC}^0[2, 3]$	$2^{O(n/\log n)}$	LWR with constant moduli
[BCG+20, Sec. 1.2]	$\mathsf{AC}^0[2]$ (XOR Normal Form)	$2^{\tilde{O}(n^{1/3})}$	Variable-Density LPN
[BCG+21, Sec. 3.1]	$\mathsf{AC}^0[2]$ (Sparse $\mathbb{F}_2$ polynomials)	$2^{\tilde{O}(\sqrt{n})}$	Heuristic
Ours (3)	$\mathsf{NC}^0[p_1, p_2] = \mathsf{CC}^0[p_1, p_2]$	$2^{O(n/\log n)}$	LAM with constant moduli
Ours (6),(7)	$\mathsf{NC}^0[p_1, p_2] = \mathsf{CC}^0[p_1, p_2]$	$2^{O(n/\log n)}$	LWR with constant moduli

Table 2: Low-depth weak PRF candidates. Here p_1, p_2 are two arbitrary distinct primes.

1.2 A concrete example of Arora-Ge with prime power moduli

Here we provide an overview of our analysis of the Arora-Ge algorithm for solving LWE with prime power moduli (Lemma 6, Lemma 31) using a concrete example. Extending it to general composite moduli is easy using the Chinese Remainder Theorem. We consider a toy example of 1-dimensional LWE with modulus $q = 16$ and $d = 4$. On input $\{(a_i, b_i := (a_is + e_i) \bmod 16)\}_{1\le i\le m}$, where m is sufficiently large, the Arora-Ge algorithm works as follows:

- For every $i \in [1, m]$, construct a polynomial $P_i(z) := (b_i - a_iz)(b_i - a_iz - 1)(b_i - a_iz - 2)(b_i - a_iz - 3)$.
- Compute the expansion form of $P_i(z)$: calculate the set of coefficients $\{c_j\}_{0\le j\le 4}$ such that $P_i(z) = c_0 + \sum_{j=1}^4 c_jz^j$.
- For every $1 \le j \le 4$, replace z^j by a new variable y_j, and we get a linear polynomial $P'_i(y_1, y_2, y_3, y_4) := c_0 + \sum_{j=1}^4 c_jy_j$.
- Solve the linear equation system $\{P'_i(y_1, y_2, y_3, y_4) \equiv 0 \pmod{16}\}_{1\le i\le m}$. Return the set of all possible y_1 values that can be a solution:

$$\mathcal{S} := \{y_1^* \mid \exists y_2^*, y_3^*, y_4^* \text{ s.t. } P'_i(y_1^*, y_2^*, y_3^*, y_4^*) \equiv 0 \pmod q \text{ for all } i = 1, ..., m\ \}$$

Since $P_i(z) \mid_{z=s} \equiv e_i(e_i-1)(e_i-2)(e_i-3) \equiv 0 \pmod{16}$, we have that $(y_1, y_2, y_3, y_4) = (s, s^2, s^3, s^4)$ is one solution of $P'_i(y_1, y_2, y_3, y_4) \equiv 0 \pmod{16}$. So $s \in \mathcal{S}$. This proves the completeness of the algorithm.

Our next goal is to prove that, with overwhelming probability over the randomness of $\{a_i, e_i\}_{1\le i\le m}$, it holds that $y_1^* \equiv s \pmod 2$ for every $y_1^* \in \mathcal{S}$, i.e., the Arora-Ge algorithm will recover $s \bmod 2$ correctly. To prove this, we only need to prove that other solutions will not make all equations

satisfiable. In fact, we will prove a stronger statement, that is, for every $y_1^* \not\equiv s \pmod 2$, every $y_2^*, y_3^*, y_4^* \in \mathbb{Z}_{16}$, and every $i \in [1, m]$,

$$\Pr_{a_i, e_i}[P_i'(y_1^*, y_2^*, y_3^*, y_4^*) \not\equiv 0 \pmod{16}] \geq \delta, \tag{8}$$

where δ is some constant in $(0, 0.1)$. Eqn. (8) implies that

$$\Pr_{\{a_i, e_i\}_{1 \leq i \leq m}}[\exists i \text{ s.t. } P_i'(y_1^*, y_2^*, y_3^*, y_4^*) \not\equiv 0 \pmod{16}] \geq 1 - (1-\delta)^m,$$

which is exactly what we want.

Now we continue to prove Eqn. (8). Consider

$$\begin{aligned} P_i(z) =& (b_i - a_i z)(b_i - a_i z - 1)(b_i - a_i z - 2)(b_i - a_i z - 3) \\ \equiv& (e_i + a_i(s-z))(e_i + a_i(s-z) - 1)(e_i + a_i(s-z) - 2)(e_i + a_i(s-z) - 3) \\ \equiv& (4e_i^3 - 18e_i^2 + 22e_i - 6)a_i(s-z) + (6e_i^2 - 18e + 11)a_i^2(s-z)^2 \\ &+ (4e_i - 6)a_i^3(s-z)^3 + a_i^4(s-z)^4. \pmod{16} \end{aligned}$$

For every $1 \leq j \leq 4$, we replace $(s-z)^j$ by a new variable $\tilde{y}_j$. We get a new linear polynomial $\tilde{P}_i'(\tilde{y}_1, \tilde{y}_2, \tilde{y}_3, \tilde{y}_4) := (4e_i^3 - 18e_i^2 + 22e_i - 6)a_i\tilde{y}_1 + (6e_i^2 - 18e + 11)a_i^2\tilde{y}_2 + (4e_i - 6)a_i^3\tilde{y}_3 + a_i^4\tilde{y}_4$. It can be proved that (Lemma 33) there is a bijection between the solution set of $P_i'(y_1, y_2, y_3, y_4) \equiv 0 \pmod{16}$ and that of $\tilde{P}_i'(\tilde{y}_1, \tilde{y}_2, \tilde{y}_3, \tilde{y}_4) \equiv 0 \pmod{16}$. Specifically, this bijection maps y_1^* to a $\tilde{y}_1^*$ such that $\tilde{y}_1^* = s - y_1$. So, to prove Eqn. (8), we need to prove for every $\tilde{y}_1^* \not\equiv 0 \pmod 2$, every $\tilde{y}_2^*, \tilde{y}_3^*, \tilde{y}_4^* \in \mathbb{Z}_{16}$, and every $i \in [1, m]$,

$$\Pr_{a_i, e_i}[\tilde{P}_i'(\tilde{y}_1^*, \tilde{y}_2^*, \tilde{y}_3^*, \tilde{y}_4^*) \not\equiv 0 \pmod{16}] \geq \delta.$$

So far, the algorithm and the proof idea are essentially the same with those of Arora-Ge. To continue, Arora-Ge uses the Schwartz-Zippel lemma, which does not immediately work with composite moduli. Our proof strategy is to use a simple linear algebraic argument.

For simplicity we assume that each $e_i \in \{0, 1, 2, 3\}$ is sampled with probability $1/4$, therefore we only need to prove

$$\Pr_{a_i}[\exists j \in \{0, 1, 2, 3\} \text{ s.t. } \tilde{P}_i'(\tilde{y}_1^*, \tilde{y}_2^*, \tilde{y}_3^*, \tilde{y}_4^*)|_{e_i = j} \not\equiv 0 \pmod{16}] \geq 4\delta.$$

By dropping $e_i = 0, 1, 2, 3$ in the coefficients of $\tilde{P}_i'$, and writing the equations in matrix form, it suffices to prove

$$\Pr_{a_i}\left[\begin{bmatrix} -6 & 11 & -6 & 1 \\ 2 & -1 & -2 & 1 \\ -2 & -1 & 2 & 1 \\ 6 & 11 & 6 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_i\tilde{y}_1^* \\ a_i^2\tilde{y}_2^* \\ a_i^3\tilde{y}_3^* \\ a_i^4\tilde{y}_4^* \end{bmatrix} \not\equiv \mathbf{0} \pmod{16}\right] \geq 4\delta.$$

Using gaussian elimination, our goal becomes proving

$$\Pr_{a_i}\left[\begin{bmatrix} 24 & 0 & 0 & 0 \\ 12 & 12 & 0 & 0 \\ 8 & 12 & 4 & 0 \\ 6 & 11 & 6 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_i\tilde{y}_1^* \\ a_i^2\tilde{y}_2^* \\ a_i^3\tilde{y}_3^* \\ a_i^4\tilde{y}_4^* \end{bmatrix} \not\equiv \mathbf{0} \pmod{16}\right] \geq 4\delta. \tag{9}$$

Since $\tilde{y}_1^* \not\equiv 0 \pmod 2$, we have that the inequality inside the probability holds as long as $a_i \not\equiv 0 \pmod 2$. Since a_i is chosen randomly, we have that the probability is greater than or equal to 0.5, which is greater than 4δ. This concludes the proof that the Arora-Ge algorithm recovers $s \bmod 2$.

Then, to recover the entire $s \in \mathbb{Z}_{16}$, we let $s' := (s - (s \bmod 2))/2 \in \mathbb{Z}_8$, and convert the original LWE instance into a new one with secret s', modulus $q' = 8$, and error upper bound $d' = 2$. Using Lemma 6 again, we can recover $s \bmod 4$ by running the Arora-Ge algorithm over $\mathbb{Z}_8$. This helps remove the error from this (new) LWE instance (since the error is less than $d' = 2$), and then we can recover the entire s' by Gaussian elimination.

Looking ahead, the number 24 in Eqn (9) corresponds to $d!$. The general statements for $n > 1$ and the general prime power moduli follow the same proof method.

2 Preliminary

Let $\mathbb{C}, \mathbb{R}, \mathbb{Q}, \mathbb{Z}, \mathbb{N}$ be the set of complex numbers, real numbers, rational numbers, integers, and natural numbers (non-negative integers). Let $\mathbb{R}^+, \mathbb{N}^+$ denote positive reals and integers. For any integer $q \geq 2$, denote $\mathbb{Z}/q\mathbb{Z}$ by $\mathbb{Z}_q$, and define $\mathbb{Z}_q^{n*} := \{\mathbf{x} \in \mathbb{Z}_q^n \mid \gcd(q, x_1, \ldots, x_n) = 1\}$. For any $d \in \mathbb{N}^+$, let $[d] := \{0, 1, \ldots, d-1\}$. For any $d \in \mathbb{N}^+$, let $\mathcal{U}_d$ denote the uniform distribution on $[d]$. For any finite set $\mathcal{S}$, let $\mathcal{U}(\mathcal{S})$ denote the uniform distribution on $\mathcal{S}$. The rounding operation $\lfloor a \rceil$ rounds a real number a to its nearest integer (if $a \in \mathbb{Z} + 0.5$, we round it to $a + 0.5$). We give "mod" lower precedence than addition/subtraction. For example, $a + b \bmod q = (a + b) \bmod q$.

A vector in $\mathbb{R}^n$ (represented in column form by default) is written as a bold lower-case letter, e.g. $\mathbf{v}$. For a vector $\mathbf{v}$, the i^{th} component of $\mathbf{v}$ will be denoted by v_i or $v(i)$ by default, unless v_i or $v(i)$ is defined for other meanings explicitly. A matrix is written as a bold capital letter, e.g. $\mathbf{A}$. The i^{th} column vector of $\mathbf{A}$ is denoted $\mathbf{a}_i$ by default, unless $\mathbf{a}_i$ is defined for other meanings explicitly. Let $\mathbf{I} := diag(1, 1, \ldots, 1)$ be the identity matrix. For any matrix $\mathbf{A}$, $\mathbf{A}^T$ denotes its transpose. If $\mathbf{A}$ is invertible, then $\mathbf{A}^{-1}$ denotes its inverse. For a distribution or a set $\mathcal{X}$, $x \leftarrow \mathcal{X}$ denotes sampling x according to the distribution or uniformly at random from $\mathcal{X}$.

The *binomial coefficient* $\binom{n}{k} := \frac{n!}{k!(n-k)!}$ represents the number of ways to choose k elements from a set of n elements. The *multinomial coefficient* $\binom{n}{k_1, k_2, \ldots, k_l} := \frac{n!}{\prod_{i=1}^{l} k_i!}$ generalizes this to partitioning n into l groups of sizes $k_1, k_2, \ldots, k_l$, and for some vector $\mathbf{k} \in \mathbb{N}^l$ such that $\|\mathbf{k}\|_1 = n$, we can abbreviate the definition as $\binom{n}{\mathbf{k}} := \frac{n!}{\prod_{i=1}^{l} k(i)!}$.

Definition 8 (Discrete Gaussian Distribution)**.** *The discrete Gaussian distribution with parameter $\alpha > 0$, denoted by $\mathcal{D}_\alpha$, is defined as the discrete distribution over $\mathbb{Z}$ with the probability density function $\mathcal{D}_\alpha(x) \propto \exp(-\pi x^2/\alpha^2)$.*

Definition 9 (Thresholded discrete distribution)**.** *Let χ be a discrete probability distribution with support $\mathcal{S}$. We say χ is c-thresholded, where $0 < c \leq 1/|\mathcal{S}|$, if for all $s \in \mathcal{S}$,*

$$\Pr_{x \leftarrow \chi}[x = s] \geq c.$$

Definition 10 (Bounded distribution)**.** *For any $B > 0$, a distribution $\mathcal{D}$ is called B-bounded if the support of $\mathcal{D}$ is a subset of $[-B, B]$.*

Definition 11 (Balanced distribution)**.** *A distribution $\mathcal{D}$ over $\mathbb{Z}$ is called balanced if* $\Pr_{x\leftarrow\mathcal{D}}[x \leq 0] \geq 1/2$ *and* $\Pr_{x\leftarrow\mathcal{D}}[x \geq 0] \geq 1/2$.

Definition 12 (Weak pseudorandom functions [NR04])**.** *Let $\mathcal{K}_n$, $\mathcal{X}_n$, $\mathcal{Y}_n$ be finite sets related to n. A function family $\mathcal{F} = \{\mathcal{F}_n = \{F_k : \mathcal{X}_n \to \mathcal{Y}_n\}_{k\in\mathcal{K}_n}\}_{n\in\mathbb{N}}$ is a weak PRF family if for any probabilistic polynomial time algorithm $\mathcal{A}$, there exists a negligible function ϵ such that for all $n \in \mathbb{N}$:*

$$|\Pr_{k\leftarrow\mathcal{K}_n}[\mathcal{A}^{F_k(\cdot)}(1^n) = 1] - \Pr_{R}[\mathcal{A}^{R(\cdot)}(1^n) = 1]| \leq \epsilon(n),$$

where $R : \mathcal{X}_n \to \mathcal{Y}_n$ is a truly random function, the adversary $\mathcal{A}$ is only allowed to query uniformly random inputs from $\mathcal{X}_n$.

The proofs of the following lemmas are postponed to Appendix C.1, C.2, C.3 and C.4 respectively.

Lemma 13. *For any prime p and positive integers n, κ, let $\mathbf{x} \in \mathbb{Z}_{p^\kappa}^n$ be any vector such that $\mathbf{x} \neq \mathbf{0}$. Then for any $b \in \mathbb{Z}_{p^\kappa}$*

$$\Pr_{\mathbf{a}\leftarrow\mathbb{Z}_{p^\kappa}^n}[\langle\mathbf{a}, \mathbf{x}\rangle \not\equiv b \pmod{p^\kappa}] \geq 1 - 1/p.$$

Lemma 14. *Let m, n, l be positive integers. Suppose $q = p_1^{\kappa_1} \dots p_\ell^{\kappa_\ell}$, where $p_1, \dots, p_\ell$ are ℓ distinct primes, $\kappa_1, \dots, \kappa_\ell$ are positive integers. Let $\mathbf{x} \in \mathbb{Z}_q^n$ be a fixed vector. Let $\mathbf{a}_1, \dots, \mathbf{a}_m \leftarrow \mathbb{Z}_q^n$ be m random vectors. Let $b_i := \langle\mathbf{a}_i, \mathbf{x}\rangle$ for all $1 \leq i \leq m$. We have*

$$\Pr_{\mathbf{a}_1,\dots,\mathbf{a}_m}[\exists\mathbf{x}' \in \mathbb{Z}_q^n,\ (\mathbf{x}' \neq \mathbf{x}) \wedge (\forall 1 \leq i \leq m,\ \langle\mathbf{a}_i, \mathbf{x}'\rangle = b_i)] \leq q^n/2^{\ell m}.$$

Lemma 15. *Let n, q be integers such that $n \geq 2$, $q \geq 2$. For $x_1, x_2, \cdots, x_n$ generated uniformly at random from $\mathbb{Z}_q$, we have*

$$\Pr_{x_1,x_2,\cdots,x_n}[\gcd(q, x_1, x_2, \cdots, x_n) \neq 1] < 1/(2^{n-1} - 1).$$

We have $|\mathbb{Z}_q^{n*}|/|\mathbb{Z}_q^n| > 1 - 1/(2^{n-1} - 1)$ as a corollary of Lemma 15.

Lemma 16. *Let $n \geq 1, p \geq 2, q \geq 2$ be positive integers. Let $k = \lceil q/p \rceil$. For some $c \in (0, 1/p]$, let $\mathcal{P}$ be a c-thresholded distribution with support $[p]$. Let the random variable $Y := \sum_{i=1}^n X_i \bmod q$, where each X_i is sampled from $\mathcal{P}$ independently. Let $\mathcal{Q}_n$ over $[q]$ be the distribution of Y. Then the statistical distance between $\mathcal{Q}_n$ and $\mathcal{U}_q$ is within $q \cdot (1 - c^k)^{\lfloor n/k \rfloor - 1}$.*

3 Hardness of Learning with Alternating Modulus

In this section, we present several reductions related to LAM, mainly for showing that LAM is as hard as LWE under certain parameters. In Subsection 3.1, we establish a search-to-decision reduction for LAM with binary secrets. Subsections 3.2 and 3.3 introduce reductions from search-LWE to search-LAM and from decisional-LWE to decisional-LAM, respectively. In Subsection 3.4, we show a reduction from search-LAM to search-LWE.

3.1 Search to Decision Reduction for LAM with Binary Secrets

For $q_1 = \mathsf{poly}(n)$, [BGM+16] demonstrates a search-to-decision reduction for LWR with binary secrets. The proof can be generalized to establish a search-to-decision reduction for any statistically injective polynomial function applied to $\mathbf{A}^T\mathbf{s}$, where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{s} \in \{0,1\}^n$. A function R over $\mathbb{Z}_q$ is said to be *statistically injective* if there exists a negligible function $\mathsf{negl}(n)$ such that

$$\Pr_{\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m},\mathbf{s}\leftarrow\{0,1\}^n}[\exists s' \in \{0,1\}^n \ s.t. \ \mathbf{s}' \neq \mathbf{s}, R(\mathbf{A}^T\mathbf{s}') = R(\mathbf{A}^T\mathbf{s})] \leq \mathsf{negl}(n),$$

where we abuse the notation by letting $R(\mathbf{y}) := (R(y(1)), R(y(2)), ..., R(y(m)))^T$ for any $y \in \mathbb{Z}_q^m$. The following theorem generalizes the search-to-decision reduction in [BGM+16].

Theorem 17. *For every $\epsilon > 0$, positive integers n, m, $q = \mathsf{poly}(n)$, statistically injective polynomial function R over $\mathbb{Z}_q$, and $\mathsf{poly}(n)$-time algorithm Dist such that*

$$|\Pr_{\mathbf{A},\mathbf{s}}[\mathsf{Dist}(\mathbf{A}, R(\mathbf{A}^T\mathbf{s})) = 1] - \Pr_{\mathbf{A}}[\mathsf{Dist}(\mathbf{A}, R(\mathcal{U}(\mathbb{Z}_q^m))) = 1]| \geq \epsilon,$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{s} \leftarrow \{0,1\}^n$, there exists a $\mathsf{poly}(n)$-time algorithm Learn and a negligible function $\mathsf{negl}(n)$ such that

$$\Pr_{\mathbf{A},\mathbf{s}}[\mathsf{Learn}(\mathbf{A}, R(\mathbf{A}^T\mathbf{s})) = \mathbf{s}] \geq \frac{\epsilon}{4qm} - \mathsf{negl}(n).$$

Before proving Theorem 17, we need two lemmas from [BGM+16] in the following.

Lemma 18. *For any ϵ, n, m, q, every $\mathsf{poly}(n)$-time computable function R over $\mathbb{Z}_q$, and $\mathsf{poly}(n)$-time algorithm Dist such that*

$$\Pr_{\mathbf{A},\mathbf{s}}[\mathsf{Dist}(\mathbf{A}, R(\mathbf{A}^T\mathbf{s})) = 1] - \Pr_{\mathbf{A}}[\mathsf{Dist}(\mathbf{A}, R(\mathcal{U}(\mathbb{Z}_q^m))) = 1] = \epsilon,$$

there exists a $\mathsf{poly}(n)$-time algorithm Pred such that

$$\Pr_{\mathbf{A},\mathbf{s},\mathbf{v}}[\mathsf{Pred}(\mathbf{A}, R(\mathbf{A}^T\mathbf{s}), \mathbf{v}) = \langle \mathbf{v}, \mathbf{s}\rangle] = \frac{1}{q} + \frac{\epsilon}{mq},$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{v} \leftarrow \mathbb{Z}_q^n$, and $\mathbf{s}$ is sampled from an arbitrary distribution over $\mathbb{Z}_q^n$.

Lemma 19. *Let $1/\epsilon, q$ be polynomial in n. There exists a $\mathsf{poly}(n)$-time oracle algorithm List such that for every algorithm Pred satisfying $|\Pr[\mathsf{Pred}(\mathbf{A}, R(\mathbf{A}^T\mathbf{s}), \mathbf{v}) = \langle \mathbf{v}, \mathbf{s}\rangle] - 1/q| \geq \epsilon$, $\mathsf{List}^{\mathsf{Pred}}$ outputs a list of entries $(q', \mathbf{s}')$ containing at least one entry such that $q' > 1$, q' divides q, and $\mathbf{s}' \equiv \mathbf{s} \pmod{q'}$, with probability at least $\epsilon/4$, where the probability is taken over $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, and an arbitrary distribution of $\mathbf{s}$ on $\mathbb{Z}_q^n$.*

Proof of Theorem 17. By Lemma 18, we take the distribution of $\mathbf{s}$ to be uniform on $\{0,1\}^n$. We know there exists a $\mathsf{poly}(n)$-time predictor Pred such that

$$|\Pr_{\substack{\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m}\\ \mathbf{s}\leftarrow\{0,1\}^n\\ \mathbf{v}\leftarrow\mathbb{Z}_q^n}}[\mathsf{Pred}(\mathbf{A}, R(\mathbf{A}^T\mathbf{s}), \mathbf{v}) = \langle \mathbf{v}, \mathbf{s}\rangle] - \frac{1}{q}| \geq \frac{\epsilon}{mq}.$$

By Lemma 19, there is a $\mathsf{poly}(n)$-time algorithm List, such that $\mathsf{List}^{\mathsf{Pred}}$ outputs a list of $(q', \mathbf{s}')$. With probability at least $\epsilon/(4mq)$, there is an entry $(q', \mathbf{s}')$ contained in the list, such that $q' > 1$, q' divides q, and $\mathbf{s}' \equiv \mathbf{s} \pmod{q'}$. Since $\mathbf{s} \in \{0,1\}^n$, $\mathbf{s}' \equiv \mathbf{s} \pmod{q'}$ implies $\mathbf{s}' = \mathbf{s}$.

Algorithm Learn simulates List on Pred, obtains a list of $(q', \mathbf{s}')$. For each $(q', \mathbf{s}')$ in the list, Learn tests whether $R(\mathbf{A}^T\mathbf{s}') = R(\mathbf{A}^T\mathbf{s})$ or not, and outputs $\mathbf{s}'$ if the condition is satisfied. Since R is statistically injective, we have $\mathbf{s}' = \mathbf{s}$ with overwhelming probability.

Concretely, the overall probability that Learn outputs the secret $\mathbf{s}$ satisfies:

$$\begin{aligned}
&\Pr_{\mathbf{A},\mathbf{s}}[\mathsf{Learn}(\mathbf{A}, R(\mathbf{A}^T\mathbf{s})) = \mathbf{s}] \\
\geq &\Pr_{\mathbf{A},\mathbf{s}}[\mathsf{Learn} \text{ outputs } \mathbf{s}' \text{ such that } R(\mathbf{A}^T\mathbf{s}') = R(\mathbf{A}^T\mathbf{s})] \\
&- \Pr_{\mathbf{A},\mathbf{s}}[\exists \mathbf{s}^* \neq \mathbf{s} \text{ such that } R(\mathbf{A}^T\mathbf{s}^*) = R(\mathbf{A}^T\mathbf{s})] \\
\geq &\Pr_{\mathbf{A},\mathbf{s}}[\mathsf{List} \text{ outputs } (q', \mathbf{s})] - \mathsf{negl}(n) \\
\geq &\frac{\epsilon}{4qm} - \mathsf{negl}(n).
\end{aligned}$$

Here the probability is taken over $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{s} \leftarrow \{0,1\}^n$. □

The following lemma shows the statistical injectivity of the LAM function.

Lemma 20. *Let q_1, q_2, n, m be positive integers such that $q_1 > 2q_2$, $q_1 \geq 15$, $q_2 \geq 3$, $\gcd(q_1, q_2) = 1$, and $m \geq 4n$. The function $f : \mathbb{Z}_{q_1}^m \mapsto \mathbb{Z}_{q_2}^m$ defined as*

$$f(x) = (x \bmod q_1) \bmod q_2$$

is a statistically injective function.

Proof. For any $\mathbf{t} \in \{0,1\}^n \setminus \{\mathbf{0}\}$, we know $\gcd(q_1, t_1, \cdots, t_n) = 1$. Thus, for $\mathbf{a} \leftarrow \mathbb{Z}_{q_1}^n$, $\langle \mathbf{a}, \mathbf{t} \rangle$ is distributed uniformly at random over $\mathbb{Z}_{q_1}$. For any $z \in \mathbb{Z}_{q_1}$ and $y \in \mathbb{Z}_{q_2}$, we have

$$\Pr_{\mathbf{a} \leftarrow \mathbb{Z}_{q_1}^n}[(\langle \mathbf{a}, \mathbf{s} \rangle \bmod q_1 - z) \bmod q_2 = y] \leq \frac{\lceil q_1/q_2 \rceil}{q_1} \leq \frac{q_1/q_2 + 1}{q_1} = \frac{1}{q_1} + \frac{1}{q_2} \leq 2/5.$$

For any $\mathbf{s}' \neq \mathbf{s}$, $\mathbf{a} \in \mathbb{Z}_{q_1}^n$,

$$f(\langle \mathbf{a}, \mathbf{s} \rangle) - f(\langle \mathbf{a}, \mathbf{s}' \rangle) = \big((\langle \mathbf{a}, \mathbf{s} \rangle - \langle \mathbf{a}, \mathbf{s}' \rangle) \bmod q_1 - b(\mathbf{a}, \mathbf{s}, \mathbf{s}')q_1\big) \bmod q_2,$$

here $b(\mathbf{a}, \mathbf{s}, \mathbf{s}') \in \{0,1\}$. Then

$$\begin{aligned}
&\Pr_{\mathbf{a} \leftarrow \mathbb{Z}_{q_1}^n}[f(\langle \mathbf{a}, \mathbf{s} \rangle) = f(\langle \mathbf{a}, \mathbf{s}' \rangle)] \\
= &\Pr_{\mathbf{a} \leftarrow \mathbb{Z}_{q_1}^n}[\big((\langle \mathbf{a}, \mathbf{s} \rangle - \langle \mathbf{a}, \mathbf{s}' \rangle) \bmod q_1 - b(\mathbf{a}, \mathbf{s}, \mathbf{s}')q_1\big) \bmod q_2 = 0] \\
\leq &\Pr_{\mathbf{a} \leftarrow \mathbb{Z}_{q_1}^n}[\big((\langle \mathbf{a}, \mathbf{s} \rangle - \langle \mathbf{a}, \mathbf{s}' \rangle) \bmod q_1 - q_1\big) \bmod q_2 = 0] \\
+ &\Pr_{\mathbf{a} \leftarrow \mathbb{Z}_{q_1}^n}[\big((\langle \mathbf{a}, \mathbf{s} \rangle - \langle \mathbf{a}, \mathbf{s}' \rangle) \bmod q_1\big) \bmod q_2 = 0] \\
\leq &2/5 + 2/5 = 4/5.
\end{aligned}$$

Therefore, for $\mathbf{A} \leftarrow \mathbb{Z}_{q_1}^{n\times m}$,

$$\Pr_{\mathbf{A}}[f(\mathbf{A}^T\mathbf{s}') = f(\mathbf{A}^T\mathbf{s})] \leq (4/5)^m \leq (256/625)^n.$$

Finally, by the union bound, we have

$$\Pr_{\mathbf{A}}[\exists \mathbf{s}' \neq \mathbf{s}, f(\mathbf{A}^T\mathbf{s}') = f(\mathbf{A}^T\mathbf{s})] \leq 2^n \cdot (256/625)^n = (512/625)^n,$$

which is exponentially small in n. □

By Theorem 17 and Lemma 20, we get the following corollary.

Corollary 21. *Let q_1, q_2, n, m be positive integers such that $m \geq 4n$, $q_1 > 2q_2$, $q_1 \geq 15$, $q_2 \geq 3$, and $\gcd(q_1, q_2) = 1$. If $DLAM_{n,m,q_1,q_2}$ with **binary secrets** is hard, then LAM_{n,m,q_1,q_2} with **binary secrets** is hard.*

3.2 Reduction from Search-LWE to Search-LAM

In this section, we prove search-LAM is at least as hard as search-LWE for certain parameters.

Theorem 22. *Let n, m, q_1, q_2 and B be positive integers such that $q_1 > 2Bmq_2$. Let D_B be a B-bounded and balanced distribution. If there exists a randomized $\mathsf{poly}(n)$-time algorithm Learn that solves LAM_{n,m,q_1,q_2} with probability ϵ, then there exists a $\mathsf{poly}(n)$-time algorithm Learn' that solves LWE_{n,m,q_1,D_B} with probability $\Omega(\epsilon^2)$, where the secret $\mathbf{s}$ is sampled from an arbitrary distribution over $\mathbb{Z}_{q_1}^{n*}$.*

Proof. Suppose there exists a randomized algorithm Learn that solves search-$\mathrm{LAM}_{n,m,q_1,q_2}$ with probability ϵ, i.e.,

$$\Pr_{r,\mathbf{A},\mathbf{s}}[\mathsf{Learn}(\mathbf{A}, (\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2, r) = \mathbf{s}] \geq \epsilon,$$

where $\mathbf{A} \leftarrow \mathbb{Z}_{q_1}^{n\times m}$, the secret $\mathbf{s}$ is sampled from an arbitrary distribution over $\mathbb{Z}_{q_1}^{n*}$, and r is the internal randomness of Learn.

For any LWE instance $(\mathbf{A}, \mathbf{b} = (\mathbf{A}^T\mathbf{s} + \mathbf{e}) \bmod q_1)$, where $\mathbf{e} \leftarrow D_B^m$, we define the following algorithm Learn':

$$\mathsf{Learn}'(\mathbf{A}, \mathbf{b}, r) = \mathsf{Learn}(q_2\mathbf{A} \bmod q_1, (q_2\mathbf{b} \bmod q_1) \bmod q_2, r).$$

Let $\mathbf{A}' := q_2\mathbf{A} \bmod q_1$. Then

$$(q_2\mathbf{b} \bmod q_1) \bmod q_2 = ((\mathbf{A}'^T\mathbf{s} + q_2\mathbf{e}) \bmod q_1) \bmod q_2.$$

Note that $\gcd(q_1, q_2) = 1$, which implies that $\mathbf{A}'$ is distributed uniformly in $\mathbb{Z}_{q_1}^{n\times m}$. Therefore, we have

$$\begin{aligned}
&\Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Learn}'(\mathbf{A}, \mathbf{b}, r) = \mathbf{s}] \\
&= \Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Learn}(\mathbf{A}', ((\mathbf{A}'^T\mathbf{s} + q_2\mathbf{e}) \bmod q_1) \bmod q_2, r) = \mathbf{s}] \\
&= \Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Learn}(\mathbf{A}, ((\mathbf{A}^T\mathbf{s} + q_2\mathbf{e}) \bmod q_1) \bmod q_2, r) = \mathbf{s}].
\end{aligned}$$

Let $\mathcal{E}$ denote the event that $(\mathbf{A}^T\mathbf{s} + q_2\mathbf{e}) \bmod q_1 \bmod q_2$ equals $\mathbf{A}^T\mathbf{s} \bmod q_1 \bmod q_2$, and let $\mathcal{E}_i$ denote the event that $(\langle \mathbf{a}_i, \mathbf{s}\rangle + q_2 e_i) \bmod q_1 \bmod q_2$ equals $\langle \mathbf{a}_i, \mathbf{s}\rangle \bmod q_1 \bmod q_2$. The event $\neg\mathcal{E}_i$ occurs only if $(\langle \mathbf{a}_i, \mathbf{s}\rangle \bmod q_1) + q_2 e_i < 0$ or $(\langle \mathbf{a}_i, \mathbf{s}\rangle \bmod q_1) + q_2 e_i \geq q_1$. For any fixed $\mathbf{A}, \mathbf{s}$, the following statements hold:

- If $Bq_2 \leq \langle \mathbf{a}_i, \mathbf{s}\rangle \bmod q_1 < q_1 - Bq_2$, then
$$\Pr_{e_i}[\mathcal{E}_i|\mathbf{A}, \mathbf{s}] = 1.$$
- If $\langle \mathbf{a}_i, \mathbf{s}\rangle \bmod q_1 < Bq_2$, then
$$\Pr_{e_i}[\mathcal{E}_i|\mathbf{A}, \mathbf{s}] \geq \Pr_{e_i}[e_i \geq 0] \geq 1/2.$$
- If $\langle \mathbf{a}_i, \mathbf{s}\rangle \bmod q_1 \geq q_1 - Bq_2$, then
$$\Pr_{e_i}[\mathcal{E}_i|\mathbf{A}, \mathbf{s}] \geq \Pr_{e_i}[e_i \leq 0] \geq 1/2.$$
- $\mathcal{E}_i$ only dependents on $\mathbf{a}_i$, that is,
$$\Pr_{e_i}[\mathcal{E}_i|\mathbf{A}, \mathbf{s}] = \Pr_{e_i}[\mathcal{E}_i|\mathbf{a}_i, \mathbf{s}].$$
- All $\mathcal{E}_i$ are pairwise independent, that is,
$$\Pr_{\mathbf{e}}[\mathcal{E}|\mathbf{A}, \mathbf{s}] = \prod_{i=1}^{m} \Pr_{e_i}[\mathcal{E}_i|\mathbf{A}, \mathbf{s}].$$

By Bayes' Formula, we have:

$$\begin{aligned}
&\Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Learn}'(\mathbf{A}, \mathbf{b}, r) = \mathbf{s}] \\
=& \Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Learn}(\mathbf{A}, ((\mathbf{A}^T\mathbf{s} + q_2\mathbf{e}) \bmod q_1) \bmod q_2, r) = \mathbf{s}|\mathcal{E}] \Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathcal{E}] \\
&+ \Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Learn}(\mathbf{A}, ((\mathbf{A}^T\mathbf{s} + q_2\mathbf{e}) \bmod q_1) \bmod q_2, r) = \mathbf{s}|\neg\mathcal{E}] \Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\neg\mathcal{E}] \\
\geq& \Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Learn}(\mathbf{A}, ((\mathbf{A}^T\mathbf{s} + q_2\mathbf{e}) \bmod q_1) \bmod q_2, r) = \mathbf{s}|\mathcal{E}] \Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathcal{E}] \\
=& \Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Learn}(\mathbf{A}, (\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2, r) = \mathbf{s}|\mathcal{E}] \Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathcal{E}] \\
=& \Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Learn}(\mathbf{A}, (\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2, r) = \mathbf{s} \wedge \mathcal{E}].
\end{aligned} \tag{10}$$

Since the domains of $r, \mathbf{A}, \mathbf{s}, \mathbf{e}$ are finite, we can express the probability as summations. For fixed $r, \mathbf{A}, \mathbf{s}$, the output of $\mathsf{Learn}(\mathbf{A}, (\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2, r)$ is fixed, indicating that the event

$\{\mathsf{Learn}(\mathbf{A}, (\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2, r) = \mathbf{s}\}$ is independent of $\mathbf{e}$.

$$
\begin{aligned}
&\Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Learn}(\mathbf{A}, (\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2, r) = \mathbf{s} \wedge \mathcal{E}] \\
=& \sum_{r^*,\mathbf{A}^*,\mathbf{s}^*} \Pr[r = r^*, \mathbf{A} = \mathbf{A}^*, \mathbf{s} = \mathbf{s}^*] \cdot \sum_{\mathbf{e}^*} \Pr[\mathbf{e} = \mathbf{e}^*, \mathsf{Learn}(\mathbf{A}^*, (\mathbf{A}^*\mathbf{s}^* \bmod q_1) \bmod q_2, r) = \mathbf{s}^* \wedge \mathcal{E}] \\
=& \sum_{r^*,\mathbf{A}^*,\mathbf{s}^*} \Pr[r = r^*, \mathbf{A} = \mathbf{A}^*, \mathbf{s} = \mathbf{s}^*, \mathsf{Learn}(\mathbf{A}^*, (\mathbf{A}^*\mathbf{s}^* \bmod q_1) \bmod q_2, r) = \mathbf{s}^*] \cdot \sum_{\mathbf{e}^*} \Pr[\mathbf{e} = \mathbf{e}^*][\mathcal{E}|\mathbf{A}^*, \mathbf{s}^*] \\
=& \sum_{r^*,\mathbf{A}^*,\mathbf{s}^*} \Pr[r = r^*, \mathbf{A} = \mathbf{A}^*, \mathbf{s} = \mathbf{s}^*, \mathsf{Learn}(\mathbf{A}^*, \mathbf{A}^*\mathbf{s}^* \bmod q_1) \bmod q_2, r) = \mathbf{s}^*] \cdot \Pr_{\mathbf{e}}[\mathcal{E}|\mathbf{A}^*, \mathbf{s}^*].
\end{aligned} \tag{11}
$$

By the Cauchy-Schwarz inequality, we have:

$$
\begin{aligned}
&(\sum_{r^*,\mathbf{A}^*,\mathbf{s}^*} \Pr[r = r^*, \mathbf{A} = \mathbf{A}^*, \mathbf{s} = \mathbf{s}^*, \mathsf{Learn}(\mathbf{A}^*, (\mathbf{A}^*\mathbf{s}^* \bmod q_1) \bmod q_2, r) = \mathbf{s}^*] \cdot \Pr_{\mathbf{e}}[\mathcal{E}|\mathbf{A}^*, \mathbf{s}^*]) \cdot \\
&(\sum_{r^*,\mathbf{A}^*,\mathbf{s}^*} \Pr[r = r^*, \mathbf{A} = \mathbf{A}^*, \mathbf{s} = \mathbf{s}^*, \mathsf{Learn}(\mathbf{A}^*, (\mathbf{A}^*\mathbf{s}^* \bmod q_1) \bmod q_2, r) = \mathbf{s}^*] / \Pr_{\mathbf{e}}[\mathcal{E}|\mathbf{A}^*, \mathbf{s}^*]) \\
\geq& (\sum_{r^*,\mathbf{A}^*,\mathbf{s}^*} \Pr[r = r^*, \mathbf{A} = \mathbf{A}^*, \mathbf{s} = \mathbf{s}^*, \mathsf{Learn}(\mathbf{A}^*, (\mathbf{A}^*\mathbf{s}^* \bmod q_1) \bmod q_2, r) = \mathbf{s}^*])^2.
\end{aligned} \tag{12}
$$

Note that:

$$
\begin{aligned}
&\sum_{r^*,\mathbf{A}^*,\mathbf{s}^*} \Pr[r = r^*, \mathbf{A} = \mathbf{A}^*, \mathbf{s} = \mathbf{s}^*, \mathsf{Learn}(\mathbf{A}^*, (\mathbf{A}^*\mathbf{s}^* \bmod q_1) \bmod q_2, r) = \mathbf{s}^*] / \Pr_{\mathbf{e}}[\mathcal{E}|\mathbf{A}^*, \mathbf{s}^*]) \\
\leq& \sum_{r^*,\mathbf{A}^*,\mathbf{s}^*} (\Pr[r = r^*, \mathbf{A} = \mathbf{A}^*, \mathbf{s} = \mathbf{s}^*] / \Pr_{\mathbf{e}}[\mathcal{E}|\mathbf{A}^*, \mathbf{s}^*]) \\
=& \sum_{\mathbf{A}^*,\mathbf{s}^*} \Pr[\mathbf{A} = \mathbf{A}^*, \mathbf{s} = \mathbf{s}^*] \frac{1}{\Pr_{\mathbf{e}}[\mathcal{E}|\mathbf{A}^*, \mathbf{s}^*]} \\
=& \sum_{\mathbf{A}^*,\mathbf{s}^*} \prod_{i=1}^{m} \frac{\Pr[\mathbf{a}_i = \mathbf{a}_i^*, \mathbf{s} = \mathbf{s}^*]}{\Pr_{e_i}[\mathcal{E}_i|\mathbf{A}^*, \mathbf{s}^*]} \\
=& \prod_{i=1}^{m} \sum_{\mathbf{a}_i^*,\mathbf{s}^*} \frac{\Pr[\mathbf{a}_i = \mathbf{a}_i^*, \mathbf{s} = \mathbf{s}^*]}{\Pr_{e_i}[\mathcal{E}_i|\mathbf{a}_i^*, \mathbf{s}^*]} \\
\leq& \prod_{i=1}^{m} (\Pr_{\mathbf{a}_i,\mathbf{s}}[Bq_2 \leq \langle \mathbf{a}_i, \mathbf{s} \rangle \bmod q_1 \leq q_1 - Bq_2] \\
&\quad + 2 \Pr_{\mathbf{a}_i,\mathbf{s}}[\langle \mathbf{a}_i, \mathbf{s} \rangle \bmod q_1 < Bq_2] + 2 \Pr_{\mathbf{a}_i,\mathbf{s}}[\langle \mathbf{a}_i, \mathbf{s} \rangle \bmod q_1 > q_1 - Bq_2]) \\
\leq_{(\clubsuit)}& (1 + 2Bq_2/q_1)^m \leq (1 + 1/m)^m \leq e,
\end{aligned} \tag{13}
$$

where $(\clubsuit)$ is derived as follows: For any $\mathbf{s} \neq \mathbf{0}$ such that $\gcd(q_1, s_1, \cdots, s_n) = 1$, $\langle \mathbf{a}_i, \mathbf{s} \rangle \bmod q_1$ distributes uniformly at random in $\mathbb{Z}_{q_1}$, so both $\Pr_{\mathbf{a}_i,\mathbf{s}}[\langle \mathbf{a}_i, \mathbf{s} \rangle \bmod q_1 < Bq_2]$ and $\Pr_{\mathbf{a}_i,\mathbf{s}}[\langle \mathbf{a}_i, \mathbf{s} \rangle \bmod$

$q_1 > q_1 - Bq_2]$ are at most Bq_2/q_1. We also note that:

$$\begin{aligned}&\sum_{r^*,\mathbf{A}^*,\mathbf{s}^*} \Pr[r = r^*, \mathbf{A} = \mathbf{A}^*, \mathbf{s} = \mathbf{s}^*, \mathsf{Learn}(\mathbf{A}^*, (\mathbf{A}^*\mathbf{s}^* \bmod q_1) \bmod q_2, r) = \mathbf{s}^*] \\ &= \Pr_{r,\mathbf{A},\mathbf{s}}[\mathsf{Learn}(\mathbf{A}, \mathbf{A}^T\mathbf{s} \bmod q_1 \bmod q_2, r) = \mathbf{s}] \geq \epsilon\end{aligned} \tag{14}$$

By Eqns. (12),(13),(14), we have:

$$\begin{aligned}&\sum_{r^*,\mathbf{A}^*,\mathbf{s}^*} \Pr[r = r^*, \mathbf{A} = \mathbf{A}^*, \mathbf{s} = \mathbf{s}^*, \mathsf{Learn}(\mathbf{A}^*, (\mathbf{A}^*\mathbf{s}^* \bmod q_1) \bmod q_2, r) = \mathbf{s}^*] \cdot \Pr_{\mathbf{e}}[\mathcal{E}|\mathbf{A}^*, \mathbf{s}^*] \\ \geq &\frac{(\sum_{r^*,\mathbf{A}^*,\mathbf{s}^*} \Pr[r = r^*, \mathbf{A} = \mathbf{A}^*, \mathbf{s} = \mathbf{s}^*, \mathsf{Learn}(\mathbf{A}^*, (\mathbf{A}^*\mathbf{s}^* \bmod q_1) \bmod q_2, r) = \mathbf{s}^*])^2}{\sum_{r^*,\mathbf{A}^*,\mathbf{s}^*} \Pr[r = r^*, \mathbf{A} = \mathbf{A}^*, \mathbf{s} = \mathbf{s}^*, \mathsf{Learn}(\mathbf{A}^*, (\mathbf{A}^*\mathbf{s}^* \bmod q_1) \bmod q_2, r) = \mathbf{s}^*] / \Pr_{\mathbf{e}}[\mathcal{E}|\mathbf{A}^*, \mathbf{s}^*])} \\ \geq &\epsilon^2/e\end{aligned} \tag{15}$$

Finally, by Eqns. (10),(11),(15), we have

$$\Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Learn}'(\mathbf{A}, \mathbf{b}, r) = \mathbf{s}] \geq \epsilon^2/e.$$

This completes the proof. □

3.3 Reduction from Decisional-LWE to Decisional-LAM

In this section, we give reductions from decisional-LWE to decisional-LAM under certain parameter settings. For $q_1 = \mathsf{poly}(n)$, Theorem 23 establishes an average-case (the secret $\mathbf{s} \leftarrow \mathcal{U}(\mathbb{Z}_{q_1}^{n*})$) reduction from decisional-LWE to decisional-LAM. For q_1 such that $q_1/(2Bmq_2)$ is super-polynomial in n, Theorem 24 establishes a worst-case (the secret $\mathbf{s}$ can be chosen from any distribution over $\mathbb{Z}_{q_1}^{n*}$) reduction from decisional-LWE to decisional-LAM.

Theorem 23. *Let q_1, q_2, n, m be integers and let α be a positive real value such that $m > 4n$, $2\alpha\sqrt{n}mq_2 < q_1 < \mathsf{poly}(n)$, $q_2 \geq 3$, and $\gcd(q_1, q_2) = 1$. Let $\mathcal{D}_\alpha$ be the discrete Gaussian distribution with parameter α. If there exists a randomized polynomial time algorithm Dist that solves $DLAM_{n,m,q_1,q_2}$ with uniform secrets with non-negligible probability, then there exists $c_1, c_2 > 0$, for $n' \leq n/\log q_1 - c_1 \log n, \alpha \geq \sqrt{c_2 \log n}$, there exists a polynomial time algorithm Dist' such that Dist' solves $DLWE_{n',m,q_1,\mathcal{D}_\alpha}$ with uniform secrets with non-negligible probability, where the secret $\mathbf{s}$ is sampled from the uniform distribution over $\mathbb{Z}_{q_1}^{n*}$.*

Theorem 24. *Let q_1, q_2, n, m, B be positive integers such that $2n < m \in \mathsf{poly}(n)$ and $\gcd(q_1, q_2) = 1$. Assume q_1 satisfies $q_1 > 2Bmq_2 \cdot T$, where T is super-polynomial in n. Let D_B be a B-bounded and balanced distribution. If there exists a $\mathsf{poly}(n)$-time algorithm Dist that solves $DLAM_{n,m,q_1,q_2}$ with probability non-negligible in n, then there exists a $\mathsf{poly}(n)$-time algorithm Dist' that solves $DLWE_{n,m,q_1,D_B}$ with probability non-negligible in n, where the secret $\mathbf{s}$ is sampled from an arbitrary distribution over $\mathbb{Z}_{q_1}^{n*}$.*

Proof of Theorem 23. When $q_1 < \mathsf{poly}(n)$, we can obtain the following chain of reductions:

uniform-secret decisional LWE $\leq_{(1)}$ binary-secret decisional LWE $\leq_{(2)}$ binary-secret search LWE $\leq_{(3)}$binary-secret search LAM $\leq_{(4)}$ binary-secret decisional LAM $\leq_{(5)}$ uniform-secret decisional LAM,

where "$\leq_{(1)}$" uses Lemma 28 ([BLP+13]), "$\leq_{(2)}$", "$\leq_{(3)}$" and "$\leq_{(4)}$" are gathered up in Lemma 25, and "$\leq_{(5)}$" is by Corollary 27. Note that for $x \leftarrow \mathcal{D}_\alpha$, we have $|x| > \alpha\sqrt{n}$ with probability $2^{-\Omega(n)}$. So $\mathcal{D}_\alpha$ can be treated as an $\alpha\sqrt{n}$-bounded and balanced distribution in the reductions. The lemmas used in our reduction are shown below. □

In the following we show Lemma 25, which establishes the reduction from DLWE to DLAM with secret chosen from any distribution over $\{0,1\}^n \backslash \{\mathbf{0}\}$. We emphasize that this lemma establishes a worst-case reduction, and Theorem 23 turns out to be an average-case reduction since Corollary 27 requires the secrets to be generated uniformly at random.

Lemma 25. *Let q_1, q_2, n, m, B be positive integers such that $m > 2n$, $2Bmq_2 < q_1 < \mathsf{poly}(n)$, and $\gcd(q_1, q_2) = 1$. Let D_B be a B-bounded and balanced distribution. If there exists a $\mathsf{poly}(n)$-time algorithm Dist that solves $DLAM_{n,m,q_1,q_2}$ with **binary secrets** with non-negligible probability, then there exists a $\mathsf{poly}(n)$-time algorithm Dist' that solves $DLWE_{n,m,q_1,D_B}$ with **binary secrets** with non-negligible probability, where the secret $\mathbf{s}$ is sampled from an arbitrary distribution over $\{0,1\}^n \setminus \{\mathbf{0}\}$.*

Proof. According to the conditions of the theorem, there exists a non-negligible value ϵ such that

$$|\Pr_{\mathbf{A},\mathbf{s}}[\mathsf{Dist}(\mathbf{A}, (\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2) = 1] - \Pr_{\mathbf{A}}[\mathsf{Dist}(\mathbf{A}, \mathcal{U}(\mathbb{Z}_{q_1}^m) \bmod q_2)) = 1]| \geq \epsilon,$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{s} \in \{0,1\}^n \backslash \{\mathbf{0}\}$. Since q_1 is polynomial in n, by Theorem 17, there exists a polynomial time algorithm Learn and a negligible function $\mathsf{negl}(n)$ such that

$$\Pr_{\mathbf{A},\mathbf{s}}[\mathsf{Learn}(\mathbf{A}, (\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2) = \mathbf{s}] \geq \frac{\epsilon}{4q_1 m} - \mathsf{negl}(n) =: \epsilon'.$$

By Theorem 22, there exists a polynomial time algorithm Learn' such that

$$\Pr_{\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Learn}'(\mathbf{A}, (\mathbf{A}^T\mathbf{s} + \mathbf{e}) \bmod q_1) = \mathbf{s}] \geq \epsilon'^2/e,$$

where $\mathbf{e} \leftarrow D_B^m$ is the error, e is the natural constant.

Consider the algorithm Dist': on input $(\mathbf{A}, \mathbf{b})$, it runs $\mathsf{Learn}'(\mathbf{A}, \mathbf{b})$ to get $\mathbf{s}$. If $\mathbf{e} = \mathbf{b} - \mathbf{A}^T\mathbf{s} \in D_B^m$ the algorithm outputs 1; otherwise it outputs 1 with probability 1/2. We have:

$$\Pr_{\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Dist}'(\mathbf{A}, \mathbf{b} = (\mathbf{A}^T\mathbf{s} + \mathbf{e}) \bmod q_1) = 1] \geq 1/2 + \epsilon'^2/(2e),$$

$$\Pr_{\mathbf{A}}[\mathsf{Dist}'(\mathbf{A}, \mathcal{U}(\mathbb{Z}_{q_1}^m)) = 1] = 1/2 + \frac{q_1^n}{(q_1 - 2B)^m} \leq 1/2 + \frac{q_1^n}{(q_1 - q_1/m)^m} \leq 1/2 + 4q_1^{n-m}.$$

Therefore,

$$\Pr_{\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Dist}'(\mathbf{A}, \mathbf{b} = (\mathbf{A}^T\mathbf{s} + \mathbf{e}) \bmod q_1) = 1] - \Pr_{\mathbf{A}}[\mathsf{Dist}'(\mathbf{A}, \mathcal{U}(\mathbb{Z}_{q_1}^m)) = 1] \geq \epsilon'^2/(2e) - 4q_1^{n-m}.$$

Since q_1 is polynomial in n, ϵ' is non-negligible, indicating that Dist' solves binary-secret decisional-LWE$_{n,m,q_1,D_B}$ with non-negligible probability. □

Next, we show the reduction from DLAM with binary secrets to DLAM with uniform secrets. We use the following lemma from [BGM+16].

Lemma 26. *Let S be any distribution supported on $\mathbb{Z}_q^{n*}$. For every function R on $\mathbb{Z}_q$, there is a polynomial-time transformation that (1) maps the distribution $(\mathbf{A}, R(\mathbf{A}^T\mathbf{s}))_{\mathbf{A}\leftarrow\mathbb{Z}_q^{m\times n},\mathbf{s}\leftarrow S}$ to $(\mathbf{A}, R(\mathbf{A}^T\mathbf{s}))_{\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m},\mathbf{s}\leftarrow\mathbb{Z}_q^{n*}}$ and (2) maps the distribution $(\mathbf{A}, R(\mathbf{u}))_{\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m},\mathbf{u}\leftarrow\mathbb{Z}_q^{m}}$ to itself.*

By setting S to $\{0,1\}^n\backslash\{\mathbf{0}\}$ and R to the LAM function, we immediately have the following corollary.

Corollary 27. *Let q_1, q_2, n, m be positive integers such that $q_1 > q_2$ and $\gcd(q_1, q_2) = 1$. If there exists a $\mathsf{poly}(n)$-time algorithm that solves decision(search)-LAM$_{n,m,q_1,q_2}$ for uniform secrets with non-negligible probability, then there exists a $\mathsf{poly}(n)$-time algorithm that solves decision(search)-LAM$_{n,m,q_1,q_2}$ for binary secrets with non-negligible probability.*

Finally, we present Lemma 28 from [BLP+13], which shows a reduction from DLWE with uniform secrets to DLWE with binary secrets.

Lemma 28. *Let n, m, q, k be positive integers, and let $\epsilon \in (0, 1/2)$, $\alpha, \delta > 0$, be such that $n \geq (k+1)\log_2 q + 2\log(1/\delta)$, $\alpha \geq \sqrt{\ln(2n(2+1/\epsilon))/\pi}$. Let $\mathcal{D}_\sigma$ be the discrete Gaussian distribution with parameter σ. If there exists an algorithm that solves $\mathsf{DLWE}_{n,m,q,\mathcal{D}_{\sqrt{10n}\alpha}}$ with binary secrets with advantage ζ, then there exists an algorithm that solves $\mathsf{DLWE}_{k,m,q,\mathcal{D}_\alpha}$ with uniform secrets with advantage at least*

$$(\zeta - \delta)/3m - 41\epsilon/2 - \sum_{p|q, p\ prime} p^{-k-1}.$$

By combining Corollary 21, Lemma 25, Corollary 27, and Lemma 28, we generalize the reduction from DLWE to DLAM with uniform secrets and $q_1 = \mathsf{poly}(n)$, which exactly proves Theorem 23. In the following, we continue to prove Theorem 24.

Proof of Theorem 24. Suppose there is a randomized algorithm Dist that solves decision-LAM$_{n,m,q_1,q_2}$ with non-negligible probability, then there exists a non-negligible function ϵ such that

$$|\Pr_{\mathbf{A},\mathbf{s}}[\mathsf{Dist}(\mathbf{A}, (\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2) = 1] - \Pr_{\mathbf{A}}[\mathsf{Dist}(\mathbf{A}, (\mathcal{U}(\mathbb{Z}_{q_1}^m) \bmod q_1) \bmod q_2) = 1]| \geq \epsilon,$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{s} \in \mathbb{Z}_q^{n*}$. Consider the following algorithm:

$$\mathsf{Dist}'(\mathbf{A}, \mathbf{b}) := \mathsf{Dist}(q_2\mathbf{A} \bmod q_1, (q_2\mathbf{b} \bmod q_1) \bmod q_2).$$

In the following, we prove that Dist' solves decision-LWE.

- When $\mathbf{b}$ is an LWE sample, i.e. $\mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e} \bmod q_1$ where $\mathbf{e} \leftarrow D_B^m$, we consider the following event:
$$(q_2(\mathbf{A}^T\mathbf{s} + \mathbf{e}) \bmod q_1) \bmod q_2 = (q_2\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2.$$
This event occurs if $B \leq (q_2\langle\mathbf{a}_i, \mathbf{s}\rangle) \bmod q_1 \leq q_1 B$ for all columns $\mathbf{a}_i$ of $\mathbf{A}$, which means
$$\begin{aligned}&\Pr_{\mathbf{A},\mathbf{s},\mathbf{e}}[(q_2\mathbf{b} \bmod q_1) \bmod q_2 = (q_2\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2]\\ \geq&(1 - 2B/q_1)^m \geq (1 - 1/(m \cdot T))^m \geq 1 - 1/T.\end{aligned}$$

Therefore, we have:

$$
\begin{aligned}
&\Pr_{\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Dist}'(\mathbf{A}, \mathbf{b} = (\mathbf{A}^T\mathbf{s} + \mathbf{e}) \bmod q_1) = 1] \\
&\geq \Pr_{\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Dist}'(\mathbf{A}, \mathbf{b}) = 1, (q_2\mathbf{b} \bmod q_1) \bmod q_2 = (q_2\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2] \\
&= \Pr_{\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Dist}(q_2\mathbf{A} \bmod q_1, (q_2\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2) = 1, (q_2\mathbf{b} \bmod q_1) \bmod q_2 = (q_2\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2] \\
&\geq \Pr_{\mathbf{A},\mathbf{s}}[\mathsf{Dist}(q_2\mathbf{A} \bmod q_1, (q_2\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2) = 1] \\
&\quad + \Pr_{\mathbf{A},\mathbf{s},\mathbf{e}}[(q_2\mathbf{b} \bmod q_1) \bmod q_2 = (q_2\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2] - 1 \\
&\geq \Pr_{\mathbf{A},\mathbf{s}}[\mathsf{Dist}(q_2\mathbf{A} \bmod q_1, (q_2\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2) = 1] - 1/T \\
&= \Pr_{\mathbf{A},\mathbf{s}}[\mathsf{Dist}(\mathbf{A}, (\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2) = 1] - 1/T.
\end{aligned}
$$

The last equation follows from the fact that $\gcd(q_1, q_2) = 1$, therefore $q_2\mathbf{A}$ is distributed uniformly in $\mathbb{Z}_{q_1}^{n\times m}$.

- When $\mathbf{b}$ is a uniform sample, we note that $\gcd(q_1, q_2) = 1$:

$$
\begin{aligned}
\Pr_{\mathbf{A}}[\mathsf{Dist}'(\mathbf{A}, \mathcal{U}(\mathbb{Z}_{q_1}^m)) = 1] &= \Pr_{\mathbf{A}}[\mathsf{Dist}(q_2\mathbf{A} \bmod q_1, (q_2\mathcal{U}(\mathbb{Z}_{q_1}^m) \bmod q_1) \bmod q_2)] \\
&= \Pr_{\mathbf{A}}[\mathsf{Dist}(\mathbf{A}, (\mathcal{U}(\mathbb{Z}_{q_1}^m) \bmod q_1) \bmod q_2)].
\end{aligned}
$$

Finally, we have the advantage of Dist' is:

$$
\begin{aligned}
&|\Pr_{\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Dist}'(\mathbf{A}, \mathbf{b} = (\mathbf{A}^T\mathbf{s} + \mathbf{e}) \bmod q_1) = 1] - \Pr_{\mathbf{A}}[\mathsf{Dist}'(\mathbf{A}, \mathcal{U}(\mathbb{Z}_{q_1}^m)) = 1]| \\
\geq &|\Pr_{\mathbf{A},\mathbf{s}}[\mathsf{Dist}(\mathbf{A}, (\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2) = 1] - \Pr_{\mathbf{A}}[\mathsf{Dist}(\mathbf{A}, (\mathcal{U}(\mathbb{Z}_{q_1}^m) \bmod q_1) \bmod q_2)]| - 1/T \\
\geq &\epsilon - 1/T.
\end{aligned}
$$

This completes the proof. □

3.4 Reduction from Search-LAM to Search-LWE

This section presents a reduction from the search-LAM problem to the search-LWE problem for specific parameter sets.

Theorem 29. *Let q_1, q_2, n, m be positive integers such that $q_1 > q_2$ and $\gcd(q_1, q_2) = 1$. If there exists a randomized $\mathsf{poly}(n)$-time algorithm Learn that solves $LWE_{n,m,q_1,D}$, where the support of D is over $[\lfloor q_1/q_2 \rceil + 1]$, with non-negligible probability, then there exists a randomized $\mathsf{poly}(n)$-time algorithm Learn' that solves LAM_{n,m,q_1,q_2} with non-negligible probability.*

Note that the reduction only guarantees that the LWE error obtained from the LAM sample is in a small support, while the error term is possibly dependent on $\mathbf{A}$ and $\mathbf{s}$. However, considering LWE where the error is in a small support but possibly dependent on $\mathbf{A}$ and $\mathbf{s}$ is still meaningful. For example, the Arora-Ge algorithm works even when the error is possibly dependent on $\mathbf{A}$ and $\mathbf{s}$.

Proof. Suppose there exists a randomized polynomial time algorithm Learn that solves $\mathrm{LWE}_{m,n,q_1,D}$ with non-negligible probability. Specifically, there is a non-negligible function ϵ such that

$$\Pr_{r,\mathbf{A},\mathbf{s},\mathbf{e}}[\mathsf{Learn}(\mathbf{A}, (\mathbf{A}^T\mathbf{s}+\mathbf{e}) \bmod q_1, r) = \mathbf{s}] \geq \epsilon,$$

where $\mathbf{A} \leftarrow \mathbb{Z}_{q_1}^{n\times m}$, $\mathbf{s} \in \mathbb{Z}_{q_1}^n$, $\mathbf{e} \leftarrow D^m$, and r is the internal randomness of Learn. For any LAM instance $(\mathbf{A}, \mathbf{b} = (\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2)$, there exists a vector $\mathbf{e} \in [\lfloor q_1/q_2 \rceil + 1]^m$ such that

$$\mathbf{b} \equiv \mathbf{A}^T\mathbf{s} - q_2\mathbf{e} \pmod{q_1}.$$

Let $(-q_2)^{-1}$ denote the multiplicative inverse of $-q_2$ in the ring $\mathbb{Z}_{q_1}$. We define the following algorithm Learn':

$$\mathsf{Learn}'(\mathbf{A}, \mathbf{b}, r) := \mathsf{Learn}(\mathbf{A}' := ((-q_2)^{-1}\mathbf{A}) \bmod q_1, \mathbf{b}' := ((-q_2)^{-1}\mathbf{b}) \bmod q_1, r),$$

where r is the internal randomness of Learn'. Since $\mathbf{b}' = (\mathbf{A}'^T\mathbf{s} + \mathbf{e}) \bmod q_1$, we have $(\mathbf{A}', \mathbf{b}')$ is an LWE instance with secret $\mathbf{s}$. So the algorithm outputs $\mathbf{s}$ with probability ϵ. This completes the proof. □

4 Arora-Ge Algorithm for LWE with Composite Modulus

In this section, we present a generalization of the Arora-Ge algorithm [AG11] for the LWE problem over composite moduli.

Theorem 30. *Let $q = p_1^{\kappa_1}p_2^{\kappa_2}\dots p_\ell^{\kappa_\ell}$, where $p_1, \dots, p_\ell$ are distinct primes and $\kappa_1, \kappa_2, \dots, \kappa_\ell \in \mathbb{N}^+$. Let $d \in [1, q)$ be an integer. Let $\chi_{d,\sigma}$ be a σ-thresholded distribution on $[d]$ for some $0 < \sigma \leq 1/d$. Let n, m, N be positive integers such that $N = \binom{n+d}{n}$, $m > 10N(\log q)/\sigma$. If $d! \bmod q \neq 0$, then there is an algorithm $\mathcal{A}$ that runs in time $\mathsf{poly}(m)$ and solves $LWE_{n,m,q,\chi_{d,\sigma}}$ with probability at least $1 - 2q^N \cdot (1-\sigma/2)^m - q^n/2^{m-1}$. Specifically, if d is constant and σ is non-negligible in n, then $\mathcal{A}$ runs in $\mathsf{poly}(n)$ time and solves $LWE_{n,m,q,\chi_{d,\sigma}}$ with overwhelming probability.*

In Subsection 4.1, we present the Arora–Ge algorithm for LWE with prime power moduli. In Subsection 4.2, we provide the analysis of the algorithm. In Subsection 4.3, we give a recursive algorithm for LWE with any composite moduli.

4.1 The Algorithm

In this subsection, we fix $q = p^\kappa$, where p is prime and κ is a positive integer. Let n, m, d be positive integers with $d < q$. For some $0 < \sigma \leq 1/d$, let $\chi_{d,\sigma}$ be a σ-thresholded distribution over $[d]$. Consider an instance $\{(\mathbf{a}_i, b_i = (\langle \mathbf{a}_i, \mathbf{s}\rangle + e_i) \bmod q)\}_{1\leq i\leq m}$ of $LWE_{n,m,q,\chi_{d,\sigma}}$. Since each $e_i \in [d]$, it follows that for every $1 \leq i \leq m$, $\prod_{\eta=0}^{d-1}\Big(b_i - \langle \mathbf{a}_i, \mathbf{s}\rangle - \eta\Big) \equiv 0 \pmod q$.

The Arora-Ge algorithm works as follows. For each sample $(\mathbf{a}_i, b_i)$, construct a polynomial over $\mathbb{Z}_q$:

$$P_i(\mathbf{z}) := P_i(z(1), \dots, z(n)) := \prod_{\eta=0}^{d-1}(b_i - \langle \mathbf{a}_i, \mathbf{z}\rangle - \eta) = \prod_{\eta=0}^{d-1}(b_i - \sum_{j=1}^{n} a_i(j)z(j) - \eta).$$

For any vector $\mathbf{v} \in \mathbb{N}^n$ with $\|\mathbf{v}\|_1 \leq d$, let $\mathbf{z}^{\mathbf{v}} := \prod_{i=1}^{n} z(i)^{v(i)}$. Then the expansion of $P_i(\mathbf{z})$ is a linear combination of these monomials $\mathbf{z}^{\mathbf{v}}$. By a standard combinatorial argument, the total number of distinct monomials $\mathbf{z}^{\mathbf{v}}$ is

$$N := \binom{d+n}{n} = O(n^d).$$

Hence, we *linearize* $P_i(\mathbf{z})$ by replacing each $\mathbf{z}^{\mathbf{v}}$ in its expansion with a new variable $y_{\mathbf{v}}$, thereby obtaining a linear equation in the variables $\{y_{\mathbf{v}}\}_{1 \leq \|\mathbf{v}\|_1 \leq d}$. For every $1 \leq j \leq d$, let $\mathbf{y}_{(j)}$ denote the vector formed by sorting $\{y_{\mathbf{v}}\}_{\|\mathbf{v}\|_1 = j}$ in reverse lexicographical order of $\mathbf{v}$, that is,

$$\begin{aligned}
\mathbf{y}_{(1)} &:= (y_{\boldsymbol{\delta}_1}, y_{\boldsymbol{\delta}_2}, y_{\boldsymbol{\delta}_3}, \ldots, y_{\boldsymbol{\delta}_n})^T \\
\mathbf{y}_{(2)} &:= (y_{2\boldsymbol{\delta}_1}, y_{\boldsymbol{\delta}_1+\boldsymbol{\delta}_2}, y_{\boldsymbol{\delta}_1+\boldsymbol{\delta}_3}, \ldots, y_{2\boldsymbol{\delta}_n})^T \\
&\ldots \\
\mathbf{y}_{(d)} &:= (y_{d\boldsymbol{\delta}_1}, y_{(d-1)\boldsymbol{\delta}_1+\boldsymbol{\delta}_2}, y_{(d-1)\boldsymbol{\delta}_1+\boldsymbol{\delta}_3}, \ldots, y_{d\boldsymbol{\delta}_n})^T,
\end{aligned}$$

where $\{\boldsymbol{\delta}_j\}_{1 \leq j \leq n}$ denotes the n-dimensional standard basis. Let $\mathbf{y} := (\mathbf{y}_{(1)}^T, \mathbf{y}_{(2)}^T, \ldots, \mathbf{y}_{(d)}^T)^T$. Denote the linearized equation by $P_i'(\mathbf{y})$. By querying the oracle sufficiently many times, we obtain enough linear equations of $\mathbf{y}$. Solving this system yields the secret $\mathbf{s} = \mathbf{y}_{(1)}$. (For details on solving a linear system of equations over a ring and truncating the solution vector, please refer to Appendix A.)

The formalized Arora-Ge Algorithm is presented in Algorithm 1. Since the algorithm involves only linear algebra of up to m dimensions, its time complexity is $\mathsf{poly}(m)$.

Algorithm 1 Arora-Ge Algorithm

function ARORA_GE($n, m, d, q, \{(\mathbf{a}_i, b_i)\}_{1 \leq i \leq m}$)
  for $i = 1$ **to** m **do**
    Construct the polynomial

$$P_i(\mathbf{z}) := \prod_{\eta=0}^{d-1} (b_i - \langle \mathbf{a}_i, \mathbf{z} \rangle - \eta) \tag{16}$$

    Expansion: compute $\{c_{i,\mathbf{v}}\}_{\mathbf{v}}$ s.t. $P_i(\mathbf{z}) = \sum_{\mathbf{v} \in \mathbb{N}^n, \|\mathbf{v}\|_1 \leq d} c_{i,\mathbf{v}} \mathbf{z}^{\mathbf{v}}$
    Linearization: construct the polynomial

$$P_i'(\mathbf{y}) = c_{i,\mathbf{0}} + \sum_{\mathbf{v} \in \mathbb{N}^n, 0 < \|\mathbf{v}\|_1 \leq d} c_{i,\mathbf{v}} y_{\mathbf{v}} \tag{17}$$

  end for
  Solve the system of linear equations $\{P_i'(\mathbf{y}) \equiv 0 \pmod q\}_{1 \leq i \leq m}$ and get the solution set $\mathcal{Y}$
  $\mathcal{S} \leftarrow \{\mathbf{y}_{(1)} \mid \exists \mathbf{y}_{(2)}, \ldots \mathbf{y}_{(d)} \text{ s.t. } (\mathbf{y}_{(1)}^T, \mathbf{y}_{(2)}^T, \ldots, \mathbf{y}_{(d)}^T)^T \in \mathcal{Y}\}$
  return $\mathcal{S}$
end function

4.2 The Solution Space of the Algorithm

In this subsection, we analyze the performance of the Arora-Ge algorithm on LWE with prime-power moduli. Concretely, we focus on the proof of Lemma 31, in which we analyze the output

space of *a single run* of Algorithm 1. This lemma is useful for designing the recursive algorithm in Subsection 4.3.

Lemma 31. *Let $q = p^\kappa$, where p is prime and κ is a positive integer. Let $d \in [1, q)$ be an integer. Let $\chi_{d,\sigma}$ be a σ-thresholded distribution on $[d]$ for some $0 < \sigma \le 1/d$. Let n, m, N be positive integers such that $N = \binom{n+d}{n}$, $m > 10N(\log q)/\sigma$. Let $\{(\mathbf{a}_i, b_i = (\langle \mathbf{a}_i, \mathbf{s}\rangle + e_i) \bmod q)\}_{1\le i\le m}$ be an instance of $LWE_{n,m,q,\chi_{d,\sigma}}$. Let*

$$\mathcal{S} \leftarrow \textsc{Arora_Ge}(n, m, d, q, \{(\mathbf{a}_i, b_i)\}_{1\le i\le m}).$$

Then

$$\Pr_{\{\mathbf{a}_i, e_i\}_i}[\exists \mathbf{s}' \in \mathcal{S} \text{ s.t. } \mathbf{s}' \not\equiv \mathbf{s} \pmod{q/\gcd(d!, q)}] < q^N \cdot (1 - \sigma(1 - 1/p))^m.$$

Furthermore, we derive the following corollary, which provides a precise characterization of the solution set generated by the Arora-Ge algorithm.

Corollary 32. *Let $q = p^\kappa$, where p is prime and κ is some positive integer. Let $d \in [1, q)$ be an integer. Let $\chi_{d,\sigma}$ be a σ-thresholded distribution on $[d]$ for some $0 < \sigma \le 1/d$. Let n, m be positive integers. Let $\{(\mathbf{a}_i, b_i = (\langle \mathbf{a}_i, \mathbf{s}\rangle + e_i) \bmod q)\}_{1\le i\le m}$ be an instance of $LWE_{n,m,q,\chi_{d,\sigma}}$, and let*

$$\mathcal{S}' := \{\mathbf{s}' \in \mathbb{Z}_q^n \mid \mathbf{s}' \equiv \mathbf{s} \pmod{q/\gcd(d!, q)}\}.$$

We have

$$\Pr[\mathcal{S}' \leftarrow \textsc{Arora_Ge}(n, m, d, q, \{(\mathbf{a}_i, b_i)\}_{1\le i\le m})] \ge 1 - q^N \cdot (1 - \sigma(1 - 1/p))^m.$$

The proof of Corollary 32 is deferred to Appendix C.6, and some examples of this result are illustrated in Table 3 in Appendix D. In the remainder of this section, we focus on proving Lemma 31.

Proof of Lemma 31. For any $1 \le i \le m$, let $P_i(\mathbf{z}), P_i'(\mathbf{y})$ be as defined in Eqns. (16), (17). We aim to prove that for any $\mathbf{y}^* = (\mathbf{y}_{(1)}^{*T}, \mathbf{y}_{(2)}^{*T}, \ldots, \mathbf{y}_{(d)}^{*T})^T \in \mathbb{Z}_q^N$ such that $\mathbf{y}_{(1)}^* \not\equiv \mathbf{s} \pmod{q/\gcd(d!, q)}$, we have

$$\Pr_{\mathbf{a}_i, e_i}[P_i'(\mathbf{y}^*) \equiv 0 \pmod q] \le 1 - \sigma(1 - 1/p). \tag{18}$$

If this inequality holds, we can conclude the following:

$$\begin{aligned}
&\Pr_{\{\mathbf{a}_i, e_i\}_i}[\exists \mathbf{s}' \in \mathcal{S} \text{ s.t. } \mathbf{s}' \not\equiv \mathbf{s} \pmod{q/\gcd(d!, q)}] \\
=&\Pr_{\{\mathbf{a}_i, e_i\}_i}[\exists \mathbf{y}^* = (\mathbf{y}_{(1)}^{*T}, \mathbf{y}_{(2)}^{*T}, \ldots, \mathbf{y}_{(d)}^{*T})^T \in \mathbb{Z}_q^N \text{ s.t. } \substack{\mathbf{y}_{(1)}^* \not\equiv \mathbf{s} \pmod{q/\gcd(d!, q)} \\ \wedge\ (\forall 1\le i\le m, P_i'(\mathbf{y}^*) \equiv 0 \pmod q)}] \\
\le& \sum_{\substack{\mathbf{y}^* = (\mathbf{y}_{(1)}^{*T}, \mathbf{y}_{(2)}^{*T}, \ldots, \mathbf{y}_{(d)}^{*T})^T \in \mathbb{Z}_q^N \\ \mathbf{y}_{(1)}^* \not\equiv \mathbf{s} \pmod{q/\gcd(d!, q)}}} \Pr_{\{\mathbf{a}_i, e_i\}_i}[\forall 1 \le i \le m, P_i'(\mathbf{y}^*) \equiv 0 \pmod q] \\
=& \sum_{\substack{\mathbf{y}^* = (\mathbf{y}_{(1)}^{*T}, \mathbf{y}_{(2)}^{*T}, \ldots, \mathbf{y}_{(d)}^{*T})^T \in \mathbb{Z}_q^N \\ \mathbf{y}_{(1)}^* \not\equiv \mathbf{s} \pmod{q/\gcd(d!, q)}}} \prod_{i=1}^m \Pr_{\mathbf{a}_i, e_i}[P_i'(\mathbf{y}^*) \equiv 0 \pmod q] \\
\le& \sum_{\substack{\mathbf{y}^* = (\mathbf{y}_{(1)}^{*T}, \mathbf{y}_{(2)}^{*T}, \ldots, \mathbf{y}_{(d)}^{*T})^T \in \mathbb{Z}_q^N \\ \mathbf{y}_{(1)}^* \not\equiv \mathbf{s} \pmod{q/\gcd(d!, q)}}} (1 - \sigma(1 - 1/p))^m \\
<& q^N \cdot (1 - \sigma(1 - 1/p))^m.
\end{aligned}$$

Thus, we only need to prove Inequality (18).

We start by noting the following equation:

$$P_i(\mathbf{z}) \equiv \prod_{\eta=0}^{d-1}(b_i - \langle \mathbf{a}_i, \mathbf{z}\rangle - \eta) \equiv \prod_{\eta=0}^{d-1}(\langle \mathbf{a}_i, \mathbf{s}-\mathbf{z}\rangle + e_i - \eta) \pmod q.$$

Next, define the polynomial

$$\tilde{P}_i(\tilde{\mathbf{z}}) := \prod_{\eta=0}^{d-1}(\langle \mathbf{a}_i, \tilde{\mathbf{z}}\rangle + b_i - \langle \mathbf{a}_i, \mathbf{s}\rangle - \eta) \equiv \prod_{\eta=0}^{d-1}(\langle \mathbf{a}_i, \tilde{\mathbf{z}}\rangle + e_i - \eta) \pmod q. \tag{19}$$

It follows that: $\tilde{P}_i(\tilde{\mathbf{z}})\big|_{\tilde{\mathbf{z}}=\mathbf{s}-\mathbf{z}} \equiv P_i(\mathbf{z}) \pmod q$.

To simplify notation, we define $\mathbf{x}^{\mathbf{v}} := \prod_{j=1}^{n} x(j)^{v(j)}$ for any vectors $\mathbf{x} \in \mathbb{Z}^n$ and $\mathbf{v} \in \mathbb{N}^n$ s.t. $\|\mathbf{v}\|_1 \le d$. Then, there exists a set of coefficients $\{\tilde{c}_{i,\mathbf{v}} \in \mathbb{Z}_q\}_{\mathbf{v}}$ such that

$$\tilde{P}_i(\tilde{\mathbf{z}}) \equiv \sum_{\mathbf{v}\in\mathbb{N}^n, \|\mathbf{v}\|_1 \le d} \tilde{c}_{i,\mathbf{v}} \tilde{\mathbf{z}}^{\mathbf{v}} \pmod q.$$

Next, we linearize $\tilde{P}_i(\tilde{\mathbf{z}})$ by replacing $\tilde{\mathbf{z}}^{\mathbf{v}}$ with a new variable $\tilde{y}_{\mathbf{v}}$, yielding

$$\tilde{P}'_i(\tilde{\mathbf{y}}) := \tilde{c}_{i,\mathbf{0}} + \sum_{\mathbf{v}\in\mathbb{N}^n, 0<\|\mathbf{v}\|_1 \le d} \tilde{c}_{i,\mathbf{v}} \tilde{y}_{\mathbf{v}}. \tag{20}$$

For every $1 \le j \le d$, let $\tilde{\mathbf{y}}_{(j)}$ denote the vector formed by sorting $\{\tilde{y}_{\mathbf{v}}\}_{\|\mathbf{v}\|_1=j}$ in reverse lexicographical order of $\mathbf{v}$. Let $\tilde{\mathbf{y}} := (\tilde{\mathbf{y}}_{(1)}^T, \tilde{\mathbf{y}}_{(2)}^T, \ldots, \tilde{\mathbf{y}}_{(d)}^T)^T$. Define the following two solution sets.

$$\mathcal{Y} := \{\mathbf{y}^* \in \mathbb{Z}_q^N \mid \forall 1 \le i \le m\ , P'_i(\mathbf{y}^*) \equiv 0 \pmod q\}. \tag{21}$$

$$\tilde{\mathcal{Y}} := \{\tilde{\mathbf{y}}^* \in \mathbb{Z}_q^N \mid \forall 1 \le i \le m\ , \tilde{P}'_i(\tilde{\mathbf{y}}^*) \equiv 0 \pmod q\}. \tag{22}$$

The following lemma provides a bijection between $\mathcal{Y}$ and $\tilde{\mathcal{Y}}$.

Lemma 33. *Let $\mathcal{Y}, \tilde{\mathcal{Y}}$ be as defined in Eqns. (21), (22). For a fixed secret $\mathbf{s} \in \mathbb{Z}_q^n$, there exists a bijection $F_{\mathbf{s}}$ between $\mathcal{Y}$ and $\tilde{\mathcal{Y}}$ such that, for any $\mathbf{y}^* = (\mathbf{y}_{(1)}^{*T}, \mathbf{y}_{(2)}^{*T}, \ldots, \mathbf{y}_{(d)}^{*T})^T$ and $\tilde{\mathbf{y}}^* := (\tilde{\mathbf{y}}_{(1)}^{*T}, \tilde{\mathbf{y}}_{(2)}^{*T}, \ldots, \tilde{\mathbf{y}}_{(d)}^{*T})^T$ satisfying $F_{\mathbf{s}}(\mathbf{y}^*) = \tilde{\mathbf{y}}^*$, we have $\tilde{\mathbf{y}}_{(1)}^* \equiv \mathbf{s} - \mathbf{y}_{(1)}^* \pmod q$.*

We postpone the proof of Lemma 33 to Appendix C.5, and continue proving Lemma 31. Recall that we only need to prove Inequality (18). Let $F_{\mathbf{s}}$ be the mapping defined in Lemma 33. Let $\tilde{\mathbf{y}}^* := (\tilde{\mathbf{y}}_{(1)}^{*T}, \tilde{\mathbf{y}}_{(2)}^{*T}, \ldots, \tilde{\mathbf{y}}_{(d)}^{*T})^T := F_{\mathbf{s}}(\mathbf{y}^*)$. Then we have

$$\tilde{\mathbf{y}}_{(1)}^* \equiv \mathbf{s} - \mathbf{y}_{(1)}^* \not\equiv \mathbf{0} \pmod{q/\gcd(d!, q)},$$

which implies $d! \cdot \tilde{\mathbf{y}}_{(1)}^* \not\equiv \mathbf{0} \pmod q$.

Applying Lemma 34 (stated below), we obtain

$$\Pr_{\mathbf{a}_i, e_i}[\tilde{P}'_i(\tilde{\mathbf{y}}^*) \bmod q \neq 0] \ge \sigma(1 - 1/p).$$

Using Lemma 33, we get

$$\Pr_{\mathbf{a}_i, e_i}[P_i'(\mathbf{y}^*) \bmod q \neq 0] \geq \sigma(1 - 1/p).$$

This completes the proof of the key inequality (18). Therefore, it completes the proof of Lemma 31. □

Lemma 34. *Assume* $\tilde{\mathbf{y}}^* := (\tilde{\mathbf{y}}_{(1)}^{*T}, \tilde{\mathbf{y}}_{(2)}^{*T}, \ldots, \tilde{\mathbf{y}}_{(d)}^{*T})^T := (\tilde{y}_{\boldsymbol{\delta}_1}^*, \tilde{y}_{\boldsymbol{\delta}_2}^*, \ldots, \tilde{y}_{d\boldsymbol{\delta}_n}^*)^T \in \mathbb{Z}_q^N$, *where* $N = \binom{n+d}{n}$. *If* $d! \cdot \tilde{\mathbf{y}}_{(1)}^* \not\equiv \mathbf{0} \pmod{q}$, *then for all* $1 \leq i \leq m$,

$$\Pr_{\mathbf{a}_i, e_i}[\tilde{P}_i'(\tilde{\mathbf{y}}^*) \bmod q \neq 0] \geq \sigma(1 - 1/p),$$

where the polynomial $\tilde{P}_i'(\tilde{\mathbf{y}})$ *is defined in Eqn.* (20).

Proof of Lemma 34. Let $\tilde{P}_i(\tilde{\mathbf{z}})$ be as defined in Eqn. (19). It expands as

$$\begin{aligned}
\tilde{P}_i(\tilde{\mathbf{z}}) &= \prod_{\eta=0}^{d-1}(\langle \mathbf{a}_i, \tilde{\mathbf{z}}\rangle + e_i - \eta) \\
&= \prod_{\eta=0}^{d-1}(e_i - \eta) + \sum_{k=1}^{d}\left(\sum_{\substack{0\leq\eta_1<\eta_2<\cdots<\eta_{d-k}\leq d-1 \\ \eta_1,\eta_2\ldots,\eta_{d-k}\neq e_i}} \prod_{j=1}^{d-k}(e_i - \eta_j)\right) \cdot \langle \mathbf{a}_i, \tilde{\mathbf{z}}\rangle^k \\
&= \prod_{\eta=0}^{d-1}(e_i - \eta) + \sum_{k=1}^{d}\left(\sum_{\substack{0\leq\eta_1<\eta_2<\cdots<\eta_{d-k}\leq d-1 \\ \eta_1,\eta_2\ldots,\eta_{d-k}\neq e_i}} \prod_{j=1}^{d-k}(e_i - \eta_j)\right) \cdot \sum_{\substack{\mathbf{v}\in\mathbb{N}^n \\ \|\mathbf{v}\|_1 = k}} \binom{k}{\mathbf{v}} \mathbf{a}_i^{\mathbf{v}} \tilde{\mathbf{z}}^{\mathbf{v}}.
\end{aligned}$$

Since $e_i \in [d]$, we have $\prod_{\eta=0}^{d-1}(e_i - \eta) = 0$. For any integer t, k, l such that $t \geq 1$, $0 \leq k < t$, $0 \leq l \leq t-1$, define

$$w_{t,k,l} := \sum_{\substack{0\leq\eta_1<\eta_2<\cdots<\eta_k\leq t-1 \\ \eta_1,\eta_2\ldots,\eta_k\neq l}} \prod_{j=1}^{k}(l - \eta_j). \tag{23}$$

Therefore, we can rewrite $\tilde{P}_i(\tilde{\mathbf{z}})$ as

$$\tilde{P}_i(\tilde{\mathbf{z}}) = \sum_{k=1}^{d} w_{d,d-k,e_i} \cdot \sum_{\substack{\mathbf{v}\in\mathbb{N}^n \\ \|\mathbf{v}\|_1 = k}} \binom{k}{\mathbf{v}} \mathbf{a}_i^{\mathbf{v}} \tilde{\mathbf{z}}^{\mathbf{v}},$$

which gives

$$\tilde{P}_i'(\tilde{\mathbf{y}}) = \sum_{k=1}^{d} w_{d,d-k,e_i} \cdot \sum_{\substack{\mathbf{v}\in\mathbb{N}^n \\ \|\mathbf{v}\|_1 = k}} \binom{k}{\mathbf{v}} \mathbf{a}_i^{\mathbf{v}} \tilde{y}_{\mathbf{v}}. \tag{24}$$

For any integer t, l such that $t \geq 1$ and $0 \leq l \leq t-1$, define

$$\boldsymbol{\omega}_{t,l} := (w_{t,t-1,l}, w_{t,t-2,l}, \ldots, w_{t,0,l})^T. \tag{25}$$

Additionally, define

$$\boldsymbol{\gamma}_i := (\sum_{\substack{\mathbf{v}\in\mathbb{N}^n\\ \|\mathbf{v}\|=1}} \binom{1}{\mathbf{v}} \mathbf{a}_i^{\mathbf{v}} \tilde{y}^*_{\mathbf{v}}, \sum_{\substack{\mathbf{v}\in\mathbb{N}^n\\ \|\mathbf{v}\|=2}} \binom{2}{\mathbf{v}} \mathbf{a}_i^{\mathbf{v}} \tilde{y}^*_{\mathbf{v}}, \ldots, \sum_{\substack{\mathbf{v}\in\mathbb{N}^n\\ \|\mathbf{v}\|=d}} \binom{d}{\mathbf{v}} \mathbf{a}_i^{\mathbf{v}} \tilde{y}^*_{\mathbf{v}})^T.$$

Then we have $\tilde{P}'_i(\tilde{\mathbf{y}}^*) = \langle \boldsymbol{\omega}_{d,e_i}, \boldsymbol{\gamma}_i \rangle$. Thus, the remaining task is to prove

$$\Pr_{\mathbf{a}_i, e_i}[\langle \boldsymbol{\omega}_{d,e_i}, \boldsymbol{\gamma}_i \rangle \bmod q \neq 0] \geq \sigma(1 - 1/p).$$

Since

$$\begin{aligned}
&\Pr_{\mathbf{a}_i, e_i}[\langle \boldsymbol{\omega}_{d,e_i}, \boldsymbol{\gamma}_i \rangle \bmod q \neq 0] \\
=&\sum_{l^*=0}^{d-1} \Pr_{\mathbf{a}_i \leftarrow \mathbb{Z}_q^n}[\langle \boldsymbol{\omega}_{d,e_i}, \boldsymbol{\gamma}_i \rangle \bmod q \neq 0 \mid e_i = l^*] \cdot \Pr_{e_i \leftarrow \chi_{d,\sigma}}[e_i = l^*] \\
\geq&\sigma \sum_{l^*=0}^{d-1} \Pr_{\mathbf{a}_i \leftarrow \mathbb{Z}_q^n}[\langle \boldsymbol{\omega}_{d,l^*}, \boldsymbol{\gamma}_i \rangle \bmod q \neq 0] \\
\geq&\sigma \Pr_{\mathbf{a}_i \leftarrow \mathbb{Z}_q^n}[\bigcup_{l^*=0}^{d-1} (\langle \boldsymbol{\omega}_{d,l^*}, \boldsymbol{\gamma}_i \rangle \bmod q \neq 0)] \\
=&\sigma \Pr_{\mathbf{a}_i \leftarrow \mathbb{Z}_q^n}[\exists l^* \in [d],\ \langle \boldsymbol{\omega}_{d,l^*}, \boldsymbol{\gamma}_i \rangle \bmod q \neq 0],
\end{aligned}$$

we just need to prove: $\Pr_{\mathbf{a}_i \leftarrow \mathbb{Z}_q^n}[\exists l^* \in [d],\ \langle \boldsymbol{\omega}_{d,l^*}, \boldsymbol{\gamma}_i \rangle \bmod q \neq 0] \geq 1 - 1/p$.

For any integer $t \geq 1$, define

$$\mathbf{W}_t := (\boldsymbol{\omega}_{t,0}, \boldsymbol{\omega}_{t,1}, \ldots, \boldsymbol{\omega}_{t,t-1})^T. \tag{26}$$

Then we only need to prove: $\Pr_{\mathbf{a}_i \leftarrow \mathbb{Z}_q^n}[\mathbf{W}_d \cdot \boldsymbol{\gamma}_i \bmod q \neq 0] \geq 1 - 1/p$.

By Proposition 35 (stated below), there exists a unimodular matrix $\mathbf{U}_d \in \mathbb{Z}^{d \times d}$ such that

$$\mathbf{U}_d \cdot \mathbf{W}_d = \begin{bmatrix} d! & 0 & 0 & 0 & \ldots & 0 \\ * & * & 0 & 0 & \ldots & 0 \\ * & * & * & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ * & * & \ldots & * & * & 0 \\ * & * & \ldots & * & * & * \end{bmatrix}.$$

Here the asterisks denote arbitrary entries. Then we have

$$\mathbf{U}_d \cdot \mathbf{W}_d \cdot \boldsymbol{\gamma}_i = (d! \cdot \gamma_i(1), *, *, \cdots, *)^T.$$

Since

$$\gamma_i(1) = \sum_{\substack{\mathbf{v}\in\mathbb{N}^n\\ \|\mathbf{v}\|=1}} \binom{1}{\mathbf{v}} \mathbf{a}_i^{\mathbf{v}} \tilde{y}^*_{\mathbf{v}} = \sum_{j=1}^{n} a_i(j) \tilde{y}^*_{\boldsymbol{\delta}_j} = \langle \mathbf{a}_i, \tilde{\mathbf{y}}^*_{(1)} \rangle,$$

we have

$$\begin{aligned}\Pr_{\mathbf{a}_i \leftarrow \mathbb{Z}_q^n}[\mathbf{W}_d \cdot \boldsymbol{\gamma}_i \bmod q \neq 0] &= \Pr_{\mathbf{a}_i \leftarrow \mathbb{Z}_q^n}[\mathbf{U}_d \cdot \mathbf{W}_d \cdot \boldsymbol{\gamma}_i \bmod q \neq 0] \\ &\geq \Pr_{\mathbf{a}_i \leftarrow \mathbb{Z}_q^n}[d! \cdot \gamma_i(1) \bmod q \neq 0] \\ &= \Pr_{\mathbf{a}_i \leftarrow \mathbb{Z}_q^n}[\langle \mathbf{a}_i, d! \cdot \tilde{\mathbf{y}}^*_{(1)} \rangle \bmod q \neq 0].\end{aligned}$$

Since we assume $d! \cdot \tilde{\mathbf{y}}_{(1)} \bmod q \neq 0$ in the statement of Lemma 34, we can apply Lemma 13, which gives: $\Pr_{\mathbf{a}_i \leftarrow \mathbb{Z}_q^n}[\langle \mathbf{a}_i, d! \cdot \tilde{\mathbf{y}}^*_{(1)} \rangle \bmod q \neq 0] \geq 1 - 1/p$. Therefore, $\Pr_{\mathbf{a}_i \leftarrow \mathbb{Z}_q^n}[\mathbf{W}_d \cdot \boldsymbol{\gamma}_i \bmod q \neq 0] \geq 1 - 1/p$. This concludes the proof of Lemma 34. □

Proposition 35. *For any integer t, k, l such that $t \geq 1$, $0 \leq k < t$, $0 \leq l \leq t-1$, recall from Eqns. (23), (25), (26) that $w_{t,k,l} := \sum_{\substack{0 \leq \eta_1 < \eta_2 < \cdots < \eta_k \leq t-1 \\ \eta_1, \eta_2 \ldots, \eta_k \neq l}} \prod_{j=1}^{k} (l - \eta_j)$, $\boldsymbol{\omega}_{t,l} := (w_{t,t-1,l}, w_{t,t-2,l}, \ldots, w_{t,0,l})^T$, $\mathbf{W}_t := (\boldsymbol{\omega}_{t,0}, \boldsymbol{\omega}_{t,1}, \ldots, \boldsymbol{\omega}_{t,t-1})^T$. Then there exists a unimodular matrix $\mathbf{U}_t \in \mathbb{Z}^{t \times t}$ such that*

$$\mathbf{U}_t \cdot \mathbf{W}_t = \begin{bmatrix} t! & 0 & 0 & 0 & \ldots & 0 \\ * & * & 0 & 0 & \ldots & 0 \\ * & * & * & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ * & * & \ldots & * & * & 0 \\ * & * & \ldots & * & * & * \end{bmatrix},$$

where the asterisks denote arbitrary entries.

Proof. We prove this by induction.

- Base case: When $t = 1$, $\mathbf{W}_1 = [w_{1,0,0}] = [1]$.
- Inductive step: Assume that for $t > 1$, there exists a unimodular matrix $\mathbf{U}_{t-1} \in \mathbb{Z}^{(t-1)\times(t-1)}$ such that

$$\mathbf{U}_{t-1} \cdot \mathbf{W}_{t-1} = \begin{bmatrix} (t-1)! & 0 & 0 & 0 & \ldots & 0 \\ * & * & 0 & 0 & \ldots & 0 \\ * & * & * & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ * & * & \ldots & * & * & 0 \\ * & * & \ldots & * & * & * \end{bmatrix}$$

Define a unimodular matrix

$$\mathbf{X}_t := \begin{bmatrix} -1 & 1 & 0 & 0 & \ldots & 0 \\ 0 & -1 & 1 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 & -1 & 1 \\ 0 & 0 & \ldots & 0 & 0 & 1 \end{bmatrix} \in \mathbb{Z}^{t \times t}.$$

By Proposition 36 (stated below), we have

$$\mathbf{X}_t \cdot \mathbf{W}_t = \begin{bmatrix} \boldsymbol{\omega}^T_{t,1} - \boldsymbol{\omega}^T_{t,0} \\ \boldsymbol{\omega}^T_{t,2} - \boldsymbol{\omega}^T_{t,1} \\ \ldots \\ \boldsymbol{\omega}^T_{t,t-1} - \boldsymbol{\omega}^T_{t,t-2} \\ \boldsymbol{\omega}^T_{t,t-1} \end{bmatrix} \overset{\text{Proposition 36}}{=} \begin{bmatrix} t \cdot \boldsymbol{\omega}^T_{t-1,0} \| 0 \\ t \cdot \boldsymbol{\omega}^T_{t-1,1} \| 0 \\ \ldots \\ t \cdot \boldsymbol{\omega}^T_{t-1,t-2} \| 0 \\ \boldsymbol{\omega}^T_{t,t-1} \end{bmatrix} = \begin{bmatrix} t \cdot \mathbf{W}_{t-1} \| \mathbf{0} \\ \boldsymbol{\omega}^T_{t,t-1} \end{bmatrix}.$$

Let $\mathbf{U}_t := \begin{bmatrix} \mathbf{U}_{t-1} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \cdot \mathbf{X}_t$. Then $\mathbf{U}_t$ is a unimodular matrix, and we have

$$\mathbf{U}_t \cdot \mathbf{W}_t = \begin{bmatrix} \mathbf{U}_{t-1} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \cdot \begin{bmatrix} t \cdot \mathbf{W}_{t-1} \| \mathbf{0} \\ \boldsymbol{\omega}_{t,t-1}^T \end{bmatrix} = \begin{bmatrix} t! & 0 & 0 & 0 & \dots & 0 \\ * & * & 0 & 0 & \dots & 0 \\ * & * & * & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ * & * & \dots & * & * & 0 \\ * & * & \dots & * & * & * \end{bmatrix}.$$

This completes the proof of Proposition 35. □

Proposition 36. *Adopting the notations in Proposition 35, for any $t \geq 2, 0 \leq l \leq t-2$, we have*

$$\boldsymbol{\omega}_{t,l+1}^T - \boldsymbol{\omega}_{t,l}^T = (t \cdot \boldsymbol{\omega}_{t-1,l}^T \| 0)$$

Proof. It holds that

$$\boldsymbol{\omega}_{t,l+1}^T - \boldsymbol{\omega}_{t,l}^T = (w_{t,t-1,l+1} - w_{t,t-1,l}, w_{t,t-2,l+1} - w_{t,t-2,l}, \dots, w_{t,0,l+1} - w_{t,0,l}).$$

Since $w_{t,0,l+1} = w_{t,0,l} = 1$, the remaining task is to prove

$$\forall 1 \leq k \leq t-1,\ w_{t,k,l+1} - w_{t,k,l} = t \cdot w_{t-1,k-1,l}.$$

First, we consider the term

$$w_{t,k,l+1} = \sum_{\substack{0 \leq \eta_1 < \eta_2 < \dots < \eta_k \leq t-1 \\ \eta_1, \eta_2 \dots, \eta_k \neq l+1}} \prod_{j=1}^{k} (l + 1 - \eta_j).$$

Since $l + 1 \neq 0$, we split the summation into two cases: one where $\eta_1 = 0$, and one where $\eta_1 \neq 0$. Thus, we have

$$\begin{aligned} w_{i,k,l+1} &= \sum_{\substack{0 = \eta_1 < \eta_2 < \dots < \eta_k \leq t-1 \\ \eta_1, \eta_2 \dots, \eta_k \neq l+1}} \prod_{j=1}^{k} (l + 1 - \eta_j) + \sum_{\substack{0 < \eta_1 < \eta_2 < \dots < \eta_k \leq t-1 \\ \eta_1, \eta_2 \dots, \eta_k \neq l+1}} \prod_{j=1}^{k} (l + 1 - \eta_j) \\ &\overset{(\spadesuit)}{=} (l+1) \cdot \sum_{\substack{0 < \eta_2 < \dots < \eta_k \leq t-1 \\ \eta_2 \dots, \eta_k \neq l+1}} \prod_{j=2}^{k} (l + 1 - \eta_j) + \sum_{\substack{0 < \eta_1 < \eta_2 < \dots < \eta_k \leq t-1 \\ \eta_1, \eta_2 \dots, \eta_k \neq l+1}} \prod_{j=1}^{k} (l + 1 - \eta_j) \\ &\overset{(\diamondsuit)}{=} (l+1) \cdot \sum_{\substack{0 \leq \eta_2 < \dots < \eta_k \leq t-2 \\ \eta_2 \dots, \eta_k \neq l}} \prod_{j=2}^{k} (l - \eta_j) + \sum_{\substack{0 \leq \eta_1 < \eta_2 < \dots < \eta_k \leq t-2 \\ \eta_1, \eta_2 \dots, \eta_k \neq l}} \prod_{j=1}^{k} (l - \eta_j) \\ &= (l+1) \cdot w_{t-1,k-1,l} + w_{t-1,k,l}, \end{aligned} \tag{27}$$

where in ($\spadesuit$) we factored out $(l + 1 - \eta_1) = (l + 1)$ from the first term, and in ($\diamondsuit$) we let we let η_j in ($\diamondsuit$) be $\eta_j - 1$ in ($\spadesuit$).

Next, we consider the term

$$w_{t,k,l} = \sum_{\substack{0\le\eta_1<\eta_2<\cdots<\eta_k\le t-1\\ \eta_1,\eta_2\ldots,\eta_k\neq l}} \prod_{j=1}^{k}(l-\eta_j).$$

Since $l \neq t-1$, we split the summation into two cases: one where $\eta_k = t-1$ and one where $\eta_k \neq t-1$. Using the same proof idea as in Eqn. (27), we obtain

$$\begin{aligned}
w_{t,k,l} &= \sum_{\substack{0\le\eta_1<\eta_2<\cdots<\eta_k<t-1\\ \eta_1,\eta_2\ldots,\eta_k\neq l}} \prod_{j=1}^{k}(l-\eta_j) + \sum_{\substack{0\le\eta_1<\eta_2<\cdots<\eta_k=t-1\\ \eta_1,\eta_2\ldots,\eta_k\neq l}} \prod_{j=1}^{k}(l-\eta_j) \\
&= \sum_{\substack{0\le\eta_1<\eta_2<\cdots<\eta_k<t-1\\ \eta_1,\eta_2\ldots,\eta_k\neq l}} \prod_{j=1}^{k}(l-\eta_j) + (l-(t-1))\cdot \sum_{\substack{0\le\eta_1<\eta_2<\cdots<\eta_{k-1}<t-1\\ \eta_1,\eta_2\ldots,\eta_{k-1}\neq l}} \prod_{j=1}^{k-1}(l-\eta_j) \qquad (28)\\
&= \sum_{\substack{0\le\eta_1<\cdots<\eta_{k-1}\le t-2\\ \eta_2\ldots,\eta_k\neq l}} \prod_{j=1}^{k-1}(l-\eta_j) + (l-(t-1))\cdot \sum_{\substack{0\le\eta_1<\eta_2<\cdots<\eta_{k-1}\le t-2\\ \eta_1,\eta_2\ldots,\eta_{k-1}\neq l}} \prod_{j=1}^{k-1}(l-\eta_j) \\
&= w_{t-1,k,l} + (l-(t-1))\cdot w_{t-1,k-1,l}
\end{aligned}$$

Subtracting Eqn. (28) from Eqn. (27) gives

$$w_{t,k,l+1} - w_{t,k,l} = t \cdot w_{t-1,k-1,l}.$$

This completes the proof. □

4.3 The Recursive Arora-Ge Algorithm

In the following lemma, we show how to recursively call the Arora-Ge algorithm to solve LWE with prime-power moduli.

Lemma 37. *Let $q = p^\kappa$, where p is prime and κ is some positive integer. Let $d \in [1, q)$ be an integer. Let $\chi_{d,\sigma}$ be a σ-thresholded distribution on $[d]$ for some $0 < \sigma \le 1/d$. Let n, m, N be positive integers such that $N = \binom{n+d}{n}$, $m > 10N(\log q)/\sigma$. When $d! \bmod q \neq 0$, there is a $\mathsf{poly}(m)$-time algorithm $\mathcal{A}$ such that, on inputting a random instance $\{(\mathbf{a}_i, b_i = (\langle \mathbf{a}_i, \mathbf{s}\rangle + e_i) \bmod q)\}_{1\le i\le m}$ of $LWE_{n,m,q,\chi_{d,\sigma}}$,*

$$\Pr_{\{(\mathbf{a}_i,b_i)\}_{1\le i\le m}}[\mathbf{s} \leftarrow \mathcal{A}(n, m, d, q, \{(\mathbf{a}_i, b_i)\}_{1\le i\le m})] > 1 - 2q^N \cdot (1-\sigma(1-1/p))^m - q^n/2^m.$$

Remark 38. *When $d! \bmod q = 0$, we have the polynomial $P_i(\mathbf{z})$ (defined in Eqn. (16)), which is a consecutive product of d integers, is always 0 modulo q (see Proposition 67 in Appendix C.6). So the condition $d! \bmod q \neq 0$ has reached the limit of the Arora-Ge algorithm.*

Remark 39. *This recursive algorithm is specifically required for solving search LWE problems. As for decision LWE, we can tell the LWE distribution from uniform by calling the Arora-Ge algorithm once, and checking whether the solution set is non-empty.*

Proof. Prove by induction on q.

Base case: When $q = p$, we aim to find an algorithm that, given any positive integer d such that $d! \bmod p \neq 0$, for an instance $\{(\mathbf{a}_i, b_i = (\langle \mathbf{a}_i, \mathbf{s}\rangle + e_i) \bmod p)\}_{1\leq i\leq m}$ of $LWE_{n,m,p,\chi_{d,\sigma}}$, outputs $\mathbf{s} \bmod p$ with probability at least $1 - 2p^N \cdot (1 - \sigma(1 - 1/p))^m$.

Since $d! \bmod p \neq 0$, we have $\gcd(d!, p) = 1$. Using Lemma 31, we can obtain $\mathbf{s} \bmod p$ with probability at least $1 - p^N \cdot (1 - \sigma(1 - 1/p))^m$ by directly applying Algorithm 1.

Inductive step: When q is composite, suppose that for any integers $\hat{q}$ such that $1 < \hat{q} < q$, $\hat{q} \mid q$, there exists an algorithm $\hat{\mathcal{A}}$ that, given any positive integer $\hat{d}$ such that $\hat{d}! \bmod \hat{q} \neq 0$, and any σ-thresholded distribution $\hat{\chi}$ on $[\hat{d}]$, for an instance $\{(\hat{\mathbf{a}}_i, \hat{b}_i = (\langle \hat{\mathbf{a}}_i, \hat{\mathbf{s}}\rangle + \hat{e}_i) \bmod \hat{q})\}_{1\leq i\leq m}$ of $LWE_{n,m,\hat{q},\hat{\chi}_{\hat{d},\sigma}}$, the following holds

$$\Pr[\hat{\mathbf{s}} \leftarrow \hat{\mathcal{A}}(n, m, \hat{d}, \hat{q}, \{(\hat{\mathbf{a}}_i, \hat{b}_i))\}_{1\leq i\leq m})] > 1 - 2\hat{q}^{\hat{N}} \cdot (1 - \sigma(1 - 1/p))^m - \hat{q}^n/2^m,$$

where $\hat{N} = \binom{n+\hat{d}}{n}$.

Our goal is to design an algorithm $\mathcal{A}$ such that

$$\Pr[\mathbf{s} \leftarrow \mathcal{A}(n, m, d, q, \{(\mathbf{a}_i, b_i)\}_{1\leq i\leq m})] > 1 - 2q^N \cdot (1 - \sigma(1 - 1/p))^m - q^n/2^m.$$

We design $\mathcal{A}$ as follows. It first calls

$$\mathcal{S} \leftarrow \textsc{Arora_Ge}(n, m, d, q, \{(\mathbf{a}_i, b_i)\}_{1\leq i\leq m}).$$

Let $q' := q/\gcd(d!, q)$. Since $d! \bmod q \neq 0$, we have $q' > 1$. Define the following event:

$$\mathcal{E}_1 : \text{ the event that } \forall \mathbf{s}' \in \mathcal{S},\ \mathbf{s}' \equiv \mathbf{s} \pmod{q'}.$$

By Lemma 31, we have

$$\Pr[\mathcal{E}_1] \geq 1 - q^N \cdot (1 - \sigma(1 - 1/p))^m. \tag{29}$$

If $\mathcal{E}_1$ does not happen, $\mathcal{A}$ fails. Otherwise, $\mathcal{A}$ continues to compute:

$$e'_i := (b_i - \langle \mathbf{a}_i, \mathbf{s}'\rangle) \bmod q' = (b_i - \langle \mathbf{a}_i, \mathbf{s}\rangle) \bmod q' = e_i \bmod q'.$$

- If $d \leq q'$, we have $e_i = (e_i \bmod q') = e'_i$. Then for every $1 \leq i \leq m$, we have the equation

  $$\langle \mathbf{a}_i, \mathbf{s}\rangle \equiv b_i - e_i \pmod{q},$$

  where $\mathbf{s}$ is the only unknown. Solving these linear equations yields a solution set $\mathcal{S}'$ (see Appendix A for details on solving linear equations over a ring). Let $\mathcal{E}_2$ denote the event that $\mathcal{S}' = \{\mathbf{s}\}$. By Lemma 14,

  $$\Pr_{\{\mathbf{a}_i\}_i}[\mathcal{E}_2] = 1 - \Pr_{\{\mathbf{a}_i\}_i}[\exists \mathbf{s}' \neq \mathbf{s} \text{ s.t. } \mathbf{s}' \in \mathcal{S}'] > 1 - q^n/2^m. \tag{30}$$

  If $\mathcal{E}_2$ does not happen, $\mathcal{A}$ fails. Otherwise, it manages to get $\mathbf{s}$. We have

  $$\begin{aligned}
  &\Pr[\mathbf{s} \leftarrow \mathcal{A}(n, m, d, q, \{(\mathbf{a}_i, b_i)\}_{1\leq i\leq m})] \\
  \geq &1 - \Pr[\neg\mathcal{E}_1] - \Pr[\neg\mathcal{E}_2] \\
  > &1 - q^N \cdot (1 - \sigma(1 - 1/p))^m - q^n/2^m \\
  > &1 - 2q^N \cdot (1 - \sigma(1 - 1/p))^m - q^n/2^m.
  \end{aligned}$$

  This completes the proof.

- If $d > q'$, $\mathcal{A}$ continues to compute

$$\hat{\mathbf{a}}_i := \mathbf{a}_i \bmod (q/q'), \quad \hat{b}_i := (((b_i - \langle \mathbf{a}_i, \mathbf{s}' \rangle - e'_i) \bmod q)/q') \bmod (q/q').$$

Next, we define $\hat{\mathbf{s}} := (\mathbf{s} - \mathbf{s}')/q', \hat{e}_i := (e_i - e'_i)/q'$. Then we have

$$\hat{b}_i = (\langle \hat{\mathbf{a}}_i, \hat{\mathbf{s}} \rangle + \hat{e}_i) \bmod (q/q').$$

Let $\hat{d} = \lfloor d/q' \rfloor$. Then $\hat{e}_i \in [\hat{d}]$. By Lemma 40 (stated below), we have $\hat{d}! \bmod (q/q') \neq 0$. Define the following event.

$$\mathcal{E}_3 : \text{ the event that } \hat{\mathcal{A}}(n, m, \hat{d}, q/q', \{(\hat{\mathbf{a}}_i, \hat{b}_i)\}_{1 \leq i \leq m}) \text{ outputs } \hat{\mathbf{s}}.$$

By the inductive hypothesis,

$$\Pr[\mathcal{E}_3] > 1 - 2(q/q')^{\hat{N}} \cdot (1 - \sigma(1 - 1/p))^m - (q/q')^n/2^m. \tag{31}$$

If $\mathcal{E}_3$ does not happen, $\mathcal{A}$ fails. Otherwise, it computes $\mathbf{s}$ by

$$\mathbf{s} = (q' \cdot \hat{\mathbf{s}} + (\mathbf{s}' \bmod q')) \bmod q.$$

Finally we have

$$\begin{aligned}
&\Pr[\mathbf{s} \leftarrow \mathcal{A}(n, m, d, q, \{(\mathbf{a}_i, b_i)\}_{1 \leq i \leq m})] \\
\geq &1 - \Pr[\neg \mathcal{E}_1] - \Pr[\neg \mathcal{E}_3] \\
> &1 - q^N \cdot (1 - \sigma(1 - 1/p))^m - 2(q/q')^N \cdot (1 - \sigma(1 - 1/p))^m - (q/q')^n/2^m \\
> &1 - 2q^N \cdot (1 - \sigma(1 - 1/p))^m - q^n/2^m.
\end{aligned}$$

Since $\mathcal{A}$ calls the Arora-Ge algorithm for at most $\log q$ times, its running time is still in $\mathsf{poly}(m)$. This completes the proof. □

Lemma 40. *Let $q = p^\kappa$ where p is prime and $\kappa \geq 2$ is an integer. Let d, q' be two integers such that $1 < q' < d < q$, $d! \bmod q \neq 0$, and $q' | q$. Then*

$$\lfloor d/q' \rfloor ! \bmod (q/q') \neq 0.$$

We postpone the proof of Lemma 40 to Appendix C.7.

Finally, we are able to prove Theorem 30. It is just an extension of Lemma 37 to arbitrary composite moduli.

Proof of Theorem 30. We design the algorithm $\mathcal{A}$ as follows. By the Chinese Remainder Theorem, there exists an index $1 \leq j^* \leq \ell$ such that $d! \bmod p_{j^*}^{\kappa_{j^*}} \neq 0$. Let the input LWE instance be $\{(\mathbf{a}_i, b_i = (\langle \mathbf{a}_i, \mathbf{s} \rangle + e_i) \bmod q\}_{1 \leq i \leq m}$. Let $\mathbf{s}' := \mathbf{s} \bmod p_{j^*}^{\kappa_{j^*}}$, $\mathbf{a}'_i := \mathbf{a}_i \bmod p_{j^*}^{\kappa_{j^*}}$, $b'_i := b_i \bmod p_{j^*}^{\kappa_{j^*}}$, $e'_i := e_i \bmod p_{j^*}^{\kappa_{j^*}}$. Using Lemma 37, there is an algorithm $\mathcal{A}'$ that runs in time poly(m) and

$$\Pr_{\{\mathbf{a}_i, e_i\}_i}[\mathbf{s}' \leftarrow \mathcal{A}'(n, m, d, p_{j^*}^{\kappa_{j^*}}, \{(\mathbf{a}'_i, b'_i)\}_{1 \leq i \leq m})] \geq 1 - 2p_{j^*}^{\kappa_{j^*} N} \cdot (1 - \sigma(1 - 1/p_{j^*}))^m - p_{j^*}^{\kappa_{j^*} n}/2^m.$$

Denote by $\mathcal{E}_1$ the event that $\mathcal{A}'(n, m, d, p_{j^*}^{\kappa_{j^*}}, \{(\mathbf{a}'_i, b'_i)\}_{1\le i\le m})$ outputs $\mathbf{s}'$. If $\mathcal{E}_1$ does not happen, we let the algorithm $\mathcal{A}$ fail. Otherwise, we continue by computing

$$e'_i = e_i \bmod p_{j^*}^{\kappa_{j^*}} = (b_i - \langle \mathbf{a}_i, \mathbf{s}\rangle) \bmod p_{j^*}^{\kappa_{j^*}} = (b'_i - \langle \mathbf{a}'_i, \mathbf{s}'\rangle) \bmod p_{j^*}^{\kappa_{j^*}}.$$

Note that $d! \bmod p_{j^*}^{\kappa_{j^*}} \neq 0$ cannot hold when $d \ge p_{j^*}^{\kappa_{j^*}}$, so we have

$$e_i \bmod q = e_i \bmod p_{j^*}^{\kappa_{j^*}} = e'_i.$$

Thus, for every $1 \le i \le m$, we have the linear equation

$$\langle \mathbf{a}_i, \mathbf{s}\rangle \equiv b_i - e_i \pmod q,$$

where $\mathbf{s}$ is the only unknown. Solving these linear equations yields a solution set $\mathcal{S}'$ (see Appendix A for details on solving linear equations over a ring). Let $\mathcal{E}_2$ denote the event that $\mathcal{S}' = \{\mathbf{s}\}$. By Lemma 14,

$$\Pr_{\{\mathbf{a}_i\}_i}[\mathcal{E}_2] = 1 - \Pr_{\{\mathbf{a}_i\}_i}[\exists \mathbf{s}' \neq \mathbf{s} \text{ s.t. } \mathbf{s}' \in \mathcal{S}'] > 1 - q^n/2^{\ell m}.$$

Finally, we have

$$\begin{aligned}
&\Pr[\mathbf{s} \leftarrow \mathcal{A}(n, m, d, q, \{(\mathbf{a}_i, b_i)\}_{1\le i\le m})] \\
\ge& 1 - \Pr[\neg\mathcal{E}_1] - \Pr[\neg\mathcal{E}_2] \\
>& 1 - 2p_{j^*}^{\kappa_{j^*}N} \cdot (1 - \sigma(1 - 1/p_{j^*}))^m - p_{j^*}^{\kappa_{j^*}n}/2^m - q^n/2^{\ell m} \\
>& 1 - 2q^N \cdot (1 - \sigma/2)^m - q^n/2^{m-1},
\end{aligned}$$

and $\mathcal{A}$ runs in $\mathsf{poly}(m)$ time because it only calls $\mathcal{A}'$ once and performs linear algebra operations on up to m dimensions. □

5 Weak PRFs do not exist in $\mathsf{NC}^0[p]$ for prime p

In this section, we show that weak PRFs do not exist in $\mathsf{NC}^0[p]$ for prime p.

We first prove that for any prime p, all Boolean circuits in $\mathsf{NC}[p]$ of depth k can be converted into $[\max(2, p-1)]^k$-degree polynomials over $\mathbb{Z}_p$ [1].

Theorem 41. *For any prime p, for any Boolean circuit C in $\mathsf{NC}[p]$ of depth k with input $\mathbf{x} \in \{0,1\}^n$, there exists a $GF(p)$ polynomial $f(x_1, x_2, \cdots, x_n)$ with degree $deg(f) \le [\max(2, p-1)]^k$ such that*

$$f(x_1, x_2, \cdots, x_n) = C(x_1, x_2, \cdots, x_n).$$

This theorem is a direct corollary of Theorem 57 and Theorem 58, whose proofs are postponed to Appendix B.3 and Appendix B.4 respectively. Let us make two remarks about Theorem 41. First, for constant p, k, the degree $deg(f)$ is in $O(1)$, and therefore the number of monomials in f

[1] This result is folklore. Here we include a proof for completeness. The result can also be derived from `https://cstheory.stackexchange.com/questions/44013/razborov-smolensky-polynomial-argument-on-textrmaccq-where-q-is-a-prime`.

is polynomial in n. Second, the polynomial $f(x_1, x_2, \cdots, x_n)$ in $GF(p)$ outputs either 0 or 1 when the input $\mathbf{x}$ is binary. When the input is not binary, the output of f may not be binary.

Next we show a polynomial-time algorithm that distinguishes all constant-degree polynomials on $\mathbb{Z}_q$ from random functions, where the inputs and outputs are restricted to be binary and the input queries are uniformly random. Here q can even be a composite number with possibly two distinct prime factors. Let us remark that the attack also works when the inputs and outputs are not binary. Here we only present the binary case since it suffices for our purpose.

Theorem 42. *For any integer $q \geq 2$, any constant-degree polynomial f over $\mathbb{Z}_q$ that outputs a binary output when the input is binary, there is a $\mathsf{poly}(n)$ time algorithm $\mathcal{A}$ and an $m \in \mathsf{poly}(n)$ such that*

$$\begin{aligned} &\Pr_{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(m)} \leftarrow \{0,1\}^n} [\mathcal{A}(\{\mathbf{x}^{(i)}, f(\mathbf{x}^{(i)})\}_{1 \leq i \leq m}) = 1] \\ &- \Pr_{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(m)} \leftarrow \{0,1\}^n} [\mathcal{A}(\{\mathbf{x}^{(i)}, \mathcal{U}(\{0,1\}))\}_{1 \leq i \leq m}) = 1] \geq 1 - \mathsf{negl}(n), \end{aligned} \tag{32}$$

Proof. Let $f(\mathbf{x}) := f(x_1, \ldots, x_n)$ be a polynomial over $\mathbb{Z}_q$ such that $deg(f) \leq d$, where d is a constant. Then there are at most $N = O(n^d)$ different monomials in f. Let $\{g_i(\mathbf{x})\}_{i \in [N]}$ denote the set of all monomials (we assume $\{g_i(\mathbf{x})\}_{i \in [N]}$ enumerates all monomials in certain fixed order, e.g. $g_1(\mathbf{x}) = 1$, $g_2(\mathbf{x}) = x_1$,..., $g_{n+1}(\mathbf{x}) = x_n$, $g_{n+2}(\mathbf{x}) = x_1^2$, etc.). Then $f(\mathbf{x})$ can be written as a linear combination of $g_i(\mathbf{x})$. Concretely, let the coefficients be $c_i \in \mathbb{Z}_q$ for $i = 1, ..., N$, and we have

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i g_i(\mathbf{x}).$$

Note that the coefficients c_i are not known to the algorithm.

Let $m := N \log_2(q) + \omega(\log n) \in \mathsf{poly}(n)$. We now design an algorithm $\mathcal{A}$ running in $\mathsf{poly}(n)$ time that distinguishes f from a truly random function over random queries, i.e., $\mathcal{A}$ satisfies Eqn. (32).

$\mathcal{A}$ works as follows. On receiving $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{1 \leq i \leq m}$, it first computes $g_j(\mathbf{x}^{(i)})$ for every $1 \leq i \leq m, 1 \leq j \leq N$. Then $\mathcal{A}$ considers the following linear system of equations of $\{z_j\}_{1 \leq j \leq N}$

$$\forall 1 \leq i \leq m, \ \sum_{j=1}^{N} g_j(\mathbf{x}^{(i)}) z_j = y^{(i)} \bmod q.$$

$\mathcal{A}$ outputs 1 if the linear system has at least one solution, outputs 0 otherwise.

- In the first case, that is, $y^{(i)} = f(\mathbf{x}^i)$, there is at least one solution: $z_j = c_j$ for all $1 \leq j \leq N$.
- In the second case, that is, $y^{(i)} \leftarrow \mathcal{U}(\{0,1\})$, we claim that with overwhelming probability, the linear system of equations has no solution, i.e., for any $z_1, \ldots, z_N \in \mathbb{Z}_q$, there exists some i^* such that $\sum_{j=1}^{N} g_j(\mathbf{x}^{(i^*)}) z_j \neq y^{(i^*)}$. We prove this by bounding the probability of its

complementary event:

$$\begin{aligned}
&\Pr[\exists z_1, \ldots, z_N \in \mathbb{Z}_q, \forall 1 \le i \le m, \sum_{j=1}^{N} g_j(\mathbf{x}^{(i)}) z_j = y^{(i)}] \\
\le &\sum_{z_1,\ldots,z_N \in \mathbb{Z}_q} \Pr[\forall 1 \le i \le m, \sum_{j=1}^{N} g_j(\mathbf{x}^{(i)}) z_j = y^{(i)}] \\
= &\sum_{z_1,\ldots,z_N \in \mathbb{Z}_q} \prod_{1 \le i \le m} \Pr[\sum_{j=1}^{N} g_j(\mathbf{x}^{(i)}) z_j = y^{(i)}] \\
= &\sum_{z_1,\ldots,z_N \in \mathbb{Z}_q} \prod_{1 \le i \le m} \frac{1}{2} \\
= &\, q^N \cdot 2^{-m},
\end{aligned}$$

where all probabilities are over $\mathbf{y} \leftarrow \mathcal{U}(\{0,1\}^m)$. Since we set $m = N \log_2(q) + \omega(\log n)$, we have

$$\Pr[\forall z_1, \ldots, z_N \in \mathbb{Z}_q, \exists 1 \le i^* \le m, \sum_{j=1}^{N} g_j(\mathbf{x}^{(i^*)}) z_j \ne y^{(i^*)}] \ge 1 - \mathsf{negl}(n).$$

□

However, this attack cannot be generalized to $\mathsf{NC}^0[q]$ when q has at least two distinct prime factors. In particular, Theorem 43 shows that there exists a circuit in $\mathsf{NC}^0[q]$ that cannot be computed by low-degree polynomials, and hence the attack presented in this section does not apply to such circuits. We emphasize that the MOD_q gate in the circuit class $\mathsf{NC}^0[q]$, whose output is binary, is not the same as the modulo-q operation over the ring $\mathbb{Z}_q$, whose computability by polynomials over $\mathbb{Z}_q$ is straightforward.

Theorem 43. *Let N, q be positive integers such that q is not a prime power. Then there is no polynomial in $\mathbb{Z}_N[X]$ with a polynomial degree and polynomially many monomials that computes MOD_q.*

Proof. We prove by contradiction. Suppose that there is a polynomial f_N in $\mathbb{Z}_N[X]$ with polynomial degree and polynomial monomials that computes MOD_q. Let p be a prime factor of N, define f_p as:

$$f_p = f_N \bmod p.$$

Then f_p is a polynomial in $\mathbb{Z}_p[X]$ with a polynomial degree and polynomially many monomials. Note that MOD_q is a function from $\{0,1\}^n$ to $\{0,1\}$, then for any $(x_1, x_2, \cdots x_n) \in \{0,1\}^n$,

$$f_p(x_1, x_2, \cdots x_n) = f_N(x_1, x_2, \cdots x_n) \bmod p = f_N(x_1, x_2, \cdots x_n).$$

Then f_p computes MOD_q. Now we show that f_p can be computed with an $\mathsf{AC}^0[p]$ circuit of depth 2 and polynomially many gates. The circuit is as follows:

- For each monomial $x_{i_1} x_{i_2} \cdots x_{i_k}$, compute it by an AND gate with unbounded fan-in $x_{i_1} \wedge x_{i_2} \wedge \cdots \wedge x_{i_k}$.

- Sum all monomials using a MOD_p gate. If a monomial is $c \cdot x_{i_1} x_{i_2} \cdots x_{i_k}$, then we decompose it to c copies of $x_{i_1} x_{i_2} \cdots x_{i_k}$ and feed it into different input sectors of the MOD_p gate.

This indicates that MOD_q can be computed in $\mathsf{AC}^0[p]$, which contradicts the following lemma in [Raz87, Smo87]:

Lemma 44. *Let p be a prime number and q is not a power of p. Then MOD_q is not computable in $\mathsf{AC}^0[p]$ with polynomially many gates.*

This completes the proof. □

6 Candidate Weak PRFs

6.1 Weak PRFs at Least as Secure as LWE

In this subsection, we propose new LAM-based weak PRFs that are at least as secure as LWE.

Construction 45. *Let n be the security parameter. For any two integers $q_1, q_2 \geq 2$ such that $\gcd(q_1, q_2) = 1$, and q_1/q_2 is super-polynomial in n, define the function family $\mathcal{F}_{q_1,q_2} := \left\{F_{\mathbf{s}} : \mathbb{Z}_{q_1}^n \to \mathbb{Z}_{q_2}\right\}_{\mathbf{s} \in \mathbb{Z}_{q_1}^n}$ as follows:*

$$F_{\mathbf{s}}(\mathbf{x}) := (\langle \mathbf{s}, \mathbf{x} \rangle \bmod q_1) \bmod q_2. \tag{33}$$

Theorem 46. *Let n be the security parameter. For any $B = \mathsf{poly}(n)$, let D_B be an arbitrary B-bounded and balanced distribution. For any two integers $q_1, q_2 \geq 2$ such that $\gcd(q_1, q_2) = 1$, and q_1/q_2 is super-polynomial in n, let $\mathcal{F}_{q_1,q_2}$ be defined as in Construction 45. Assume the hardness of $DLWE_{n,m,q_1,D_B}$ for any $m = \mathsf{poly}(n)$. Then $\mathcal{F}_{q_1,q_2}$ is a weak PRF family.*

The proof of Theorem 46 is straightforward by Theorem 24.

6.2 Low-Depth Weak PRFs Candidates from LAM

In this subsection, we construct new candidate weak PRFs in $\mathsf{NC}^0[q]$ for any positive integer $q > 2$ that is not a prime-power. When $q = p_1^{\kappa_1} p_2^{\kappa_2}$, the new candidate weak PRFs are in $\mathsf{NC}^0[p_1, p_2]$ for any distinct primes p_1, p_2.

Construction 47. *Let n be the security parameter. Let $q_1, q_2 \geq 2$ be two constant integers such that q_1 is not a prime power, $\gcd(q_1, q_2) = 1$, and $(\lfloor q_1/q_2 \rfloor)! \bmod q_1 = 0$. Let p be a prime number such that $p | q_1$. Let $\ell = \Theta(n)$. We define the function family $\mathcal{G}_{q_1,q_2,p} := \{g_{\mathbf{S}} : \mathbb{Z}_{q_1}^n \to \mathbb{Z}_p\}_{\mathbf{S} \in \mathbb{Z}_{q_1}^{n \times \ell}}$ as follows:*

$$g_{\mathbf{S}}(\mathbf{x}) := \left(\sum_{i=1}^{\ell} ((\langle \mathbf{s}_i, \mathbf{x} \rangle \bmod q_1) \bmod q_2) \right) \bmod p.$$

That is, the mapping $g_{\mathbf{S}}$ first computes an alternating-moduli instance $(\langle \mathbf{s}_i, \mathbf{x} \rangle \bmod q_1) \bmod q_2$ for every column $\mathbf{s}_i$ of $\mathbf{S}$; then, it sums up all the ℓ instances over $\mathbb{Z}_p$ to get the final output.

Computability in $\mathsf{NC}^0[q_1]$. Here we prove the function $g_{\mathbf{S}}$ is computable in $\mathsf{NC}^0[q_1]$. We first show the following theorem, which we prove later in Appendix B.5, Theorem 60.

Theorem 48. *For any constants $q_1, q_2 \geq 2$ and a prime $p|q_1$, the function $g_{\mathbf{S}}(\mathbf{x})$ can be computed in $\mathsf{NC}^0[q_1]$ with depth 3.*

Let us remark that Theorem 48 holds regardless of whether q_1 is a prime power or not. Assume $q_1 = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$ for some constant positive integers $\alpha_1, \alpha_2, \cdots, \alpha_k$ and distinct primes $p_1, p_2, \cdots, p_k$, and then we have $\mathsf{NC}^0[q_1] = \mathsf{NC}^0[p_1, p_2, \cdots, p_k]$ (the proof is given in Appendix B.7). Specifically, when $q_1 = p_1^{\alpha_1}$ is a prime power, the function $g_{\mathbf{S}}$ is computable in $\mathsf{NC}^0[p_1]$, where there exist no weak PRFs according to the result in Section 5. Hence, the smallest safe parameter choice is $k = 2$. Concretely, by selecting $q_1 = p_1^{\alpha_1} p_2^{\alpha_2}$, we have $g_{\mathbf{S}}$ is both computable in $\mathsf{NC}^0[p_1, p_2]$ and secure against the polynomial time attack in Section 5.

Security Analysis. The security of our candidate weak PRF family $\mathcal{G}_{q_1,q_2,p}$ follows from the hardness of $\mathrm{DLAM}_{n,m,q_1,q_2}$ combined with an application of Lemma 16. However, the hardness of $\mathrm{DLAM}_{n,m,q_1,q_2}$ with constant q_1, q_2 cannot be based on any existing assumptions. Therefore, we provide more evidence for the security of $\mathcal{G}_{q_1,q_2,p}$ by analyzing some typical attacks in the following.

Arora-Ge Attack. The Arora-Ge algorithm can break LAM for certain parameter sets, potentially leading to a key recovery attack on the weak PRF family. Note that for any LAM sample $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle \bmod q_1 \bmod q_2)$, we can find an integer $e \in [\lfloor q_1/q_2 \rceil + 1]$ such that $b = (\langle \mathbf{a}, \mathbf{s} \rangle - q_2 \cdot e) \bmod q_1$. Specifically, when $((q_1 - 1) \bmod q_2) < b < q_2$ (such b must exist since $\gcd(q_1, q_2) = 1$), we have $e \in [\lfloor q_1/q_2 \rceil]$.

To apply the Arora-Ge attack to the LAM problem, we first collect a set of samples $\{(\mathbf{a}_i, b_i)\}_{1 \leq i \leq m}$ with $((q_1 - 1) \bmod q_2) < b_i < q_2$ for each $1 \leq i \leq m$ (this is achieved by discarding the samples with $0 \leq b_i \leq ((q_1 - 1) \bmod q_2)$ from an instance of $\mathrm{LAM}_{n,O(m),q_1,q_2}$). Then, for every i, compute

$$(\mathbf{a}_i', b_i') := (-q_2^{-1}\mathbf{a}_i \bmod q_1, -q_2^{-1} b_i \bmod q_1) = (-q_2^{-1}\mathbf{a}_i \bmod q_1, \langle -q_2^{-1}\mathbf{a}_i, \mathbf{s} \rangle + e_i) \bmod q_1),$$

where $q_2^{-1} \in \mathbb{Z}_{q_1}$ satisfies $q_2^{-1} q_2 \bmod q_1 = 1$. Then $\{(\mathbf{a}_i', b_i')\}_{1 \leq i \leq m}$ is a set of LWE samples with error in $[\lfloor q_1/q_2 \rfloor]$. Apply the Arora–Ge algorithm to this set. If $(\lfloor q_1/q_2 \rfloor)! \bmod q_1 \neq 0$, the algorithm succeeds in finding $\mathbf{s}$. Otherwise, it learns nothing about $\mathbf{s}$.

To summarize, the Arora-Ge algorithm solves the LAM problem when $\lfloor q_1/q_2 \rfloor! \bmod q_1 \neq 0$. In our construction, we add a restriction on q_1, q_2 that $\lfloor q_1/q_2 \rfloor! \bmod q_1 = 0$, so the Arora-Ge attack does not work in our case.

Linear Cryptanalysis. Linear cryptanalysis [Mat93] is a type of attack that attempts to find linear relationships between the bits of random input–output pairs. Previous works [Tak24, Vau03] have shown that, for a general hypothesis class $\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y}\}$, proving exponential security against both linear and differential[2] cryptanalysis reduces to showing that the following value $\epsilon_{\mathcal{H}}$ is exponentially small:

$$\epsilon_{\mathcal{H}} := \max_{x \in \mathcal{X}} \mathbb{E}_{x' \leftarrow \mathcal{X}} \left[\varepsilon_{\mathcal{H}}(x, x')^2 \right]^{1/2},$$

[2]Differential cryptanalysis does not apply to weak PRFs. [BR17]

where for any $x, x' \in \mathcal{X}$,

$$\varepsilon_{\mathcal{H}}(x,x') := \begin{cases} \sum_{y,y' \in \mathcal{Y}} \left| \Pr_{h \leftarrow \mathcal{H}}[h(x)=y, h(x')=y'] - |\mathcal{Y}|^{-2} \right|, & \text{if } x \neq x' \\ \sum_{y \in \mathcal{Y}} \left| \Pr_{h \leftarrow \mathcal{H}}[h(x)=y] - |\mathcal{Y}|^{-1} \right|, & \text{if } x = x' \end{cases}$$

Note that if $\mathcal{H}$ is pairwise independent, then $\varepsilon_{\mathcal{H}}(x,x') = 0$ for distinct $x, x' \in \mathcal{X}$. Hence, $\epsilon_{\mathcal{H}}$ can be viewed as a measure of pairwise independence of $\mathcal{H}$.

Instead of bounding $\epsilon_{\mathcal{G}_{q_1,q_2,p}}$, we focus on another function family, which is formed by replacing the domain of $g_{\mathbf{S}}$ with $\mathbb{Z}_{q_1}^{n*}$. Concretely, we define $\mathcal{G}^*_{q_1,q_2,p} := \{g^*_{\mathbf{S}} : \mathbb{Z}_{q_1}^{n*} \to \mathbb{Z}_p\}_{\mathbf{S}}$, where for any $\mathbf{x} \in \mathbb{Z}_{q_1}^{n*}$, $g^*_{\mathbf{S}}(\mathbf{x}) := g_{\mathbf{S}}(\mathbf{x})$. By Lemma 15, we have that $\mathcal{G}^*_{q_1,q_2,p}$ and $\mathcal{G}_{q_1,q_2,p}$ are statistically indistinguishable given random queries. Therefore, we only need to bound $\epsilon_{\mathcal{G}^*_{q_1,q_2,p}}$.

Theorem 49. $\epsilon_{\mathcal{G}^*_{q_1,q_2,p}} \leq 1/2^{O(n)}$.

Proof. For any $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}_{q_1}^{n*}$, let $\mathcal{E}$ be the event that there exists two invertible matrices $\mathbf{U} \in \mathbb{Z}_{q_1}^{n\times n}, \mathbf{V} \in \mathbb{Z}_{q_1}^{2\times 2}$ such that $\mathbf{U}(\mathbf{x}\|\mathbf{x}')\mathbf{V} = \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix}$. We have for any fixed $\mathbf{x}$, $\Pr_{\mathbf{x}' \leftarrow \mathbb{Z}_{q_1}^{n*}}[\mathcal{E}] = 1 - 1/2^{O(n)}$. And for all $\mathbf{z}, \mathbf{z}' \in \mathbb{Z}_{q_1}^{\ell}$,

$$\begin{aligned}
&\Pr_{\mathbf{S} \leftarrow \mathbb{Z}_{q_1}^{n\times\ell}}[\mathbf{S}^T\mathbf{x} = \mathbf{z} \wedge \mathbf{S}^T\mathbf{x}' = \mathbf{z}' \mid \mathcal{E}] \\
=&\Pr_{\mathbf{S} \leftarrow \mathbb{Z}_{q_1}^{n\times\ell}}[\mathbf{S}^T\mathbf{U}\begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix}\mathbf{V} = (\mathbf{z}\|\mathbf{z}') \mid \mathcal{E}] \\
=&\Pr_{\mathbf{S} \leftarrow \mathbb{Z}_{q_1}^{n\times\ell}}[\mathbf{S}^T \cdot \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix} = (\mathbf{z}\|\mathbf{z}')\mathbf{V}^{-1} \mid \mathcal{E}] \\
=&\Pr_{\mathbf{S} \leftarrow \mathbb{Z}_{q_1}^{n\times\ell}}[\begin{bmatrix} s_1(1), & s_1(2) \\ s_2(1), & s_2(2) \\ \cdots, & \cdots \\ s_\ell(1), & s_\ell(2)\end{bmatrix} = (\mathbf{z}\|\mathbf{z}')\mathbf{V}^{-1} \mid \mathcal{E}] \\
=&1/q_1^{2\ell}.
\end{aligned}$$

Hence, when $\mathcal{E}$ happens, $\mathbf{S}^T\mathbf{x}$ and $\mathbf{S}^T\mathbf{x}'$ are independent, which means $g^*_{\mathbf{S}}(\mathbf{x})$ and $g^*_{\mathbf{S}}(\mathbf{x}')$ are also independent. By Lemma 16, the statistical distance between $g^*_{\mathbf{S}}(\mathbf{x})$ and $\mathcal{U}(\mathbb{Z}_p)$ is $1/2^{O(n)}$. Therefore, for $y, y' \in \mathbb{Z}_p$, we have

$$\left| \Pr_{\mathbf{S} \leftarrow \mathbb{Z}_{q_1}^{n\times\ell}}[g^*_{\mathbf{S}}(\mathbf{x}) = y, g^*_{\mathbf{S}}(\mathbf{x}') = y' \mid \mathcal{E}] - 1/p^2 \right| = 1/2^{O(n)}.$$

Finally, we have

$$\begin{aligned}
\epsilon^2_{\mathcal{G}^*_{q_1,q_2,p}} &= \max_{\mathbf{x}\in\mathbb{Z}_{q_1}^{n*}} \mathbb{E}_{\mathbf{x}'\leftarrow\mathbb{Z}_{q_1}^{n*}}[\varepsilon_{\mathcal{G}^*_{q_1,q_2,p}}(\mathbf{x},\mathbf{x}')^2] \\
&= \max_{\mathbf{x}\in\mathbb{Z}_{q_1}^{n*}} \big(\Pr_{\mathbf{x}'\leftarrow\mathbb{Z}_{q_1}^{n*}}[\mathcal{E}] \cdot \mathbb{E}_{\mathbf{x}'\leftarrow\mathbb{Z}_{q_1}^{n*}}[\varepsilon_{\mathcal{G}^*_{q_1,q_2,p}}(\mathbf{x},\mathbf{x}')^2 \mid \mathcal{E}] \\
&\quad + \Pr_{\mathbf{x}'\leftarrow\mathbb{Z}_{q_1}^{n*}}[\neg\mathcal{E}] \cdot \mathbb{E}_{\mathbf{x}'\leftarrow\mathbb{Z}_{q_1}^{n*}}[\varepsilon_{\mathcal{G}^*_{q_1,q_2,p}}(\mathbf{x},\mathbf{x}')^2 \mid \neg\mathcal{E}]\big) \\
&\le \max_{\mathbf{x}\in\mathbb{Z}_{q_1}^{n*}} \mathbb{E}_{\mathbf{x}'\leftarrow\mathbb{Z}_{q_1}^{n*}}[\varepsilon_{\mathcal{G}^*_{q_1,q_2,p}}(\mathbf{x},\mathbf{x}')^2 \mid \mathcal{E}] + 1/2^{O(n)} \\
&= \max_{\mathbf{x}\in\mathbb{Z}_{q_1}^{n*}} \mathbb{E}_{\mathbf{x}'\leftarrow\mathbb{Z}_{q_1}^{n*}}[\big(\sum_{y,y'\in\mathbb{Z}_p} \big| \Pr_{\mathbf{S}\leftarrow\mathbb{Z}_{q_1}^{n\times\ell}}[g^*_{\mathbf{S}}(\mathbf{x})=y, g^*_{\mathbf{S}}(\mathbf{x}')=y'] - 1/p^2\big|\big)^2 \mid \mathcal{E}] + 1/2^{O(n)} \\
&\le \max_{\mathbf{x}\in\mathbb{Z}_{q_1}^{n*}} \mathbb{E}_{\mathbf{x}'\leftarrow\mathbb{Z}_{q_1}^{n*}}[\big(\sum_{y,y'\in\mathbb{Z}_p} 1/2^{O(n)}\big)^2 \mid \mathcal{E}] + 1/2^{O(n)} \\
&= 1/2^{O(n)}.
\end{aligned}$$

Thus, $\epsilon_{\mathcal{G}^*_{q_1,q_2,p}}$ is exponentially small, which means $\mathcal{G}^*_{q_1,q_2,p}$ is exponentially secure against linear cryptanalysis. □

BKW Attack. When converting an LAM instance $(\mathbf{A}, \mathbf{b} = (\mathbf{A}^T\mathbf{s} \bmod q_1) \bmod q_2)$ into an LWE instance, each entry of $\mathbf{b}$ can be expressed as:

$$b_i \equiv \langle \mathbf{a}_i, \mathbf{s}\rangle - q_2 e_i \pmod{q_1}.$$

Note that the probability that $e_i = \lfloor q_1/q_2 \rfloor$ is smaller than the probability that $e_i = k$ for some $k < \lfloor q_1/q_2 \rfloor$. This implies that the BKW algorithm [BKW03] provides a $2^{O(n/\log n)}$ time attack on the LAM problem with constant moduli. However, we do not know how to apply it directly to $\mathcal{G}_{q_1,q_2,p}$.

Algebraic Attacks. We have shown in Theorem 43 that when q has at least two distinct prime factors, the MOD_q gate cannot be computed by any polynomials in $\mathbb{Z}_N$ for an arbitrary N, even if $N = q$. Our theorem aims to show that when q has at least two prime factors, *not all* circuits in $\mathsf{NC}^0[q]$ can be computed by low-degree polynomials (unlike for prime power q, Theorem 41 shows *all* circuits in $\mathsf{NC}^0[q]$ can be computed by low-degree polynomials). This gives some evidence that our weak PRF candidates modulo composite, non-prime-power q are unlikely to be broken by algebraic attacks.

6.3 Low-Depth Weak PRF Candidates from LWR

We also propose new candidate weak PRFs based on the LWR problem.

Construction 50. *Let n be the security parameter. Let $q_1, q_2 \ge 2$ be two constant integers such that q_1 is not a prime power, and $(\lfloor q_1/q_2 \rfloor)! \bmod q_1 = 0$. Let p be a prime number such that $p|q_1$. Let $\ell = \Theta(n)$. We define the function family $\mathcal{L}_{q_1,q_2,p} := \{L_{\mathbf{S}} : \mathbb{Z}_{q_1}^n \to \mathbb{Z}_p\}_{\mathbf{S}\in\mathbb{Z}_{q_1}^{n\times\ell}}$ as follows:*

$$L_{\mathbf{S}}(\mathbf{x}) := \left(\sum_{i=1}^{\ell} \left\lfloor \frac{q_2(\langle \mathbf{x}, \mathbf{s}_i\rangle \bmod q_1)}{q_1} \right\rceil \bmod q_2 \right) \bmod p.$$

Remark 51. *The choice of q_1, q_2 in this construction is more flexible than that in Construction 47, since the security of LWR does not necessarily require $gcd(q_1, q_2) = 1$.*

Similar to the LAM-based construction, $\mathcal{L}_{q_1,q_2,p}$ is a weak PRF family in $\mathsf{NC}^0[q_1]$ assuming the hardness of $\mathrm{DLWR}_{n,m,q_1,q_2}$ for every $m = \mathsf{poly}(n)$. We omit the computability and security analysis of $\mathcal{L}_{q_1,q_2,p}$, which are similar to that of $\mathcal{G}_{q_1,q_2,p}$.

Specifically, when $q_2|q_1$, we further propose a simpler candidate weak PRF family as follows.

Construction 52. *Let n be the security parameter. Let $q_1, q_2 \geq 2$ be two constant integers such that q_1 is not a prime power, $q_2|q_1$, and $(q_1/q_2)! \bmod q_1 = 0$. We define the function family $\mathcal{K}_{q_1,q_2} := \left\{K_{\mathbf{s}} : \mathbb{Z}_{q_1}^n \to \mathbb{Z}_{q_2}\right\}$ as:*

$$K_{\mathbf{s}}(\mathbf{x}) := \left\lfloor \frac{q_2(\langle \mathbf{x}, \mathbf{s} \rangle \bmod q_1)}{q_1} \right\rceil \bmod q_2.$$

In this case, the output of $K_{\mathbf{s}}$ is uniformly distributed over $\mathbb{Z}_{q_2}$. We conjecture that $\mathcal{K}$ is a weak PRF family with security subexponential in n. It is not exponentially secure since the BKW attack works on LWR even if $q_2|q_1$. For example, let $q_1 = 6, q_2 = 2$. Then any instance of $\mathrm{DLWR}_{n,m,6,2}$ can be transformed to an instance of $\mathrm{LWE}_{n,m,6,\mathcal{U}([3])}$. Modulo this LWE instance by 2, and we will get a learning parity with noise (LPN) instance with noise rate 1/3. Thus, the BKW attack still applies. We can follow the methodology in subsection 6.2 to analyze other properties of $\mathcal{K}_{q_1,q_2}$, i.e., (i) computablity in $\mathsf{NC}^0[q_1]$, (ii) security against linear cryptanalysis, (iii) security against the Arora-Ge attack, and (iv) evidence of security against algebraic attacks.

Acknowledgments

We thank Yuval Ishai and anonymous reviewers for their valuable comments.

References

[ABG+14] Adi Akavia, Andrej Bogdanov, Siyao Guo, Akshay Kamath, and Alon Rosen. Candidate weak pseudorandom functions in $\mathsf{AC}^0 \circ \mathsf{MOD2}$. In *ITCS*, pages 251–260. ACM, 2014.

[ACF+15] Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. Algebraic algorithms for LWE problems. *ACM Commun. Comput. Algebra*, 49(2):62, 2015.

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *ICALP (1)*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 2011.

[APRR24] Navid Alamati, Guru-Vamsi Policharla, Srinivasan Raghuraman, and Peter Rindal. Improved alternating-moduli prfs and post-quantum signatures. In *CRYPTO (8)*, volume 14927 of *Lecture Notes in Computer Science*, pages 274–308. Springer, 2024.

[AR16] Benny Applebaum and Pavel Raykov. Fast pseudorandom functions based on expander graphs. In *TCC (B1)*, volume 9985 of *Lecture Notes in Computer Science*, pages 27–56, 2016.

[AR25] Irati Manterola Ayala and Håvard Raddum. Zeroed out: Cryptanalysis of weak prfs in alternating moduli. *IACR Transactions on Symmetric Cryptology*, 2025(2):1–15, 2025.

[BCG+20] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Correlated pseudorandom functions from variable-density LPN. In *FOCS*, pages 1069–1080. IEEE, 2020.

[BCG+21] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Low-complexity weak pseudorandom functions in $\mathsf{AC}^0[\mathsf{MOD2}]$. In *CRYPTO (4)*, volume 12828 of *Lecture Notes in Computer Science*, pages 487–516. Springer, 2021.

[BCPR25] Christina Boura, Geoffroy Couteau, Léo Perrin, and Yann Rotella. Sok: On shallow weak prfs A common symmetric building block for MPC protocols. *IACR Trans. Symmetric Cryptol.*, 2025(3):289–336, 2025.

[BGM+16] Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon Rosen. On the hardness of learning with rounding over small modulus. In *TCC (A1)*, volume 9562 of *Lecture Notes in Computer Science*, pages 209–224. Springer, 2016.

[BIP+18] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. Exploring crypto dark matter: - new simple PRF candidates and their applications. In *TCC (2)*, volume 11240 of *Lecture Notes in Computer Science*, pages 699–729. Springer, 2018.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *STOC*, pages 575–584. ACM, 2013.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 719–737. Springer, 2012.

[BR17] Andrej Bogdanov and Alon Rosen. Pseudorandom functions: Three decades later. In *Tutorials on the Foundations of Cryptography*, pages 79–158. Springer International Publishing, 2017.

[CCKK21] Jung Hee Cheon, Wonhee Cho, Jeong Han Kim, and Jiseung Kim. Adventures in crypto dark matter: Attacks and fixes for weak pseudorandom functions. In *Public Key Cryptography (2)*, volume 12711 of *Lecture Notes in Computer Science*, pages 739–760. Springer, 2021.

[DGH+21] Itai Dinur, Steven Goldfeder, Tzipora Halevi, Yuval Ishai, Mahimna Kelkar, Vivek Sharma, and Greg Zaverucha. Mpc-friendly symmetric cryptography from alternating moduli: Candidates, protocols, and applications. In *CRYPTO (4)*, volume 12828 of *Lecture Notes in Computer Science*, pages 517–547. Springer, 2021.

[DJK25] Youlong Ding, Aayush Jain, and Ilan Komargodski. A new approach for lpn-based pseudorandom functions: Low-depth and key-homomorphic. In *STOC*, pages 1898–1909. ACM, 2025.

[JMN23] Thomas Johansson, Willi Meier, and Vu Nguyen. Differential cryptanalysis of mod-2/mod-3 constructions of binary weak prfs. In *ISIT*, pages 477–482. IEEE, 2023.

[LMN89] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. In *FOCS*, pages 574–579. IEEE Computer Society, 1989.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.

[NMSÜ25] Miguel Cueto Noval, Simon-Philipp Merz, Patrick Stählin, and Akin Ünal. On the soundness of algebraic attacks against code-based assumptions. In *EUROCRYPT (6)*, volume 15606 of *Lecture Notes in Computer Science*, pages 385–415. Springer, 2025.

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004.

[Raz87] Alexander A Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mat. Zametki*, 41(4):598–607, 1987.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 77–82, 1987.

[STA20] Chao Sun, Mehdi Tibouchi, and Masayuki Abe. Revisiting the hardness of binary error LWE. In *ACISP*, volume 12248 of *Lecture Notes in Computer Science*, pages 425–444. Springer, 2020.

[Ste24] Matthias Johann Steiner. The complexity of algebraic algorithms for LWE. In *EUROCRYPT (3)*, volume 14653 of *Lecture Notes in Computer Science*, pages 375–403. Springer, 2024.

[Tak24] Rustem Takhanov. Almost pairwise independence and resilience to deep learning attacks. *IACR Commun. Cryptol.*, 1(3):32, 2024.

[Vau03] Serge Vaudenay. Decorrelation: A theory for block cipher security. *J. Cryptol.*, 16(4):249–286, 2003.

[YS16] Yu Yu and John P. Steinberger. Pseudorandom functions in almost constant depth from low-noise LPN. In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 154–183. Springer, 2016.

Appendix

A Solving linear equations on a ring

In this section, we formalize a folklore method to solve any linear system of equations over $\mathbb{Z}_q$ for any non-prime q. By the Chinese Remainder Theorem, we only need to focus on the prime power

ring $\mathbb{Z}_{p^\kappa}$. We first prove the following proposition.

Proposition 53. *Let p be any prime number and m, n, κ be any positive integers. There exists a polynomial-time algorithm that, for any matrix $\mathbf{A} \in \mathbb{Z}_{p^\kappa}^{m\times n}$, finds invertible matrices $\mathbf{L} \in \mathbb{Z}_{p^\kappa}^{m\times m}, \mathbf{R} \in \mathbb{Z}_{p^\kappa}^{n\times n}$ such that*

$$\mathbf{LAR} = \mathbf{\Lambda},$$

where the matrix $\mathbf{\Lambda} \in \mathbb{Z}_{p^\kappa}^{m\times n}$ satisfies:

(i) For any $1 \le i \le m, 1 \le j \le n$, if $i \ne j$, then $\Lambda(i,j) = 0$.

(ii) For any $1 \le i \le \min(m,n)$, if $\Lambda(i,i) \ne 0$, then there exists an integer $0 \le \alpha_i < \kappa$ such that

$$\Lambda(i,i) = p^{\alpha_i}$$

.

(iii) For any $1 \le i_1 < i_2 \le \min(m,n)$, if $\Lambda(i_2,i_2) \ne 0$, then

$$\Lambda(i_1,i_1) \ne 0 \text{ and } \Lambda(i_1,i_1) \le \Lambda(i_2,i_2),$$

where the comparison "$\le$" is done on $\mathbb{Z}$.

Proof. Any element in $\mathbb{Z}_{p^\kappa}$ can be factored as the product of a unit of $\mathbb{Z}_{p^\kappa}$ and a power of p. (Note that $0 = 1 \cdot p^\kappa$.) So for every $1 \le i \le m, 1 \le j \le n$, we find a unit $u_{i,j}$ and an integer $0 \le e_{i,j} \le \kappa$ such that $A(i,j) = u_{i,j} \cdot p^{e_{i,j}}$.

Without loss of generality, assume $n \le m$. Our goal is to find invertible matrices $\mathbf{L} \in \mathbb{Z}_{p^\kappa}^{m\times m}, \mathbf{R} \in \mathbb{Z}_{p^\kappa}^{n\times n}$ such that

$$\mathbf{\Lambda} := \mathbf{LAR} = \begin{bmatrix} \mathbf{\Lambda'} \\ \mathbf{0} \end{bmatrix},$$

where $\mathbf{\Lambda'} = diag\{p^{\alpha_1}, \ldots, p^{\alpha_n}\}$ with $0 \le \alpha_1 \le \cdots \le \alpha_n \le \kappa$.

We prove this by induction.

- When $m \ge n = 1$, we only need to show there exists an invertible matrix $\mathbf{L} \in \mathbb{Z}_{p^\kappa}^{m\times m}$ such that

  $$\mathbf{LA} = [p^\alpha, 0, 0, \ldots, 0]^T$$

  for some $0 \le \alpha \le \kappa$. Let

  $$i^* := \arg\min_{1\le i\le m} e_{i,1}.$$

  We can assume $i^* = 1$ without loss of generality, since when $i^* \ne 1$, we can swap the position of $A(i^*,1)$ and $A(1,1)$, which can be done by multiplying on the left an invertible matrix to $\mathbf{A}$. Let $u_{1,1}^{-1}$ be the inverse of $u_{1,1}$ on $\mathbb{Z}_{p^\kappa}$. The invertible matrix $\mathbf{L}$ is given by

  $$\mathbf{L} := \begin{bmatrix} u_{1,1}^{-1} & 0 & 0 & \ldots & 0 \\ -u_{1,1}^{-1} \cdot u_{2,1} \cdot p^{e_{2,1}-e_{1,1}} & 1 & 0 & \ldots & 0 \\ -u_{1,1}^{-1} \cdot u_{3,1} \cdot p^{e_{3,1}-e_{1,1}} & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ -u_{1,1}^{-1} \cdot u_{m,1} \cdot p^{e_{m,1}-e_{1,1}} & 0 & 0 & \ldots & 1 \end{bmatrix}.$$

  Then we have $\mathbf{LA} = [p^{e_{1,1}}, 0, 0, \ldots, 0]^T$.

- When $m \geq n > 1$, assume the conclusion holds for $m-1, n-1$. Let

$$(i^*, j^*) := \operatorname*{arg\,min}_{1 \leq i \leq m, 1 \leq j \leq n} e_{i,j}.$$

Then we have $\mathbf{A} \mod p^{e_{i^*,j^*}} = \mathbf{0}$. We can assume $i^* = j^* = 1$ without loss of generality, since when $i^*, j^* \neq 1$, we can swap the i^*th row and the first row by multiplying on the left an invertible matrix to $\mathbf{A}$, and swap the j^*th column and the first column by multiplying on the right an invertible matrix to $\mathbf{A}$.

Let

$$\mathbf{L}_0 := \begin{bmatrix} u_{1,1}^{-1} & 0 & 0 & \dots & 0 \\ -u_{1,1}^{-1} \cdot u_{2,1} \cdot p^{e_{2,1}-e_{1,1}} & 1 & 0 & \dots & 0 \\ -u_{1,1}^{-1} \cdot u_{3,1} \cdot p^{e_{3,1}-e_{1,1}} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ -u_{1,1}^{-1} \cdot u_{m,1} \cdot p^{e_{m,1}-e_{1,1}} & 0 & 0 & \dots & 1 \end{bmatrix}$$

$$\mathbf{R}_0 := \begin{bmatrix} 1 & -u_{1,2} \cdot p^{e_{1,2}-e_{1,1}} & -u_{1,3} \cdot p^{e_{1,3}-e_{1,1}} & \dots & -u_{1,n} \cdot p^{e_{1,n}-e_{1,1}} \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

Let $\alpha_1 = e_{1,1}$. Then we have

$$\mathbf{L}_0 \mathbf{A} \mathbf{R}_0 = \begin{bmatrix} p^{\alpha_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_1 \end{bmatrix}$$

where $\mathbf{A}_1 \in \mathbb{Z}_{p^\kappa}^{(m-1)\times(n-1)}$ satisfies $\mathbf{A}_1 \mod p^{\alpha_1} = \mathbf{0}$.

By assumption, we have there exist invertible matrices $\mathbf{L}_1 \in \mathbb{Z}_{p^\kappa}^{(m-1)\times(m-1)}, \mathbf{R}_1 \in \mathbb{Z}_{p^\kappa}^{(n-1)\times(n-1)}$ such that

$$\mathbf{L}_1 \mathbf{A}_1 \mathbf{R}_1 = \begin{bmatrix} \mathbf{\Lambda}_1' \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{\Lambda}_1' = diag\{p^{\alpha_2}, p^{\alpha_3}, \dots, p^{\alpha_n}\}$, with $0 \leq \alpha_2 \leq \dots \leq \alpha_n \leq \kappa$. Because $\mathbf{A}_1 \mod p^{\alpha_1} = \mathbf{0}$, we have $p^{\alpha_2} \mod p^{\alpha_1} = 0$. So $0 \leq \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_n \leq \kappa$. Let

$$\mathbf{L} := \mathbf{L}_0 \cdot \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_1 \end{bmatrix},$$

$$\mathbf{R} := \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_1 \end{bmatrix} \cdot \mathbf{R}_0,$$

$$\mathbf{\Lambda}' := diag\{p^{\alpha_1}, p^{\alpha_2}, \dots, p^{\alpha_n}\}.$$

Then we have $\mathbf{L}, \mathbf{R}$ are invertible, and

$$\mathbf{LAR} = \begin{bmatrix} \mathbf{\Lambda}' \\ \mathbf{0} \end{bmatrix}.$$

□

Now we show how to solve linear equations over the ring $\mathbb{Z}_{p^\kappa}$.

Lemma 54. *Let p, κ, m, n be any positive integers. There is an algorithm running in time $\mathsf{poly}(n, m, \kappa, \log p)$ that given input $\mathbf{A} \in \mathbb{Z}_{p^\kappa}^{m \times n}$, $\mathbf{b} \in \mathbb{Z}_{p^\kappa}^{m}$, finds the set of solutions of $\mathbf{Ax} = \mathbf{b}$. That is, it finds $\mathcal{X}$ such that*

$$\mathbf{x} \in \mathcal{X} \Leftrightarrow \mathbf{Ax} = \mathbf{b} \bmod p^\kappa.$$

Proof. Use Proposition 53, we can find

$$\mathbf{\Lambda} := \mathbf{LAR} = \begin{bmatrix} \mathbf{\Lambda}' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where $\mathbf{L} \in \mathbb{Z}_{p^\kappa}^{m \times m}, \mathbf{R} \in \mathbb{Z}_{p^\kappa}^{n \times n}$ are invertible, and

$$\mathbf{\Lambda}' = diag\{p^{\alpha_1}, p^{\alpha_2}, \ldots, p^{\alpha_t}\}$$

for some $0 \le t \le \min(m, n)$, $0 \le \alpha_1 \le \alpha_2 \le \cdots \le \alpha_t < \kappa$.

Let $\mathbf{y} = \mathbf{R}^{-1}\mathbf{x}$, $\mathbf{z} = \mathbf{Lb}$. We have $\mathbf{Ax} = \mathbf{b}$ is equivalent to

$$\mathbf{\Lambda y} = \mathbf{z}.$$

Assume $z(i) = v_i \cdot p^{\beta_i}$, where v_i is a unit of $\mathbb{Z}_{p^\kappa}$ and $0 \le \beta_i \le \kappa$ for every $1 \le i \le m$. Let $\alpha_{t+1} = \cdots = \alpha_m = \kappa$.

- If for all $1 \le i \le m$, $\alpha_i \le \beta_i$, we have

  $$\mathcal{Y} := \{(v_1 \cdot p^{\beta_1 - \alpha_1} + c_1 \cdot p^{\kappa - \alpha_1}, \ldots, v_n \cdot p^{\beta_n - \alpha_n} + c_n \cdot p^{\kappa - \alpha_n})^T \mid c_1, \ldots, c_n \in \mathbb{Z}\}$$

  is the solution set of $\mathbf{\Lambda y} = \mathbf{z}$.
- Otherwise, the solution set of $\mathbf{\Lambda y} = \mathbf{z}$ is $\mathcal{Y} := \emptyset$.

Finally, we have

$$\mathcal{X} := \{\mathbf{Ry} \mid \mathbf{y} \in \mathcal{Y}\}$$

is the solution set of $\mathbf{Ax} = \mathbf{b}$. □

Corollary 55. *Let m, n be any positive integers. There is an algorithm running in time $\mathsf{poly}(n, m, \kappa, \log p)$ that given $\mathbf{A} \in \mathbb{Z}_{p^\kappa}^{m \times n}$, $\mathbf{b} \in \mathbb{Z}_{p^\kappa}^{m}$, and an integer $\bar{n} \in [1, n]$, it finds the set $\bar{\mathcal{X}} \subseteq \mathbb{Z}_{p^\kappa}^{\bar{n}}$ such that*

$$\bar{\mathbf{x}} \in \bar{\mathcal{X}} \Leftrightarrow \exists \underline{\mathbf{x}} \in \mathbb{Z}_{p^\kappa}^{n - \bar{n}} \ s.t. \ \mathbf{A} \begin{bmatrix} \bar{\mathbf{x}} \\ \underline{\mathbf{x}} \end{bmatrix} = \mathbf{b}.$$

Proof. As is in the proof of Lemma 54, we find $\{\alpha_i\}_{1 \le i \le m}, \{\beta_i\}_{1 \le i \le m}$, and an invertible matrix $\mathbf{R} \in \mathbb{Z}_{p^\kappa}^{n \times n}$.

- If for all $1 \leq i \leq m$, $\alpha_i \leq \beta_i$, we have

$$\bar{\mathcal{Y}} := \{(v_1 \cdot p^{\beta_1 - \alpha_1} + c_1 \cdot p^{\kappa - \alpha_1}, \ldots, v_{\bar{n}} \cdot p^{\beta_{\bar{n}} - \alpha_{\bar{n}}} + c_{\bar{n}} \cdot p^{\kappa - \alpha_{\bar{n}}})^T \mid c_1, \ldots, c_{\bar{n}} \in \mathbb{Z}\}$$

  is the set of $\bar{\mathbf{y}}$ such that

$$\exists \underline{\mathbf{y}} \in \mathbb{Z}_{p^\kappa}^{n-\bar{n}} \; s.t. \; \mathbf{\Lambda} \begin{bmatrix} \bar{\mathbf{y}} \\ \underline{\mathbf{y}} \end{bmatrix} = \mathbf{z}.$$

- Otherwise, the solution set is $\bar{\mathcal{Y}} := \emptyset$.

We divide $\mathbf{R}$ into two parts, that is, find $\bar{\mathbf{R}} \in \mathbb{Z}_{p^\kappa}^{\bar{n} \times n}$ and $\underline{\mathbf{R}} \in \mathbb{Z}_{p^\kappa}^{(n-\bar{n}) \times n}$ such that $\mathbf{R} = \begin{bmatrix} \bar{\mathbf{R}} \\ \underline{\mathbf{R}} \end{bmatrix}$. Then we have

$$\bar{\mathcal{X}} := \{\bar{\mathbf{R}}\bar{\mathbf{y}} \mid \bar{\mathbf{y}} \in \bar{\mathcal{Y}}\}$$

is the set of $\bar{\mathbf{x}}$ such that

$$\exists \underline{\mathbf{x}} \in \mathbb{Z}_{p^\kappa}^{n-\bar{n}} \; s.t. \; \mathbf{A} \begin{bmatrix} \bar{\mathbf{x}} \\ \underline{\mathbf{x}} \end{bmatrix} = \mathbf{b}.$$

□

B $\mathsf{NC}[q]$, Polynomials and Our wPRFs

In this section, we define the circuit class $\mathsf{NC}[q]$ and present a generic approach for computing $\mathsf{NC}[q]$ circuits using polynomials and for computing specific functions using $\mathsf{NC}[q]$ circuits. In Appendix B.5, we compute the explicit depth of our weak PRF candidates using $\mathsf{NC}^0[q_1]$ circuits. In Appendix B.6, we compute the depth of our weak PRF candidates in various circuit classes, results in different depth. This demonstrates the impact of the parameters and the MOD gate we use.

B.1 Definitions

For any integer $d \geq 0$, a circuit C is in $\mathsf{NC}^d[\mathsf{MOD}_{2,\mathsf{poly}}]$, abbreviated as $\mathsf{NC}^d[2]$, if it is of depth $O(\log^d n)$ and consists of the following gates: AND with fan-in 2, OR with fan-in 2, NOT, and MOD_2 with polynomial fan-in (namely, on input $\mathbf{x} \in \{0,1\}^s$, where $s \in \mathsf{poly}(n)$, $\mathsf{MOD}_2(\mathbf{x})$ outputs $\sum_{i=1}^{s} x_i \bmod 2$). Sometimes we omit d in the notation $\mathsf{NC}[2]$ and explicitly mention the depth instead. Similarly, we define $\mathsf{NC}^d[\mathsf{MOD}_{\mathsf{q},\mathsf{poly}}]$ (abbreviated as $\mathsf{NC}^d[q]$) with an additional polynomial fan-in MOD_q gate (on input $\mathbf{x} \in \{0,1\}^n$, output 0 if and only if the number of 1s in $\mathbf{x}$ is multiple of q).

For any polynomial f, $D(f)$ denotes the degree of f, and $M(f)$ denotes the number of monomials in the polynomial f. For an non-negative integer x, we use $\mathsf{Bin}(\mathsf{x})$ to denote the binary representation of x, from the least significant bit to the most significant bit. For example, $\mathsf{Bin}(6) = 011$. For $k \in \mathbb{N}^+$ and an integer $0 \leq x < 2^k$, we use $\mathsf{Bin}_k(x) \in \{0,1\}^k$ to denote the binary representation with fixed length k. That is, when $x < 2^{k-1}$, we pad some 0s after $\mathsf{Bin}(x)$ to get $\mathsf{Bin}_k(x)$ (for example, $\mathsf{Bin}_4(5) = 1010$). For a binary representation $\mathbf{y}$, we use $V(\mathbf{y})$ to denote the integer value of $\mathbf{y}$ (for example, $V(101) = 5$). When computing a function using $\mathsf{NC}[q]$, we represent an integer in $\mathbb{Z}_q$ using $\lceil \log q \rceil$ bits by its binary representation, and we use $\mathcal{B}_q$ to denote the set $\{\mathsf{Bin}_{\lceil \log q \rceil}(x) | x \in \mathbb{Z}, 0 \leq x < q\}$. That is, we use $\mathcal{B}_q$ to denote the set consisting of binary representations of elements in $\mathbb{Z}_q$.

B.2 The relationship between $\mathsf{NC}^0[q]$ and $\mathsf{CC}^0[q]$

$\mathsf{CC}^0[q]$ is the class of constant-depth and polynomial-size circuits consisting of unbounded fan-in MOD_q gates and NOT gate. $\mathsf{NC}^0[q]$ has more gates than $\mathsf{CC}^0[q]$, indicating that $\mathsf{CC}^0[q] \subseteq \mathsf{NC}^0[q]$.

When $q > 2$, we have the following theorem characterizing the relationship between the two complexity classes.

Theorem 56. *If $q > 2$, then $\mathsf{NC}^0[q] = \mathsf{CC}^0[q]$.*

Proof. We only need to prove that 2-fan-in AND and OR gates are in $\mathsf{CC}^0[q]$. These two gates are implemented as follows:

- $\mathrm{AND}(x_1, x_2)$: compute
$$\neg\mathsf{MOD}_q(x_1, x_2, \underbrace{1, ..., 1}_{q-2}).$$
The above MOD_q outputs 0 if and only if both x_1 and x_2 are 1, thus the above circuit achieves the functionality of $\mathrm{AND}(x_1, x_2)$.
- $\mathrm{OR}(x_1, x_2)$: note that
$$\mathrm{OR}(x_1, x_2) = \neg\mathrm{AND}(\neg x_1, \neg x_2).$$

□

When $q = 2$, AND can not be implemented by MOD_2, thus $\mathsf{NC}^0[2]$ is strictly larger than $\mathsf{CC}^0[2]$.

B.3 How to Compute $\mathsf{NC}[2]$ Circuits Using Low-Degree Polynomials

Theorem 57. *For any circuit C in $\mathsf{NC}[2]$ of depth k with inputs $x_1, x_2, \cdots, x_n$, there exists a function $h(\cdot) \in \mathsf{poly}(n)$ and a $GF(2)$ polynomial $f(x_1, x_2, \cdots, x_n)$ with $D(f) \leq 2^k$, $M(f) \leq (h(n)+2)^{2^{k-1}} - 2$, such that*
$$f(x_1, x_2, \cdots, x_n) = C(x_1, x_2, \cdots, x_n).$$

Remark: The upper bound of $M(f)$ is the value of the following sequence $\{a_k\}$, where $a_0 := 1, a_1 := h(n)$, and
$$\forall k \geq 2,\ a_k := (a_{k-1} + 2)^2 - 2.$$

Proof. Let $h(n)$ be the upper bound for the number of fan-in of the MOD_2 gate. We prove by induction on k. For $k = 0$, We have $f(x_1, x_2, \cdots, x_n) = x_i$ for some i or $f(x_1, x_2, \cdots, x_n) = 0, 1$, thus $D(f) \leq 1$, $M(f) \leq 1$. Suppose that the theorem holds for $k-1$, we consider the case of k. We consider the output gate of C, for which there are four cases.

1. **AND.** Suppose that the output gate of C is AND, then C can be expressed as $C = C_1 \wedge C_2$, where C_1, C_2 are circuits of depth no larger than $k-1$. Then, by the induction assumption, there are polynomials f_1, f_2 corresponding to C_1, C_2, such that
$$f = f_1 f_2.$$

Then

$$D(f) = D(f_1) + D(f_2) \leq 2^k,$$

$$M(f) \leq M(f_1)M(f_2) \leq (M(f_1)+1)(M(f_2)+1)+1 \leq (h(n)+2)^{2^{k-1}} - 2.$$

2. **OR.** Suppose that the output gate of C is OR, then C can be expressed as $C = C_1 \vee C_2$, where C_1, C_2 are circuits of depth no larger than $k-1$. Then, by the induction assumption, there are polynomials f_1, f_2 corresponding to C_1, C_2, such that

$$f = 1 - (1-f_1)(1-f_2).$$

Then

$$D(f) = D(f_1) + D(f_2) \leq 2^k,$$

$$M(f) \leq (M(f_1)+1)(M(f_2)+1)+1 \leq (h(n)+2)^{2^{k-1}} - 2.$$

3. **NOT.** Suppose that the output gate of C is NOT, then C can be expressed as $C = \neg C_1$, where C_1 is circuit of depth no larger than $k-1$. Then there exists a polynomial f_1 corresponding to C_1 such that

$$f = 1 - f_1.$$

Then

$$D(f) = D(f_1) \leq 2^{k-1},$$

$$M(f) \leq M(f_1) + 1 \leq (h(n)+2)^{2^{k-1}} - 2.$$

4. **MOD_2.** Suppose that the output gate of C is MOD_2, then C can be expressed as $C = C_1 \oplus C_2 \oplus \cdots \oplus C_s$, where C_i are circuits of depth no larger than $k-1$, and $s \leq h(n)$. Then there exists polynomials $f_1, f_2, \cdots, f_s$ corresponding to $C_1, C_2, \cdots, C_s$, such that

$$f = f_1 + f_2 + \cdots + f_s.$$

Then

$$D(f) = \max_i D(f_i) \leq 2^{k-1},$$

$$M(f) \leq \sum_{i=1}^{s} M(f_i) \leq h(n)((h(n)+2)^{2^{k-2}} - 2) \leq (h(n)+2)^{2^{k-1}} - 2.$$

□

B.4 How to Compute $\mathsf{NC}[p]$ Circuits Using Low-Degree Polynomials

Theorem 58. *Let $p \geq 3$. For any Boolean circuit C in $\mathsf{NC}[p]$ of depth k with inputs $x_1, x_2, \cdots, x_n \in \{0,1\}^n$, there exists a function $h(\cdot) \in \mathsf{poly}(n)$ and a $GF(p)$ polynomial $f(x_1, x_2, \cdots, x_n)$ with $D(f) \leq (p-1)^k$, $M(f) \leq a_k$, such that*

$$f(x_1, x_2, \cdots, x_n) = C(x_1, x_2, \cdots, x_n).$$

Here a_k satisfies $a_0 := 1$, and for $k \geq 0$,

$$a_{k+1} := (h(n)a_k + 1)^{p-1} + 1.$$

Remark: We can derive an upper bound for a_k using the following fact:

$$a_{k+1} \le (h(n)a_k + h(n))^{p-1} - 1 \quad \Rightarrow \quad a_{k+1} + 1 \le (h(n)(a_k+1))^{p-1}.$$

Solve this recursive relation, we get

$$a_k \le (h(n))^{-(p-1)/(p-2)} \left(2h(n)^{(p-1)/(p-2)}\right)^{(p-1)^k} - 1.$$

For constant k, a_k is polynomial in n.

Proof. Let $h(n)$ be the upper bound for the number of fan-in of the MOD_p gate. We prove by induction on k. For $k = 0$, We have $f(x_1, x_2, \cdots, x_n) = x_i$ for some i or $f(x_1, x_2, \cdots, x_n) = 0, 1$, thus $D(f) \le 1$, $M(f) \le 1$. Suppose that the theorem holds for $k-1$, we consider the case of k. We consider the output gate of C, for which there are 4 cases.

1. **AND.** Suppose the output gate of C is AND, then C can be expressed as $C = C_1 \wedge C_2$, where C_1, C_2 are circuits of depth no larger than $k-1$. Then, by the induction assumption, there are polynomials f_1, f_2 corresponding to C_1, C_2, such that

   $$f = f_1 f_2.$$

   Then

   $$D(f) = D(f_1) + D(f_2) \le (p-1)^k,$$
   $$M(f) \le M(f_1)M(f_2) \le a_{k-1}^2 \le a_k.$$

2. **OR.** Suppose that the output gate of C is OR, then C can be expressed as $C = C_1 \vee C_2$, where C_1, C_2 are circuits of depth no larger than $k-1$. Then, by the induction assumption, there are polynomials f_1, f_2 corresponding to C_1, C_2, such that

   $$f = 1 - (1 - f_1)(1 - f_2).$$

   Then

   $$D(f) = D(f_1) + D(f_2) \le (p-1)^k,$$
   $$M(f) \le (M(f_1)+1)(M(f_2)+1) + 1 \le (a_{k-1}+1)^2 + 1 \le a_k.$$

3. **NOT.** Suppose that the output gate of C is NOT, then C can be expressed as $C = \neg C_1$, where C_1 is circuit of depth no larger than $k-1$. Then there exists a polynomial f_1 corresponding to C_1 such that

   $$f = 1 - f_1.$$

   Then

   $$D(f) = D(f_1) \le (p-1)^{k-1},$$
   $$M(f) \le M(f_1) + 1 \le a_{k-1} + 1 \le a_k.$$

4. **MOD_p.** Suppose the output gate of C is MOD_p, then C can be expressed as $C = \mathsf{MOD}_\mathsf{p}(C_1, C_2, \cdots, C_s)$, where C_i are circuits of depth no larger than $k-1$, and $s \leq h(n)$. Then by the induction assumption, there exists polynomials $f_1, f_2, \cdots, f_s$ corresponding to $C_i, 1 \leq i \leq h(n)$, such that

$$f^* = f_1 + f_2 + \cdots + f_s,$$
$$f = 1 + \prod_{i=1}^{p-1}(i - f^*).$$

Now we prove that f computes MOD_p correctly. When $f^* \equiv i \pmod p$ for some $1 \leq i \leq p-1$, we have $f \equiv 1 \pmod p$; when $f^* \equiv 0 \pmod p$, by Wilson's theorem, we have $f \equiv 1+(p-1)! \equiv 1-1=0$. Note that Wilson's theorem requires p to be a prime, and that is why our theorem only applies to prime p.

Finally, we bound $D(f)$ and $M(f)$ as follows:

$$D(f) \leq (p-1)\max_i D(f_i) \leq (p-1)^k,$$

$$M(f) \leq (\sum_{i=1}^{s} M(f_i)+1)^{p-1}+1 \leq (h(n)a_{k-1}+1)^{p-1}+1 = a_k.$$

□

B.5 The Explicit Depth of Our wPRFs

In this section, we show (in Theorem 60) how to compute the wPRF candidates $g_\mathbf{S}(\mathbf{x})$ and $L_\mathbf{S}(\mathbf{x})$ in $\mathsf{NC}^0[q_1]$ with depth 3.

Lemma 59. *For any positive integers q_1, q_2 such that $q_1 | q_2$, $\mathsf{NC}^0[q_1] \subseteq \mathsf{NC}^0[q_2]$.*

Proof. We only need to show that $\mathsf{MOD}_{q_1} \in \mathsf{NC}^0[q_2]$. We can compute MOD_{q_1} using MOD_{q_2} by repeating each input for $\frac{q_2}{q_1}$ times. That is,

$$\mathsf{MOD}_{q_1}(x_1, x_2, \cdots, x_n) = \mathsf{MOD}_{q_2}(x_1, x_1, \cdots, x_1, x_2, x_2, \cdots, x_2, ..., x_n, x_n, \cdots, x_n),$$

where each x_i appears for $\frac{q_2}{q_1}$ times in the input of MOD_{q_2}. □

Theorem 60. *For any q_1, q_2, p such that $q_1 \in \mathsf{poly}(n)$, $p | q_1$, $q_1 > q_2$, $gcd(q_1, q_2) = 1$, the functions $g_\mathbf{S}(\mathbf{x})$ and $L_\mathbf{S}(\mathbf{x})$ can be computed in $\mathsf{NC}^0[q_1]$ with depth 3.*

Proof. We prove only for $g_\mathbf{S}$. The proof for $L_\mathbf{S}$ is similar. Recall the weak PRF family $\mathcal{G} := \{g_\mathbf{S} : \mathbb{Z}_{q_1}^n \to \mathbb{Z}_p\}_{\mathbf{S} \in \mathbb{Z}_{q_1}^{n \times \ell}}$, where p is a factor of q_1:

$$g_\mathbf{S}(\mathbf{x}) := \left(\sum_{i=1}^{\ell} ((\langle \mathbf{s}_i, \mathbf{x} \rangle \bmod q_1) \bmod q_2) \right) \bmod p. \tag{34}$$

The circuit consists of three layers. We show explicitly the implementation of these three layers using the MOD_{q_1} gate. Note that if p is a factor of q_1, MOD_p gate can be implemented using MOD_{q_1} (that is, $\mathsf{MOD}_p(\mathbf{x}) = \mathsf{MOD}_{q_1}(\underbrace{\mathbf{x}, \mathbf{x} \cdots, \mathbf{x}}_{q_1/p \text{ times}})$).

Layer 1:

- Input: $\mathbf{x}$ in the binary representation.
- Output: for each $y \in \mathbb{Z}_{q_1}$ and $i = 1, 2, \cdots, \ell$, output a bit

$$\Delta_y(\langle \mathbf{s}_i, \mathbf{x} \rangle) := \mathsf{MOD}_{q_1}\big(\mathsf{unary}(\langle \mathbf{s}_i, \mathbf{x} \rangle + q_1 - y)\big),$$

  where $\mathsf{unary}(k) = \underbrace{111 \cdots 111}_{\text{repeat } k \text{ times}}$.
- Functionality: For each y and i, $\Delta_y(\langle \mathbf{s}_i, \mathbf{x} \rangle)$ determines whether

$$\langle \mathbf{s}_i, \mathbf{x} \rangle \bmod q_1 = y,$$

  which equals 0 if true or 1 otherwise.
- Implementation: For each y and i, $\Delta_y(\langle \mathbf{s}_i, \mathbf{x} \rangle)$ is computed by a MOD_{q_1} gate with the following input bits:
  - For all $j = 1, 2, \cdots, n$, $k = 1, 2, \cdots, \lceil \log(q_1) \rceil$, let $s_{i,j}$ be the j-th entry of $\mathbf{s}_i$, and $x_{j,k}$ be the k-th entry of $\mathsf{Bin}_{\lceil \log(q_1) \rceil}(x_j)$. Repeat $x_{j,k}$ for $s_{i,j} 2^{k-1}$ times. (This is by $\langle \mathbf{s}_i, \mathbf{x} \rangle = \sum_{j=1}^{n} \sum_{k=1}^{\lceil \log(q_1) \rceil} s_{i,j} 2^{k-1} x_{j,k}$.)
  - Repeat 1 for $(q_1 - y)$ times.

Layer 2:

- Input: $\Delta_y(\langle \mathbf{s}_i, \mathbf{x} \rangle)$, for every $y \in \mathbb{Z}_{q_1}$ and $i = 1, 2, \cdots, \ell$.
- Output: For each $z \in \mathbb{Z}_p$, output a bit

$$\Delta'_z := \mathsf{MOD}_p\Big(\mathsf{unary}\big(p - z + \sum_{i=1}^{\ell} \sum_{y \in \mathbb{Z}_{q_1}} \big(1 - \Delta_y(\langle \mathbf{s}_i, \mathbf{x} \rangle)\big)\big((y \bmod q_2) \bmod p\big)\big)\Big).$$

- Functionality: For each $z \in \mathbb{Z}_p$, Δ'_z determines whether

$$\sum_{i=1}^{\ell} \big((\langle \mathbf{s}_i, \mathbf{x} \rangle \bmod q_1) \bmod q_2\big) \bmod p = z,$$

  which equals 0 if true or 1 otherwise.
- Implementation: For each $z \in \mathbb{Z}_p$, Δ'_z is computed by a MOD_p gate with the following input bits:
  - For all $i = 1, 2, \cdots, \ell, y \in \mathbb{Z}_{q_1}$, repeat 1 for $(y \bmod q_2) \bmod p$ times.
  - For all $i = 1, 2, \cdots, \ell, y \in \mathbb{Z}_{q_1}$, repeat $\Delta_y(\langle \mathbf{s}_i, \mathbf{x} \rangle)$ for $(p-1)\big((y \bmod q_2) \bmod p\big)$ times.
  - Repeat 1 for $p - z$ times.

Layer 3:

- Input: Δ'_z for all $z \in \mathbb{Z}_p$.

- Output: the binary representation of

$$g_{\mathbf{S}}(\mathbf{x}) = \sum_{z \in \mathbb{Z}_p} z \cdot (1 - \Delta'_z) \bmod p.$$

- Functionality: On input $\{\Delta'_z\}_{z \in \mathbb{Z}_p}$, this layer find the only $z^* \in \mathbb{Z}_p$ such that $\Delta'_{z^*} = 0$, and output $g_{\mathbf{S}}(\mathbf{x}) = \mathsf{Bin}_{\lceil \log(p) \rceil}(z^*)$. Concretely, for each $k = 1, 2, \cdots, \lceil \log(p) \rceil$, the k-th bit of $\mathsf{Bin}_{\lceil \log(p) \rceil}(g_{\mathbf{S}}(\mathbf{x}))$ can be computed by

$$\sum_{z \in \mathbb{Z}_p} z_k \cdot (1 - \Delta'_z) \bmod p,$$

  where we abuse the notation that let z_k denote the k-th bit of $\mathsf{Bin}(z)$.
- Implementation: For each $k = 1, 2, \cdots, \lceil \log(p) \rceil$, the k-th output bit is computed by a MOD_p gate (by Lemma 59, this can be realized by a MOD_{q_1} gate because $p | q_1$) with the following input bits:
  - For all $z \in \mathbb{Z}_p$, a bit z_k.
  - For all $z \in \mathbb{Z}_p$, if $z_k = 1$, repeat Δ'_z for $(p-1)$ times.
  - Repeat 1 for $p - 1$ times. (Since the sum is either 0 or 1, this flips the output of MOD_p gate.)

□

B.6 The Depth of Our wPRFs in Other Settings.

The depth of $g_{\mathbf{S}}(\mathbf{x})$ when $p = 2$. In the implementation of $g_{\mathbf{S}}(\mathbf{x})$, we need an additional layer to compute the output value. However, when $p = 2$, the function value $g_{\mathbf{S}}(\mathbf{x}) = \Delta'_0$. Therefore, we can compute the function in $\mathsf{NC}^0[q_1]$ with depth 2. As an example, the parameters can be selected as $q_1 = 24, q_2 = 5, p = 2$, and the function is:

$$g_{\mathbf{S}}(\mathbf{x}) := \left(\sum_{i=1}^{\ell} ((\langle \mathbf{s}_i, \mathbf{x} \rangle \bmod 24) \bmod 5) \right) \bmod 2. \tag{35}$$

The depth of $g_{\mathbf{S}}(\mathbf{x})$ in $\mathsf{NC}^0[p_1, p_2]$ when $q_1 = p_1 p_2$, p is a factor of p_1 and $\gcd(p_1, p_2) = 1$. Provided that p_1 and p_2 are coprime and $q_1 = p_1 p_2$, the MOD_{q_1} gate can be simulated in the circuit class $\mathsf{NC}^0[p_1, p_2]$ by the equivalence $\mathsf{MOD}_{q_1}(\mathbf{x}) = \mathsf{MOD}_{p_1}(\mathbf{x}) \vee \mathsf{MOD}_{p_2}(\mathbf{x})$.

To compute the function $g_{\mathbf{S}}(\mathbf{x})$ in $\mathsf{NC}^0[p_1, p_2]$, we only need to replace the MOD_{q_1} gate in the first layer by $\mathsf{MOD}_{p_1} \vee \mathsf{MOD}_{p_2}$. This increases the depth by 1. Therefore, the depth of $g_{\mathbf{S}}(\mathbf{x})$ in $\mathsf{NC}^0[p_1, p_2]$ is 4.

The depth of the wPRF candidate $K_{\mathbf{s}}(\mathbf{x})$. We show that the function $K_{\mathbf{s}}(\mathbf{x})$ can be computed in $\mathsf{NC}^0[q_1]$ with depth 2. Since the first layer is the same as $g_{\mathbf{S}}(\mathbf{x})$, we only show how to construct the second layer.

- Input: $\Delta_y(\langle \mathbf{s}, \mathbf{x} \rangle)$, for all $y \in \mathbb{Z}_{q_1}$.

- Output: the binary representation of

$$K_{\mathbf{s}}(\mathbf{x}) = \sum_{y \in \mathbb{Z}_{q_1}} (1 - \Delta_y(\langle \mathbf{s}, \mathbf{x} \rangle)) \cdot (\lfloor q_2(y \bmod q_1)/q_1 \rceil \bmod q_2).$$

- Functionality: On input $\{\Delta_y(\langle \mathbf{s}, \mathbf{x} \rangle)\}_{y \in \mathbb{Z}_{q_1}}$, this layer find the only $y^* \in \mathbb{Z}_{q_1}$ such that $\Delta_{y^*}(\langle \mathbf{s}, \mathbf{x} \rangle) = 0$, and output $K_{\mathbf{s}}(\mathbf{x}) = \lfloor q_2(y^* \bmod q_1)/q_1 \rceil \bmod q_2$. Concretely, for each $k = 1, 2, \cdots, \lceil \log(q_2) \rceil$, the k-th bit of $\mathsf{Bin}_{\lceil \log(q_2) \rceil}(K_{\mathbf{S}}(\mathbf{x}))$ can be computed by

$$\sum_{y \in \mathbb{Z}_{q_1}} (1 - \Delta_y(\langle \mathbf{s}, \mathbf{x} \rangle)) \cdot \mathsf{Bin}^{(k)}(\lfloor q_2(y \bmod q_1)/q_1 \rceil \bmod q_2),$$

  where we abuse the notation that let $\mathsf{Bin}^{(k)}(t)$ denote the k-th bit of $\mathsf{Bin}_{\lceil \log(q_2) \rceil}(t)$ for any $t \in \mathbb{Z}_{q_2}$.

- Implementation: For each $k = 1, 2, \cdots, \lceil \log(q_2) \rceil$, the k-th output bit is computed by a MOD_{q_1} gate with the following input bits:
  - For all $y \in \mathbb{Z}_{q_1}$, a bit $\mathsf{Bin}^{(k)}(\lfloor q_2(y \bmod q_1)/q_1 \rceil \bmod q_2)$.
  - For all $y \in \mathbb{Z}_{q_1}$, if $\mathsf{Bin}^{(k)}(\lfloor q_2(y \bmod q_1)/q_1 \rceil \bmod q_2) = 1$, repeat $\Delta'_y(\langle \mathbf{s}, \mathbf{x} \rangle)$ for $(q_1 - 1)$ times.
  - Repeat 1 for $q_1 - 1$ times. (Since the number of bits in the first two rows listed above is either 0 or 1 when mod q_1, this flips the output of MOD_{q_1} gate.)

B.7 How to Reduce $\mathsf{NC}[q]$ for Composite q to Smaller Moduli

We first show that for $q = p^k$, $\mathsf{NC}^0[q] = \mathsf{NC}^0[p]$. Let us start from showing how to implement MOD_{p^k} using MOD_p.[3]

Definition 61 (Rounding by p)**.** *Define* Rby_p*:*$\{0,1\}^n \to \{0,1\}^n$ *as follows: Given an input* $\mathbf{x} \in \{0,1\}^n$ *such that the number of 1s in* $\mathbf{x}$ *is* t*,* Rby_p *outputs a vector* $\mathbf{y} \in \{0,1\}^n$ *such that the number of 1s in* $\mathbf{y}$ *is* $\lfloor t/p \rfloor$*.*

Rby_p is useful for providing the carries when computing modulo-p^k additions. Rby_p can be implemented as follows. Define $\Sigma_0 := 0$. For $i = 1, \ldots, n$, define

$$\Sigma_i := \mathsf{MOD}_p(x_1, x_2, \cdots, x_i).$$

Let $\mathbf{y} := (y_1, \ldots, y_n)$, where

$$y_i := \Sigma_{i-1} \wedge (\neg \Sigma_i).$$

This realizes the Rby_p functionality with a depth-3 $\mathsf{NC}[p]$ circuit. The following lemma shows its correctness.

Lemma 62 (Correctness of Rby_p)**.** $\mathbf{y}$ *defined above satisfies: the number of 1s in* $\mathbf{y}$ *is* $\lfloor t/p \rfloor$*.*

[3]This is a generalization of the reduction in $\mathsf{AC}^0[p]$ [Smo87], similar idea can be found in the following website: https://cs.stackexchange.com/questions/56343/computing-mod-4-function-using-mod-2-or-and-not-gates

Proof. Define $\Gamma_0 := 0$. For $i = 1, \ldots, n$, define

$$\Gamma_i := (x_1 + x_2 + \cdots + x_i) \bmod p.$$

For $l \in [p]$, let $S_l := \{j | 1 \le j \le n, \Gamma_{j-1} + 1 \equiv \Gamma_j \equiv i \pmod p\}$. Then, since we compute y_i by $\Sigma_{i-1} \wedge (\neg \Sigma_i)$, the number of 1s in $\mathbf{y}$ equals the size of S_0. Let $\mathcal{S} = \cup_{i=0}^{p-1} S_i$. We know that $x_i = 1$ if and only if $\Gamma_{i-1} + 1 \equiv \Gamma_i \pmod p$, therefore $|\mathcal{S}| = t$. Let $i_1 < i_2 < \cdots < i_k$ be all elements in $\mathcal{S}$. By induction, we have $i_\ell \in S_{\ell \bmod q}$. Then we have the following relations:

$$\sum_{i=0}^{p-1} |S_i| = |\mathcal{S}| = t, \tag{36}$$

$$|S_1| \ge |S_2| \ge \cdots \ge |S_{p-1}| \ge |S_0| \ge |S_1| - 1. \tag{37}$$

By (36) and (37), we get $|S_0| = \lfloor t/p \rfloor$. □

Lemma 63. *Let k be any positive integer. MOD_{p^k} can be computed in $\mathsf{NC}[p]$ with depth $\le 3k$.*

Proof. Let $x_1, x_2, \cdots, x_n$ denote the input, and suppose the number of 1s in the input is t. We sequentially compute the sequences $\mathbf{Y}_i$ (for $i = 1, 2, \cdots, k$) as follows:

- $\mathbf{Y}_1 = (x_1, x_2, \cdots, x_n)$.
- $\mathbf{Y}_i = \mathsf{Rby}_p(\mathbf{Y}_{i-1})$.

We know $\mathbf{Y}_i$ can be computed in $\mathsf{NC}[p]$ within depth $3i - 3$, and by the functionality of Rby_p, the number if 1s in $\mathbf{Y}_i$ is $\lfloor t/p^{i-1} \rfloor$. Thus, the functionality MOD_{p^k} can be computed as follows:

$$\mathsf{MOD}_{p^k}(x_1, x_2, \cdots, x_n) = \vee_{i=1}^{k} \mathsf{MOD}_p(\mathbf{Y}_i).$$

□

Now we have the following facts:

1. For any integers q_1, q_2 such that $q_2 | q_1$, $\mathsf{NC}^0[q_2] \subseteq \mathsf{NC}^0[q_1]$ (by Lemma 59).
2. For any integer p and constant k, $\mathsf{NC}^0[p^k] = \mathsf{NC}^0[p]$ (by Lemma 63).
3. For any integers q_1, q_2 such that $\gcd(q_1, q_2) = 1$, $\mathsf{NC}^0[q_1 q_2] = \mathsf{NC}^0[q_1, q_2]$. This follows from Lemma 59 and the fact that $\mathsf{MOD}_{q_1 q_2} = \neg((\neg \mathsf{MOD}_{q_1}) \wedge (\neg \mathsf{MOD}_{q_2}))$.

Therefore, for constant positive integers $\alpha_1, \alpha_2, \cdots, \alpha_k$ and distinct primes $p_1, p_2, \cdots, p_k$, we have the following relations:

$$\mathsf{NC}^0[p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}] = \mathsf{NC}^0[p_1^{\alpha_1}, p_2^{\alpha_2}, \cdots, p_k^{\alpha_k}] = \mathsf{NC}^0[p_1, p_2, \cdots, p_k].$$

C Other Omitted Proofs

C.1 Proof of Lemma 13

Proof. Let α be the smallest integer such that $p^\alpha \cdot \mathbf{x} \equiv \mathbf{0} \pmod{p^\kappa}$.

- If $p^\alpha \cdot b \not\equiv 0 \pmod{p^\kappa}$, we have

$$\begin{aligned}\Pr_{\mathbf{a}\leftarrow\mathbb{Z}_{p^\kappa}^n}[\langle \mathbf{a}, \mathbf{x}\rangle \not\equiv b \pmod{p^\kappa}] &\geq \Pr_{\mathbf{a}\leftarrow\mathbb{Z}_{p^\kappa}^n}[p^\alpha \cdot \langle \mathbf{a}, \mathbf{x}\rangle \not\equiv p^\alpha \cdot b \pmod{p^\kappa}] \\ &= \Pr_{\mathbf{a}\leftarrow\mathbb{Z}_{p^\kappa}^n}[0 \not\equiv p^\alpha \cdot b \pmod{p^\kappa}] \\ &= 1\end{aligned}$$

- If $p^\alpha \cdot b \equiv 0 \pmod{p^\kappa}$, then we may write $b \equiv p^{\kappa-\alpha} \cdot b' \pmod{p^\kappa}$. Because $\mathbf{x} \neq \mathbf{0}$, we have $\alpha \geq 1$, and there exists some $\mathbf{y} \in \mathbb{Z}_{p^\kappa}$ such that $\mathbf{x} = p^{\kappa-\alpha} \cdot \mathbf{y}$ and $\mathbf{y} \mod p \neq \mathbf{0}$. Let $j^* \in [1, n]$ be some index for which $y(j^*) \mod p \neq 0$. Then

$$\begin{aligned}\Pr_{\mathbf{a}\leftarrow\mathbb{Z}_{p^\kappa}^n}[\langle \mathbf{a}, \mathbf{x}\rangle \not\equiv b \pmod{p^\kappa}] &= \Pr_{\mathbf{a}\leftarrow\mathbb{Z}_{p^\kappa}^n}[p^{\kappa-\alpha}\langle \mathbf{a}, \mathbf{y}\rangle \not\equiv p^{\kappa-\alpha} \cdot b' \pmod{p^\kappa}] \\ &\geq \Pr_{\mathbf{a}\leftarrow\mathbb{Z}_{p^\kappa}^n}[p^{\kappa-1}\langle \mathbf{a}, \mathbf{y}\rangle \not\equiv p^{\kappa-1} \cdot b' \pmod{p^\kappa}] \\ &= \Pr_{\mathbf{a}\leftarrow\mathbb{Z}_{p^\kappa}^n}[\langle \mathbf{a}, \mathbf{y}\rangle \not\equiv b' \pmod{p}] \\ &= \Pr_{\mathbf{a}\leftarrow\mathbb{Z}_{p^\kappa}^n}[a(j^*)y(j^*) \not\equiv b' - \sum_{\substack{1\leq j\leq n \\ j\neq j^*}} a(j)y(j) \pmod{p}] \\ &= 1 - 1/p,\end{aligned}$$

  where the last "=" is due to

$$\forall r \in \mathbb{Z},\ \Pr_{a(j^*)\leftarrow\mathbb{Z}_{p^\kappa}}[a(j^*)y(j^*) \not\equiv r \pmod{p}] = 1 - 1/p.$$

□

C.2 Proof of Lemma 14

Proof. For any $\mathbf{x}' \in \mathbb{Z}_q^n$ such that $\mathbf{x}' \neq \mathbf{x}$, let $\Delta\mathbf{x} = \mathbf{x}' - \mathbf{x} \neq \mathbf{0}$. We have

$$\begin{aligned}
&\Pr_{\{\mathbf{a}_i\}_i}[\forall 1 \leq i \leq m,\ \langle \mathbf{a}_i, \mathbf{x}' \rangle = b_i] \\
=&\Pr_{\{\mathbf{a}_i\}_i}[\forall 1 \leq i \leq m,\ \langle \mathbf{a}_i, \mathbf{x}' \rangle = \langle \mathbf{a}_i, \mathbf{x} \rangle] \\
=&\Pr_{\{\mathbf{a}_i\}_i}[\forall 1 \leq i \leq m,\ \langle \mathbf{a}_i, \Delta\mathbf{x} \rangle = 0] \\
=&\prod_{i=1}^{m} \Pr_{\mathbf{a}_i}[\langle \mathbf{a}_i, \Delta\mathbf{x} \rangle = 0] \\
=&\prod_{i=1}^{m} \prod_{j=1}^{\ell} \Pr_{\mathbf{a}_i}[\langle \mathbf{a}_i, \Delta\mathbf{x} \rangle \mod p_j^{\kappa_j} = 0] \\
\leq&\prod_{i=1}^{m} \prod_{j=1}^{\ell} (1/p_j) \\
\leq&2^{-\ell m},
\end{aligned}$$

where the fourth "=" uses the Chinese Remainder Theorem, and the first "$\leq$" uses Lemma 13. Then,

$$\begin{aligned}
&\Pr_{\{\mathbf{a}_i\}_i}[\exists \mathbf{x}' \in \mathbb{Z}_q^n,\ (\mathbf{x}' \neq \mathbf{x}) \wedge (\forall 1 \leq i \leq m,\ \langle \mathbf{a}_i, \mathbf{x}' \rangle = b_i)] \\
\leq& \sum_{\mathbf{x}' \in \mathbb{Z}_q^n,\ \mathbf{x}' \neq \mathbf{x}} \Pr_{\{\mathbf{a}_i\}_i}[\forall 1 \leq i \leq m,\ \langle \mathbf{a}_i, \mathbf{x}' \rangle = b_i] \\
\leq& \sum_{\mathbf{x}' \in \mathbb{Z}_q^n,\ \mathbf{x}' \neq \mathbf{x}} 2^{-\ell m} \\
<& q^n / 2^{\ell m}
\end{aligned}$$

□

C.3 Proof of Lemma 15

Proof. For any prime factor p of q, we have $\Pr_{(x_1,\ldots,x_n)\in\mathbb{Z}_q^n}[\forall 1 \le i \le n,\ p|x_i] = 1/p^n$. Then, for any $n \ge 2$, by union bound,

$$
\begin{aligned}
&\Pr_{(x_1,\ldots,x_n)\in\mathbb{Z}_q^n}[\gcd(q, x_1, x_2, \cdots, x_n) \neq 1] \\
\le &\sum_{p|q,\ p \text{ prime}} \Pr_{(x_1,\ldots,x_n)\in\mathbb{Z}_q^n}[\forall 1 \le i \le n,\ p|x_i] \\
\le &\sum_{p|q,\ p \text{ prime}} \frac{1}{p^n} \\
\le &\sum_{k=2}^{q} \frac{1}{k^n} \\
\le &\sum_{t=1}^{\lfloor \log_2 q \rfloor} \sum_{k=2^t}^{2^{t+1}-1} \frac{1}{k^n} \\
\le &\sum_{t=1}^{\lfloor \log_2 q \rfloor} \frac{2^t}{2^{tn}} \\
= &\sum_{t=1}^{\lfloor \log_2 q \rfloor} \left(\frac{1}{2^{n-1}}\right)^t \\
<_{(*)} &\sum_{t=1}^{\infty} \left(\frac{1}{2^{n-1}}\right)^t \\
= &\frac{1}{2^{n-1}} \cdot \frac{1}{1 - 1/2^{n-1}} \\
= &\frac{1}{2^{n-1} - 1},
\end{aligned}
$$

where (*) uses $n \ge 2$, so that the summation converges. □

C.4 Proof of Lemma 16

Proof. We first prove Lemma 16 when $p \ge q$. We define $f(n)$ as:

$$f(n) = \max_{y_1, y_2 \in [q]} (\mathcal{Q}_n(y_1) - \mathcal{Q}_n(y_2)).$$

Then $f(1) \le 1$, and $\forall y_1, y_2 \in [q]$, $\mathcal{Q}_n(y_1) - \mathcal{Q}_n(y_2) \le f(n)$. We only need to show that $f(n) \le (1-c)^{n-1}$ for all $n \ge 1$. If this is true, then we have:

$$\sum_{x\in[q]} |\mathcal{Q}_n(x) - \frac{1}{q}| \le \sum_{x\in[q]} f(n) = q(1-c)^{n-1},$$

which means that the statistical distance between $\mathcal{Q}_n$ and $\mathcal{U}_q$ is within $q(1-c)^{n-1}$.

For all $n \geq 2$, for $\mathcal{Q}_n$, we have the following recursive formula: for all $t \in [q]$,

$$\mathcal{Q}_n(t) = \sum_{i=0}^{p-1} \mathcal{P}(i \bmod p)\mathcal{Q}_{n-1}((t-i) \bmod q)$$

Since $p \geq q$, there exists $\{\alpha_{j,i} \geq c\}_{i,j\in[q]}$ such that for all $t \in [q]$, $\sum_{i=0}^{q-1} \alpha_{t,i} = 1$, and

$$\mathcal{Q}_n(t) = \sum_{i=0}^{q-1} \alpha_{t,i}\mathcal{Q}_{n-1}(i).$$

For any $t_1, t_2 \in [q]$, we have

$$\begin{aligned}
\mathcal{Q}_n(t_1) - \mathcal{Q}_n(t_2) &= \sum_{i=0}^{q-1} (\alpha_{t_1,i} - \alpha_{t_2,i})\mathcal{Q}_{n-1}(i) \\
&= \sum_{i\in\mathcal{S}_+} \beta_{t_1,t_2,i}\mathcal{Q}_{n-1}(i) - \sum_{j\in\mathcal{S}_-} \gamma_{t_1,t_2,j}\mathcal{Q}_{n-1}(j) \\
&\leq \left(\sum_{i\in\mathcal{S}_+} \beta_{t_1,t_2,i}\right) \mathcal{Q}_{n-1}(i^*) - \left(\sum_{j\in\mathcal{S}_-} \gamma_{t_1,t_2,j}\right) \mathcal{Q}_{n-1}(j^*),
\end{aligned}$$

where $\mathcal{S}_+ \subseteq [q]$ is the set of i such that $\beta_{t_1,t_2,i} = \alpha_{t_1,i} - \alpha_{t_2,i} \geq 0$, and $\mathcal{S}_- \subseteq [q]$ is the set of j such that $-\gamma_{t_1,t_2,j} = \alpha_{t_1,j} - \alpha_{t_2,j} < 0$. i^* is the index such that $\mathcal{Q}_{n-1}(i^*) = \max_{i\in\mathcal{S}_+} \mathcal{Q}_{n-1}(i)$, j^* is the index such that $\mathcal{Q}_{n-1}(j^*) = \min_{j\in\mathcal{S}_-} \mathcal{Q}_{n-1}(j)$. Note that $\mathcal{S}_+ \cup \mathcal{S}_- = [q]$. We have

$$\begin{aligned}
&\sum_{i\in\mathcal{S}_+} \beta_{t_1,t_2,i} - \sum_{j\in\mathcal{S}_-} \gamma_{t_1,t_2,j} \\
=&\sum_{i\in\mathcal{S}_+} (\alpha_{t_1,i} - \alpha_{t_2,i}) + \sum_{j\in\mathcal{S}_-} (\alpha_{t_1,j} - \alpha_{t_2,j}) \\
=&\sum_{i\in\mathcal{S}_+\cup\mathcal{S}_-} \alpha_{t_1,i} - \sum_{i\in\mathcal{S}_+\cup\mathcal{S}_-} \alpha_{t_2,i} \\
=&1 - 1 = 0
\end{aligned}$$

Note that $\mathcal{S}_+ \neq \emptyset$. Otherwise, $\alpha_{t_1,i} < \alpha_{t_2,i}, \forall i \in [q]$, which contradicts the fact that $\sum_{i=0}^{q-1} \alpha_{t_1,i} = \sum_{i=0}^{q-1} \alpha_{t_2,i} = 1$. Suppose $i^* \in \mathcal{S}_+$, then we have

$$\sum_{i\in\mathcal{S}_+} \beta_{t_1,t_2,i} = \sum_{i\in\mathcal{S}_+} \alpha_{t_1,i} - \sum_{i\in\mathcal{S}_+} \alpha_{t_2,i} \leq \sum_{i=0}^{q-1} \alpha_{t_1,i} - \alpha_{t_2,i^*} \leq 1 - c.$$

Then for any $t_1, t_2 \in [q]$, we have:

$$\mathcal{Q}_n(t_1) - \mathcal{Q}_n(t_2) \leq \sum_{i\in\mathcal{S}_+} \beta_{t_1,t_2,i}(\mathcal{Q}_{n-1}(i^*) - \mathcal{Q}_{n-1}(j^*)) \leq \sum_{i\in\mathcal{S}_+} \beta_{t_1,t_2,i} f(n-1) \leq (1-c)f(n-1).$$

Therefore,

$$f(n) \leq (1-c)f(n-1),$$

which means

$$f(n) \le (1-c)^{n-1}, \forall n \ge 1.$$

Now we consider the general case where $p < q$. Let $\mathcal{P}_k$ denote the distribution of $\sum_{i=1}^{k} X_i$ where each $X_i \in \mathcal{P}$. Then $\mathcal{P}_k$ is a c^k-thresholded distribution with support $[kp]$. We are guaranteed that $kp \ge q$. Applying the result above for $\mathcal{P}_k$, we know that the statistical distance between $\mathcal{Q}_{nk}$ and $\mathcal{U}_q$ is within $q(1-c^k)^{n-1}$. Therefore, the statistical distance between $\mathcal{Q}_n$ and $\mathcal{U}_q$ is within $q(1-c^k)^{\lfloor n/k \rfloor - 1}$. □

C.5 Proof of Lemma 33

Proof. For any $\mathbf{z}$, we have

$$\begin{aligned}
(\mathbf{s}-\mathbf{z})^{\mathbf{v}} =& \prod_{j=1}^{n} (s(j)-z(j))^{v(j)} \\
=& \sum_{\mathbf{0}\le\mathbf{t}\le\mathbf{v}} \prod_{j=1}^{n} (-1)^{t(j)} \cdot \binom{v(j)}{t(j)} \cdot s(j)^{v(j)-t(j)} \cdot z(j)^{t(j)} \\
=& \mathbf{s}^{\mathbf{v}} + \sum_{\mathbf{0}<\mathbf{t}\le\mathbf{v}} (-1)^{\|\mathbf{t}\|_1} \binom{\mathbf{v}}{\mathbf{t}} \mathbf{s}^{\mathbf{v}-\mathbf{t}} \mathbf{z}^{\mathbf{t}}
\end{aligned}$$

where we adopt the shorthand $\binom{\mathbf{v}}{\mathbf{t}} := \prod_{j=1}^{n} \binom{v(j)}{t(j)}$. Since

$$\begin{aligned}
P_i(\mathbf{z}) =& \tilde{P}_i(\tilde{\mathbf{z}})\big|_{\tilde{\mathbf{z}}=\mathbf{s}-\mathbf{z}} \\
=& \sum_{\mathbf{v}\in\mathbb{N}^n, \|\mathbf{v}\|_1 \le d} \tilde{c}_{i,\mathbf{v}} (\mathbf{s}-\mathbf{z})^{\mathbf{v}} \\
=& \sum_{\mathbf{v}\in\mathbb{N}^n, \|\mathbf{v}\|_1 \le d} \tilde{c}_{i,\mathbf{v}} (\mathbf{s}^{\mathbf{v}} + \sum_{\mathbf{0}<\mathbf{t}\le\mathbf{v}} (-1)^{\|\mathbf{t}\|_1} \binom{\mathbf{v}}{\mathbf{t}} \mathbf{s}^{\mathbf{v}-\mathbf{t}} \mathbf{z}^{\mathbf{t}}),
\end{aligned}$$

we have its linearization is

$$P'_i(\mathbf{y}) = \sum_{\mathbf{v}\in\mathbb{N}^n, \|\mathbf{v}\|_1 \le d} \tilde{c}_{i,\mathbf{v}} (\mathbf{s}^{\mathbf{v}} + \sum_{\mathbf{0}<\mathbf{t}\le\mathbf{v}} (-1)^{\|\mathbf{t}\|_1} \binom{\mathbf{v}}{\mathbf{t}} \mathbf{s}^{\mathbf{v}-\mathbf{t}} y_{\mathbf{t}}).$$

For every $\mathbf{x} = (\mathbf{x}_{(1)}^T, \mathbf{x}_{(2)}^T, \ldots, \mathbf{x}_{(d)}^T)^T = (x_{\boldsymbol{\delta}_1}, x_{\boldsymbol{\delta}_2}, \ldots, x_{d\boldsymbol{\delta}_n})^T \in \mathbb{Z}_q^N$, define

$$f_{\mathbf{s}}(\mathbf{x}, \mathbf{v}) := (\mathbf{s}^{\mathbf{v}} + \sum_{\mathbf{0}<\mathbf{t}\le\mathbf{v}} (-1)^{\|\mathbf{t}\|_1} \binom{\mathbf{v}}{\mathbf{t}} \mathbf{s}^{\mathbf{v}-\mathbf{t}} \cdot x_{\mathbf{t}}) \bmod q,$$

and we have

$$P'_i(\mathbf{y}) \equiv \sum_{\mathbf{v}\in\mathbb{N}^n, \|\mathbf{v}\|_1 \le d} \tilde{c}_{i,\mathbf{v}} f_{\mathbf{s}}(\mathbf{y}, \mathbf{v}) \pmod q. \tag{38}$$

Recall the definition of $\tilde{P}'_i$:

$$\tilde{P}'_i(\tilde{\mathbf{y}}) = \sum_{\mathbf{v}\in\mathbb{N}^n, \|\mathbf{v}\|_1 \le d} \tilde{c}_{i,\mathbf{v}} \tilde{y}_{\mathbf{v}}. \tag{39}$$

By Eqn (38) and Eqn (39), we have

$$\tilde{P}'_i(\tilde{\mathbf{y}})\big|_{\{\forall \mathbf{v},\ \tilde{y}_{\mathbf{v}}=f_{\mathbf{s}}(\mathbf{y},\mathbf{v})\}} \equiv P'_i(\mathbf{y}) \pmod q.$$

Define

$$F_{\mathbf{s}}(\mathbf{x}) := (f_{\mathbf{s}}(\mathbf{x},\boldsymbol{\delta}_1), f_{\mathbf{s}}(\mathbf{x},\boldsymbol{\delta}_2), \ldots, f_{\mathbf{s}}(\mathbf{x},d\boldsymbol{\delta}_n))^T,$$

and then we have

$$\tilde{P}'_i(\tilde{\mathbf{y}})\big|_{\tilde{\mathbf{y}}=F_{\mathbf{s}}(\mathbf{y})} \equiv P'_i(\mathbf{y}) \pmod q.$$

This means for any $\mathbf{y}^* \in \mathcal{Y}$, if we let $\tilde{\mathbf{y}}^* := F_{\mathbf{s}}(\mathbf{y}^*)$, then we have $\tilde{P}'_i(\tilde{\mathbf{y}}^*) \equiv P'_i(\mathbf{y}^*) \equiv 0 \pmod q$, which indicates $\tilde{\mathbf{y}}^* \in \tilde{\mathcal{Y}}$. Hence, $F_{\mathbf{s}}$ is a mapping from $\mathcal{Y}$ to $\tilde{\mathcal{Y}}$.

Then we show $F_{\mathbf{s}}$ is a bijection. We first express $f_{\mathbf{s}}$ in the vector form. For every $\mathbf{v} \in \mathbb{N}^n$ such that $0 < \|\mathbf{v}\| \le d$, define a vector $\boldsymbol{\mu}_{\mathbf{s},\mathbf{v}} = (\mu_{\mathbf{s},\mathbf{v},\boldsymbol{\delta}_1}, \mu_{\mathbf{s},\mathbf{v},\boldsymbol{\delta}_2}, \ldots, \mu_{\mathbf{s},\mathbf{v},d\boldsymbol{\delta}_n})^T \in \mathbb{Z}_q^N$ such that

$$f_{\mathbf{s}}(\mathbf{x},\mathbf{v}) \equiv \mathbf{s}^{\mathbf{v}} + \langle \boldsymbol{\mu}_{\mathbf{s},\mathbf{v}}, \mathbf{x}\rangle \pmod q.$$

Notably, $\mu_{\mathbf{s},\mathbf{v},\mathbf{v}} = \pm 1$, and $\mu_{\mathbf{s},\mathbf{v},\mathbf{v}'} = 0$ for every $\mathbf{v}'$ such that $\mathbf{v}' \neq \mathbf{v}$ and $\|\mathbf{v}'\|_1 \ge \|\mathbf{v}\|_1$.

Let $\mathbf{M}_{\mathbf{s}} := (\boldsymbol{\mu}_{\mathbf{s},\boldsymbol{\delta}_1}, \boldsymbol{\mu}_{\mathbf{s},\boldsymbol{\delta}_2}, \ldots, \boldsymbol{\mu}_{\mathbf{s},d\boldsymbol{\delta}_n})^T$. Then $\mathbf{M}_{\mathbf{s}}$ is a lower-triangular matrix with each diagonal entry being ± 1. Consequently, $\mathbf{M}_{\mathbf{s}}$ is invertible. Moreover,

$$\begin{aligned}
&\mathbf{M}_{\mathbf{s}}\mathbf{x} + (\mathbf{s}^{\boldsymbol{\delta}_1}, \mathbf{s}^{\boldsymbol{\delta}_2}, \ldots, \mathbf{s}^{d\boldsymbol{\delta}_n})^T \\
\equiv &(\langle \boldsymbol{\mu}_{\mathbf{s},\boldsymbol{\delta}_1}, \mathbf{x}\rangle + \mathbf{s}^{\boldsymbol{\delta}_1}, \langle \boldsymbol{\mu}_{\mathbf{s},\boldsymbol{\delta}_2}, \mathbf{x}\rangle + \mathbf{s}^{\boldsymbol{\delta}_2}, \ldots, \langle \boldsymbol{\mu}_{\mathbf{s},d\boldsymbol{\delta}_n}, \mathbf{x}\rangle + \mathbf{s}^{d\boldsymbol{\delta}_n})^T \\
\equiv &(f_{\mathbf{s}}(\mathbf{x},\boldsymbol{\delta}_1), f_{\mathbf{s}}(\mathbf{x},\boldsymbol{\delta}_2), \ldots, f_{\mathbf{s}}(\mathbf{x},d\boldsymbol{\delta}_n))^T \\
\equiv &F_{\mathbf{s}}(\mathbf{x}) \pmod q,
\end{aligned}$$

showing that $F_{\mathbf{s}}$ is indeed a bijection.

Finally, we verify that if $F_{\mathbf{s}}(\mathbf{y}^*) = \tilde{\mathbf{y}}^*$, then $\tilde{\mathbf{y}}^*_{(1)} = \mathbf{s} - \mathbf{y}^*_{(1)}$. In fact,

$$\forall 1 \le j \le n,\ \tilde{y}^*_{\boldsymbol{\delta}_j} \equiv f_{\mathbf{s}}(\mathbf{y}^*, \boldsymbol{\delta}_j) \equiv s(j) - y^*_{\boldsymbol{\delta}_j} \pmod q.$$

This establishes $\tilde{\mathbf{y}}^*_{(1)} \equiv \mathbf{s} - \mathbf{y}^*_{(1)} \pmod q$, completing the proof of Lemma 33. □

C.6 Proof of Corollary 32

We only need to prove Lemma 64 in the following. The proof of Corollary 32 is a simple combination of Lemma 31 and Lemma 64.

Lemma 64. *Let $q = p^\kappa$, where p is prime and κ is some positive integer. Let $d \in [1, q)$ be an integer. Let $\chi_{d,\sigma}$ be a σ-thresholded distribution on $[d]$ for some $0 < \sigma \le 1/d$. Let n, m be positive integers. Let $\{(\mathbf{a}_i, b_i = (\langle \mathbf{a}_i, \mathbf{s}\rangle + e_i) \bmod q)\}_{1\le i\le m}$ be an instance of $LWE_{n,m,q,\chi_{d,\sigma}}$, and let*

$$\mathcal{S} \leftarrow \textsc{Arora_Ge}(n, m, d, q, \{(\mathbf{a}_i, b_i)\}_{1\le i\le m}).$$

Then for any $\mathbf{s}' \in \mathbb{Z}_q^n$ satisfying $\mathbf{s}' \equiv \mathbf{s} \pmod{q/\gcd(d!, q)}$, we have $\mathbf{s}' \in \mathcal{S}$.

Remark 65. *Let $\mathcal{S}' := \{\mathbf{s}' \in \mathbb{Z}_q^n \mid \mathbf{s}' \equiv \mathbf{s} \pmod{q/\gcd(d!, q)}\}$. Lemma 31 shows $\mathcal{S} \subseteq \mathcal{S}'$, Lemma 64 shows $\mathcal{S}' \subseteq \mathcal{S}'$, and Corollary 32 shows $\mathcal{S}' = \mathcal{S}$.*

Proof. Let $P_i'(\mathbf{y}), \tilde{P}_i'(\tilde{\mathbf{y}})$ be as defined in Eqns. (17), (20). Let $\mathcal{Y}, \tilde{\mathcal{Y}}$ be as defined in Eqns. (21) , (22), and $F_{\mathbf{s}}$ be the mapping from Lemma 33. Since $\mathbf{s} \in \mathcal{S}$, there exists some $\mathbf{y}^* = (\mathbf{y}_{(1)}^{*T}, \mathbf{y}_{(2)}^{*T}, \ldots, \mathbf{y}_{(d)}^{*T})^T \in \mathcal{Y}$ such that $\mathbf{y}_{(1)}^* = \mathbf{s}$. Let $\tilde{\mathbf{y}}^* := (\tilde{\mathbf{y}}_{(1)}^{*T}, \tilde{\mathbf{y}}_{(2)}^{*T}, \ldots, \tilde{\mathbf{y}}_{(d)}^{*T})^T := F_{\mathbf{s}}(\mathbf{y}^*) \in \tilde{\mathcal{Y}}$. Then we have

$$\tilde{\mathbf{y}}_{(1)}^* \equiv \mathbf{s} - \mathbf{y}_{(1)}^* \equiv \mathbf{0} \pmod{q}.$$

For any $\mathbf{v} \in \mathbb{N}^n$ s.t. $1 \leq \|\mathbf{v}\|_1 \leq d$, define

$$\Delta\tilde{y}_{\mathbf{v}} := \begin{cases} s(j) - s'(j), & \exists 1 \leq j \leq n, 1 \leq k \leq d \text{ s.t. } \mathbf{v} = k\boldsymbol{\delta}_j \\ 0, & \text{otherwise.} \end{cases} \tag{40}$$

For each $1 \leq k \leq d$, let $\Delta\tilde{\mathbf{y}}_{(k)}$ denote the vector formed by sorting $\{\Delta\tilde{y}_{\mathbf{v}}\}_{\|\mathbf{v}\|_1=k}$ in reverse lexicographical order of $\mathbf{v}$. Then we have $\Delta\tilde{\mathbf{y}}_{(1)} = \mathbf{s} - \mathbf{s}'$. Let $\Delta\tilde{\mathbf{y}} := (\Delta\tilde{\mathbf{y}}_{(1)}^T, \Delta\tilde{\mathbf{y}}_{(2)}^T, \ldots, \Delta\tilde{\mathbf{y}}_{(d)}^T)^T$.

By Lemma 66 (stated below), we have $\tilde{P}_i'(\Delta\tilde{\mathbf{y}}) \equiv 0 \pmod{q}$. Since $\tilde{\mathbf{y}}^* \in \tilde{\mathcal{Y}}$, it follows that $\tilde{P}_i'(\tilde{\mathbf{y}}^*) \equiv 0 \pmod{q}$. Define $\tilde{\mathbf{y}}'^* := (\tilde{\mathbf{y}}_{(1)}'^{*T}, \tilde{\mathbf{y}}_{(2)}'^{*T}, \ldots, \tilde{\mathbf{y}}_{(d)}'^{*T})^T := \tilde{\mathbf{y}}^* + \Delta\tilde{\mathbf{y}}$. Then $\tilde{\mathbf{y}}_{(1)}'^* \equiv \mathbf{s} - \mathbf{s}' \pmod{q}$, and

$$\tilde{P}_i'(\tilde{\mathbf{y}}'^*) \equiv \tilde{P}_i'(\tilde{\mathbf{y}}^*) + \tilde{P}_i'(\Delta\tilde{\mathbf{y}}) \equiv 0 \pmod{q}.$$

Hence $\tilde{\mathbf{y}}'^* \in \tilde{\mathcal{Y}}$. By Lemma 33, there exists $\mathbf{y}'^* = (\mathbf{y}_{(1)}^{*T}, \mathbf{y}_{(2)}^{*T}, \ldots, \mathbf{y}_{(d)}^{*T})^T \in \mathcal{Y}$ such that

$$\mathbf{y}_{(1)}'^* \equiv \mathbf{s} - \tilde{\mathbf{y}}_{(1)}'^* \equiv \mathbf{s}' \pmod{q}.$$

So we have $\mathbf{s}' \in \mathcal{S}$. □

Lemma 66. *Adopting the notation in the statement and proof of Lemma 64, we have*

$$\forall 1 \leq i \leq m,\ \tilde{P}_i'(\Delta\tilde{\mathbf{y}}) \equiv 0 \pmod{q},$$

where $\tilde{P}_i'(\tilde{\mathbf{y}})$ is defined in Eqn. (20).

Proof. For any integers t, k, l such that $t \geq 1$, $0 \leq k < t$, $0 \leq l \leq t-1$, let $w_{t,k,l}$ be as defined in Eqn. (23). We have

$$\begin{aligned}
\tilde{P}_i'(\Delta\tilde{\mathbf{y}}) &\equiv \sum_{k=1}^{d} w_{d,d-k,e_i} \sum_{\substack{\mathbf{v}\in\mathbb{N}^n \\ \|\mathbf{v}\|_1=k}} \binom{k}{\mathbf{v}} \mathbf{a}_i^{\mathbf{v}} \Delta\tilde{y}_{\mathbf{v}}^*. \\
&\equiv \sum_{k=1}^{d} w_{d,d-k,e_i} \sum_{j=1}^{n} \binom{k}{k\boldsymbol{\delta}_j} \mathbf{a}_i^{k\boldsymbol{\delta}_j} \Delta\tilde{y}_{k\boldsymbol{\delta}_j}^* \\
&\equiv \sum_{k=1}^{d} w_{d,d-k,e_i} \sum_{j=1}^{n} a_i(j)^k (s'(j) - s(j)) \\
&\equiv \sum_{j=1}^{n} (s'(j) - s(j)) \sum_{k=1}^{d} w_{d,d-k,e_i} a(j)^k \\
&\equiv \sum_{j=1}^{n} (s'(j) - s(j)) \prod_{\eta\in[d]} (a_i(j) + (e_i - \eta)) \pmod{q}
\end{aligned}$$

Since

$$\mathbf{s}' \equiv \mathbf{s} \pmod{q/gcd(d!, q)},$$

it follows that

$$\forall 1 \leq j \leq n,\ s'(j) - s(j) \equiv 0 \pmod{q/gcd(d!, q)}.$$

Meanwhile, by Proposition 67 (stated below), we have

$$\forall 1 \leq j \leq n,\ \prod_{\eta \in [d]} (a_i(j) + (e_i - \eta)) \equiv 0 \pmod{d!}.$$

As a result,

$$1 \leq j \leq n,\ (s'(j) - s(j)) \prod_{\eta \in [d]} (a_i(j) + (e_i - \eta)) \equiv 0 \pmod{q},$$

which implies

$$\tilde{P}_i'(\Delta \tilde{\mathbf{y}}^*) \equiv 0 \pmod{q}.$$

This completes the proof. □

Proposition 67. *For any $x \in \mathbb{Z}$, $d \in \mathbb{N}^+$,*

$$\prod_{\eta \in [d]} (x - \eta) \equiv 0 \pmod{d!}$$

Proof. Without loss of generality, assume $0 \leq x < d!$. Define $f_d(x) := \prod_{\eta \in [d]}(x - \eta)$. It suffices to prove $f_d(x) \equiv 0 \pmod{d!}$.

- When $0 \leq x < d$, there exists some $\eta \in [d]$ such that $x - \eta = 0$, so $f_d(x) = 0$.
- When $d \leq x < d!$, we have

  $$\frac{f_d(x)}{d!} = \frac{x!}{(x-d)!d!} = \binom{x}{d}$$

  is an integer, so it holds that $f_d(x) \equiv 0 \pmod{d!}$.

□

C.7 Proof of Lemma 40

Proof. Since $d > q'$, we have $d - d/q' > q' - 1$,, and thus $d - \lfloor d/q' \rfloor \geq q'$. Then, there must exist an integer $x \in (\lfloor d/q' \rfloor, d]$ such that $q' \mid x$. Consequently, we have:

$$\lfloor d/q' \rfloor! \mid (d!/x) \mid (d!/q').$$

This implies

$$gcd(\lfloor d/q' \rfloor!, q/q') \leq gcd(d!/q', q/q') = gcd(d!, q)/q'.$$

Since $d!$ mod $q \neq 0$, we know that $gcd(d!, q) < q$. Thus

$$gcd(\lfloor d/q' \rfloor!, q/q') \leq gcd(d!/q', q/q') < q/q',$$

which means $\lfloor d/q' \rfloor!$ mod $(q/q') \neq 0$. □

D The output of Arora-Ge Algorithm when $q = p^\kappa$

Table 3 shows what we learn by running the Arora-Ge algorithm *once* on $\mathrm{LWE}_{n,m,q,\chi_{d,\sigma}}$. For certain q and d, we indicate in the table that the output is $\mathbf{s} \bmod u$, if the Arora-Ge algorithm outputs the set $\{\mathbf{s}' \in \mathbb{Z}_q^n \mid \mathbf{s}' \equiv \mathbf{s} \pmod{u}\}$, where $\mathbf{s}$ is the secret of the input LWE instance and u is a positive integer. (Actually, we have $u = q/gcd(d!, q)$ by Corollary 32.) Notably, when the output is $\mathbf{s} \bmod 1$, it means the algorithm outputs the set $\mathbb{Z}_q^n$, i.e., we learn nothing about $\mathbf{s}$.

q	d	output	q	d	output	q	d	output	q	d	output
2	1	$\mathbf{s} \bmod 2$	16	6	$\mathbf{s} \bmod 1$	64	7	$\mathbf{s} \bmod 4$	27	5	$\mathbf{s} \bmod 9$
2	2	$\mathbf{s} \bmod 1$	32	1	$\mathbf{s} \bmod 32$	64	8	$\mathbf{s} \bmod 1$	27	6	$\mathbf{s} \bmod 3$
4	1	$\mathbf{s} \bmod 4$	32	2	$\mathbf{s} \bmod 16$	3	1	$\mathbf{s} \bmod 3$	27	7	$\mathbf{s} \bmod 3$
4	2	$\mathbf{s} \bmod 2$	32	3	$\mathbf{s} \bmod 16$	3	2	$\mathbf{s} \bmod 3$	27	8	$\mathbf{s} \bmod 3$
4	3	$\mathbf{s} \bmod 2$	32	4	$\mathbf{s} \bmod 4$	3	3	$\mathbf{s} \bmod 1$	27	9	$\mathbf{s} \bmod 1$
4	4	$\mathbf{s} \bmod 1$	32	5	$\mathbf{s} \bmod 4$	9	1	$\mathbf{s} \bmod 9$	81	1	$\mathbf{s} \bmod 81$
8	1	$\mathbf{s} \bmod 8$	32	6	$\mathbf{s} \bmod 2$	9	2	$\mathbf{s} \bmod 9$	81	2	$\mathbf{s} \bmod 81$
8	2	$\mathbf{s} \bmod 4$	32	7	$\mathbf{s} \bmod 2$	9	3	$\mathbf{s} \bmod 3$	81	3	$\mathbf{s} \bmod 27$
8	3	$\mathbf{s} \bmod 4$	32	8	$\mathbf{s} \bmod 1$	9	4	$\mathbf{s} \bmod 3$	81	4	$\mathbf{s} \bmod 27$
8	4	$\mathbf{s} \bmod 1$	64	1	$\mathbf{s} \bmod 64$	9	5	$\mathbf{s} \bmod 3$	81	5	$\mathbf{s} \bmod 27$
16	1	$\mathbf{s} \bmod 16$	64	2	$\mathbf{s} \bmod 32$	9	6	$\mathbf{s} \bmod 1$	81	6	$\mathbf{s} \bmod 9$
16	2	$\mathbf{s} \bmod 8$	64	3	$\mathbf{s} \bmod 32$	27	1	$\mathbf{s} \bmod 27$	81	7	$\mathbf{s} \bmod 9$
16	3	$\mathbf{s} \bmod 8$	64	4	$\mathbf{s} \bmod 8$	27	2	$\mathbf{s} \bmod 27$	81	8	$\mathbf{s} \bmod 9$
16	4	$\mathbf{s} \bmod 2$	64	5	$\mathbf{s} \bmod 8$	27	3	$\mathbf{s} \bmod 9$	81	9	$\mathbf{s} \bmod 1$
16	5	$\mathbf{s} \bmod 2$	64	6	$\mathbf{s} \bmod 4$	27	4	$\mathbf{s} \bmod 9$			

Table 3: Calling Arora-Ge Once for LWE