MULTIVARIATE EXPONENTIAL EQUATIONS WITH UNKNOWN COEFFICIENTS

Trey Li

University of Manchester, Manchester, UK
trey.li@manchester.ac.uk

Abstract. We introduce a novel class of equations defined over Euclidean domains. These abstract equations establish a unified framework for deriving new, concrete computational problems useful for cryptography. We prove that solving a single such equation is NP-hard. For systems of these equations, we further prove NP-hardness, average-case hardness, random self-reducibility, search-to-decision reducibility, and trapdoorizability. Based on the hardness of solving these systems, we construct various cryptographic primitives. Our results are proved in an abstract, domain-agnostic manner and hold for a wide range of Euclidean domains. This generality allows the framework to accommodate rich mathematical structures, providing both theoretical depth and flexibility for diverse cryptographic applications.

1. Introduction

Many computational problems in cryptography involve solving polynomial equations, particularly in post-quantum cryptography.

In code-based cryptography, the syndrome decoding (SD) problem [7] asks to solve a linear system for a solution with bounded Hamming weight, while the learning parity with noise (LPN) problem [31, 9, 39] asks to solve a linear system with noise. Both are linear systems.

In lattice-based cryptography, the short integer solution (SIS) problem [2] asks to solve a linear system for a solution with bounded norm, while the learning with errors (LWE) problem [40] asks to solve a linear system with errors. They are also linear systems. Their ring and module variants [35, 38, 34, 46, 32], as well as the NTRU problem [26], extend these formulations to systems over rings and modules, which may involve non-linearity yet remain polynomial in nature.

In multivariate cryptography, the central problem is the multivariate quadratic (MQ) problem [17, 18], which is still restricted to polynomial equations.

In this work, we explore a new class of equations that exhibit nice properties not observed in previously studied problems.

1.1. Equations.

Let $\mathcal{R}$ be a Euclidean domain and $\mathfrak{p}$ be a prime ideal of $\mathcal{R}$. Let L be a set of primes of $\mathcal{R}$ and $\mathbb{z}$ be a subset of $\mathbb{Z}$. We consider equations of the form

$$\xi \cdot \prod_{i=1}^{n} a_i^{x_i} = b \pmod{\mathfrak{p}},$$

2020 *Mathematics Subject Classification.* Primary: 11R99, 94A60; Secondary: 68Q17, 11Y40.

Key words and phrases. Multivariate exponential equations, Euclidean domains, NP-hardness, average-case hardness, random self-reducibility, search-to-decision reducibility, post-quantum cryptography.

The author thanks Steven Galbraith and Libin Wang for valuable discussions.

where a_i and b are given elements of $\mathcal{R}$. The goal is to find a vector $(\xi, x_1, \ldots, x_n)$ in $L \times \mathbb{Z}^n$ such that the two sides are equivalent modulo $\mathfrak{p}$. We call such an equation a *multivariate exponential equation*.

Two questions are natural:

(1) How many solutions does the equation have?
(2) How can we solve for one or all solutions?

The equation can have zero, finite or infinite solutions. We introduce a lattice technique to identify equations with a unique solution up to different coefficients. The uniqueness of solutions is important for applications such as collision-free hash functions and digital signatures. This lattice technique is later used to prove the NP-hardness of solving this kind of equations in many Euclidean domains.

1.2. **Systems.** After handling single equations, we study systems of equations. They are systems of the form:

$$\begin{cases} f_1(\xi_1, x_1, \ldots, x_n) & = b_1 \pmod{\mathfrak{p}} \\ f_2(\xi_2, x_1, \ldots, x_n) & = b_2 \pmod{\mathfrak{p}} \\ & \cdots \\ f_k(\xi_k, x_1, \ldots, x_n) & = b_k \pmod{\mathfrak{p}} \end{cases}$$

where each equation with $f_i(\xi_i, x_1, \ldots, x_n) = \xi_i \cdot \prod_{j=1}^{n} a_{i,j}^{x_j}$ is a multivariate exponential equation. We call such a system a *multivariate exponential system*.

Solving this kind of systems is NP-hard in many Euclidean domains because solving a single equation is. We further prove various cryptographic properties for the problem of solving such systems, including average-case hardness, random self-reducibility, search-to-decision reducibility, and trapdoorizability.

1.3. **Applications.** Eventually, we show how the problem of solving multivariate exponential systems can be used to construct various cryptographic primitives, including:

- homomorphic encryption
- hash function
- trapdoor function
- digital signature
- multilinear map with coefficients
- multiparty key exchange
- broadcast encryption
- obfuscation
- reusable fuzzy extractor

The constructions utilize specific properties of Euclidean domain, such as unique factorization, division with remainder, and other properties related to prime elements. These properties are not provided by existing polynomial-system-solving problems.

Our cryptographic constructions are proved generally, meaning that we can plug in any concrete Euclidean domain of interest, and the correctness and security of the resulting constructions automatically hold.

1.4. **Contributions.** To the best of our knowledge, we are the first to consider multivariate exponential equations over Euclidean domains in the area of complexity and cryptography. Our contributions are threefold:

(1) **Complexity.** We proved *general* NP-hardness for the equation-solving problem. We proved *general* average-case hardness, *general* random self-reducibility, and *general* search-to-decision reducibility for the system-solving problem.
(2) **Cryptography.** We presented *general* constructions for several important cryptographic primitives.
(3) **Foundation.** Our general equations provide a framework to derive concrete computational problems (simply by specifying concrete Euclidean domains) that are guaranteed to be hard and applicable.

We explain the novelty of our methodology below.

Traditionally, computational problems are studied concretely and not known to be hard in general. For example, the discrete logarithm problem is hard in a multiplicative group but easy in an additive group. The LWE problem is believed quantum hard in rational integers and certain polynomial rings (e.g., the Module-LWE problem [32]), yet little research explores its hardness in more general algebraic structures. In contrast, our system-solving problem adopts a more general methodology, aiming to establish universal hardness across all Euclidean domains of interest, including, but not limited to, integers and polynomials.

Historically, most computational problems were not known to have wide applications when they were invented. For example, it took years to find applications for different hidden subgroup problems [29]; and we still lack many applications for the multivariate quadratic problem [17, 18]. In contrast, our equations provide a framework to generate computational problems that are initially known to have many applications.

In cryptography, an insightful interest is to find a powerful primitive that implies many other primitives. Typical examples include homomorphic encryption [41], multilinear map [10], and program obfuscation [5], etc. However, a powerful primitive itself needs to build upon a hard computational problem. Our framework traces back to the root and directly addresses computational problems, seeking to identify a powerful computational problem that implies many cryptographic primitives.

1.5. **Organization.** Section 2 gives preliminaries of ring theory and defines notations that will be used in the rest of the paper. Section 3 defines the equation-solving problem, studies its number of solutions, and proves its NP-hardness. Section 4 defines the system-solving problem, studies its number of solutions, proves its NP-hardness, average-case hardness, random self-reducibility, search-to-decision reducibility, and trapdoorizability. Section 5 presents various cryptographic applications, some of which are not known to be achievable under existing hardness assumptions in such simple ways. Section 6 presents final remarks.

2. Preliminaries and notations

2.1. **Ring theory.** An *integral domain* is a commutative ring with identity that has no zero divisors. In an integral domain, an element is said to be *irreducible* if it is neither a unit nor a product of non-units. In a commutative ring, an element is said to be *prime* if it is a non-unit and whenever it divides the product of two elements it divides at least one of them. In an integral domain, every prime is irreducible, but the converse is not true in general.

A *unique factorization domain* is an integral domain in which every nonzero non-unit has a unique factorization into irreducible elements up to reordering of the

factors. In a unique factorization domain, irreducible elements are precisely prime elements.

A *principal ideal domain* is an integral domain in which every ideal is principal. In a commutative ring, every prime element generates a prime ideal and every principal prime ideal is generated by a prime element. Hence in a principal ideal domain, a nonzero[1] ideal is prime if and only if it is generated by a prime element.

A *Euclidean domain* is an integral domain $\mathcal{R}$ where there exists a function $d : \mathcal{R}\backslash\{0\} \to \mathbb{N}$ such that for any $a, b \in \mathcal{R}$, $b \neq 0$ there exist $q, r \in \mathcal{R}$ such that $a = qb+r$ and either $d(r) = 0$ or $d(r) < d(b)$. The function d is called a *Euclidean function* or *degree function*, and $d(a)$ is called the *degree* of a. Every Euclidean domain is a principal ideal domain. Every principal ideal domain is a unique factorization domain. A unique factorization domain is a principal ideal domain if and only if every nonzero prime ideal is maximal. Hence every nonzero prime ideal of a Euclidean domain is maximal.

A quotient ring $\mathcal{R}/\mathcal{I}$ is a field if and only if $\mathcal{I}$ is a maximal ideal. Hence for a Euclidean domain $\mathcal{R}$, if $\mathcal{I}$ is a nonzero prime ideal, then the quotient ring $\mathcal{R}/\mathcal{I}$ is a field. We call a quotient ring $\mathcal{R}/\mathcal{I}$ that is a field a *residue field*.

A ring $\mathcal{R}$ is called a *ring with finite residue fields*[2] if for all maximal ideals $\mathcal{I}$ of $\mathcal{R}$, the residue field $\mathcal{R}/\mathcal{I}$ is finite. The multiplicative group of a finite field is cyclic. Its order is smaller than the order of the field by one.

From the above we have the following. Every prime element q of a Euclidean domain $\mathcal{R}$ generates a (nonzero) prime ideal (q). The ideal (q) is maximal and thus $\mathcal{R}/(q)$ is a field. If $\mathcal{R}$ is a ring with finite residue fields, then $\mathcal{R}/(q)$ is a finite field. In that case, the multiplicative group $(\mathcal{R}/(q))^\times$ of $\mathcal{R}/(q)$ is cyclic, and its order is $|\mathcal{R}/(q)| - 1$ where $|\mathcal{R}/(q)|$ is the order of $\mathcal{R}/(q)$.

We denote multiplications in $\mathcal{R}/\mathcal{I}$ by $a \cdot b \pmod{\mathcal{I}}$ for $a, b \in \mathcal{R}$. When $\mathcal{I} = (q)$ we simply denote it as $a \cdot b \pmod{q}$. We also denote $\mathcal{R}/(q)$ as $\mathcal{R}_q$, and $(\mathcal{R}/(q))^\times$ as $\mathcal{R}_q^\times$.

2.2. **Common notations.** We denote matrices by bold uppercase letters such as $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, etc.; and denote vectors by bold lowercase letters such as $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$, etc.

Operation notations. Let $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n)$ be two vectors. Denote $\cdot$ as inner product between two vectors or normal multiplication between two elements, i.e., $\boldsymbol{a} \cdot \boldsymbol{b} := \sum_{i=1}^n a_i \cdot b_i$. Let $\boldsymbol{A} = (a_{i,j})$ and $\boldsymbol{B} = (b_{i,j})$ be two matrices. Denote $\circ$ as Hadamard product (or entry-wise product), i.e., $\boldsymbol{A} \circ \boldsymbol{B} := (a_{i,j} \cdot b_{i,j})$. Let x be an element. Denote $x\boldsymbol{A} := (xa_{i,j})$, $\boldsymbol{A}^{\circ x} := (a_{i,j}^x)$ and $x^{\boldsymbol{A}} := (x^{a_{i,j}})$. Namely applying x to $\boldsymbol{A}$ entry-wise. Denote $\boldsymbol{A}|x$ to be $\boldsymbol{A}$ with each entry replaced by either 1 or 0 depending on whether $a_{i,j}$ divides x.

Asymptotic notations. A function $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is *negligible* if for every positive polynomial $p(n)$, there exists a constant n_0 such that $f(n) < 1/p(n)$ for all $n > n_0$. A function $f : \mathbb{N} \to \mathbb{R}$ is *noticeable* if there exist a positive polynomial $p(n)$ and a constant n_0 such that $f(n) > 1/p(n)$ for all $n > n_0$. A function $f : \mathbb{N} \to \mathbb{R}$ is *overwhelming* if $1 - f$ is negligible. We denote generic negligible functions, polynomial functions and super-polynomial functions by $negl(\cdot)$, $poly(\cdot)$ and $spoly(\cdot)$, respectively; different occurrences of *negl*, *poly* and *spoly* can be different concrete functions.

[1] In an integral domain, the ideal (0) is a prime ideal but 0 is not defined as a prime.

[2] Refer to [28] for rings with finite quotient rings.

Reduction notations. Let P_1 and P_2 be two computational problems of binary sizes polynomial in n. Denote $P_1 \leq P_2$ (called P_1 reduces to P_2) if there exists a probabilistic polynomial time algorithm that transforms any solution to P_2 to a solution to P_1 with probability noticeable in n. Denote $P_1 = P_2$ (called P_1 computationally equivalent to P_2) if $P_1 \leq P_2$ and $P_2 \leq P_1$.
Other notations. Denote $s \leftarrow S$ as to sample an element s uniformly from S if S is a set, or to sample s according to S if S is a distribution.

3. Multivariate exponential equations with unknown coefficients

3.1. **Qualified Euclidean domain.** Note that not all Euclidean domains are suitable for our study. For example, a Euclidean domain where multiplication cannot be computed in polynomial time is not of computational interest. Specifically, we will focus on the following Euclidean domains.

Definition 3.1. A Euclidean domain $\mathcal{R}$ is said to be *qualified* if:

(1) it is a ring with finite residue fields;
(2) there exist a degree function d such that for all $a, b \in \mathcal{R}$, if $d(a) < d(b)$ then $a \bmod b = a$;
(3) there exist polynomial time algorithms to compute the degree function d, the multiplications and inverses in $\mathcal{R}$, to test divisibilities ($a \mid b$ or $a \nmid b$) for any elements $a, b \in \mathcal{R}$, and to sample random primes from $\mathcal{R}$;
(4) for any prime $q \in \mathcal{R}$, there exist polynomial time algorithms to compute multiplications and inverses in $\mathcal{R}_q^\times$, to sample random elements from $\mathcal{R}_q^\times$, and to compute the size $|\mathcal{R}_q|$ of $\mathcal{R}_q$.

Conditions (3) and (4) can be easily verified by looking at existing algorithms. We mainly discuss criteria for Conditions (1) and (2).

For Condition (1), we refer to [28] for a criterion to identify a wide range of rings with finite quotient rings. Combining the fact that we only consider quotient rings $\mathcal{R}_q$ where $\mathcal{R}$ is a Euclidean domain and (q) is a prime (hence maximal) ideal, we can identify a wide range of rings with finite residue fields.

For Condition (2), the following lemma gives a (sufficient but not necessary) criterion for a wide range of qualified degree functions.

Lemma 3.2. *Let $\mathcal{R}$ be a Euclidean domain with degree function d. If $\max\{d(a), d(b)\} \leq d(a+b)$ for all $a, b \in \mathcal{R}$, then $d(a) < d(b)$ if and only if $a \bmod b = a$.*

Proof. It is standard that in a Euclidean domain $\mathcal{R}$: (1) $d(a) \leq d(ab)$ for all $a, b \in \mathcal{R}\backslash\{0\}$; and (2) the quotient q and remainder r in $a = qb + r$ are unique for all $a, b \in \mathcal{R}$ if and only if $\max\{d(a), d(b)\} \leq d(a+b)$ for all $a, b \in \mathcal{R}$. Hence if $\max\{d(a), d(b)\} \leq d(a+b)$ for all $a, b \in \mathcal{R}$, then for any elements $a, b \in \mathcal{R}$ with $d(a) < d(b)$, we have $d(a) = d(qb + r) < d(b)$. Now suppose $q \neq 0$, then $d(b) \leq d(qb) \leq d(qb + r) = d(a)$, a contradiction. Hence we must have $q = 0$ and thus $r = a$ and thus $a \bmod b = a$. □

In fact, many common Euclidean domains are qualified Euclidean domains. For example, the ring $\mathbb{Z}$ of integers where the degree of an integer is the integer itself; the polynomial ring $\mathbb{F}_2[x]$ where the degree of a polynomial is the polynomial degree; the ring $\mathbb{Z}[i]$ of Gaussian integers where the degree of a Gaussian integer is its absolute square: $N(a + bi) = a^2 + b^2$; etc. Remark that the polynomial case $\mathbb{F}_2[x]$ does not satisfy the criterion given by Lemma 3.2, i.e., it does not satisfy $\max\{d(a),$

$d(b)\} \leq d(a+b)$ since the highest terms of two polynomials can be canceled out when the two polynomials are added together.

3.2. **Multivariate exponential equations with unknown coefficients.** Now we define our equations.

Definition 3.3. Let $\mathcal{R}$ be a qualified Euclidean domain and q be a prime of $\mathcal{R}$. Let L be a set of primes of $\mathcal{R}$ and $\mathbb{z}$ be a subset of $\mathbb{Z}_{\geq 0}$. An unknown-coefficient multivariate exponential equation over $\mathcal{R}$ is an equation of the form

$$\xi \cdot \boldsymbol{a}^{\boldsymbol{x}} = b \pmod{q},$$

where $\boldsymbol{a} \in \mathcal{R}^n$ and $b \in \mathcal{R}^\times$ are given constants, and $\xi \in L$ and $\boldsymbol{x} \in \mathbb{z}^n$ are unknowns.

Note that we force $b \neq 0$ to avoid the trivial case where no solutions exist.

3.3. **Existence and uniqueness of solutions.** If the solution domain is not restricted, namely L is the set of all primes of $\mathcal{R}$ and $\mathbb{z}$ is the full set $\mathbb{Z}_{\geq 0}$, then there are infinitely many solutions $(\xi, \boldsymbol{x})$. In fact, for any $i \in \{1, \ldots, n\}$, for any assignment to $(\xi, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, we can solve for x_i. Since there are infinitely many such assignments (including equivalences), there are infinitely many solutions $(\xi, x_1, \ldots, x_n)$ to the equation.

If both L and $\mathbb{z}$ are finite, then there are finitely many assignments to the sequence $(\xi, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ hence finitely many solutions $(\xi, x_1, \ldots, x_n)$ to the equation.

Now we investigate conditions for the solution to be unique. We show that if $d(q)$ is sufficiently large then the equation either has no solutions or a unique solution. Of course, in the case where q is very large such that $d(\xi \cdot \boldsymbol{a}^{\boldsymbol{x}}) < d(q)$ for all $(\xi, \boldsymbol{x}) \in L \times \mathbb{z}^n$, the conclusion is trivially true as $\mathcal{R}$ is a unique factorization domain. For example, the solution $(\xi, x_1, x_2, x_3) = (7, 1, 1, 0)$ to the equation $\xi \cdot 2^{x_1} \cdot 3^{x_2} \cdot 5^{x_3} = 42 \pmod{997}$ is clearly unique if $L = \{7, 11\}$ and $\mathbb{z} = \{0, 1\}$, as the modulus 997 is unreasonably large such that "(mod 997)" can always be ignored.

Therefore, the interesting case is when $d(q) < d(\xi \cdot \boldsymbol{a}^{\boldsymbol{x}})$ for at least some $(\xi, \boldsymbol{x}) \in L \times \mathbb{z}^n$, especially when $\mathbb{z} \subsetneq \{0, \ldots, |\mathcal{R}_q| - 2\}$ (note that $\mathcal{R}_q^\times$ is a multiplicative group of order $|\mathcal{R}_q| - 1$). We will focus on this case in the following. We call our technique for proving uniqueness of solutions the *product embedding* technique.

To illustrate the idea, we first introduce a special lattice, then define Gaussian-predictability for using the Gaussian heuristic [27, 20] in this lattice, and then show how Gaussian-predictability guarantees a product embedding, and eventually show how product embedding is associated with uniqueness of solutions. In the next section this technique is further used to prove the NP-hardness of the equation-solving problem.

We now introduce the special lattice. Let $\mathcal{R}$ be a ring, and $(\mathcal{R}/\mathcal{I})^\times$ be the multiplicative group of the quotient ring $\mathcal{R}/\mathcal{I}$. Consider the group homomorphism

$$\phi : \mathbb{Z}^m \to (\mathcal{R}/\mathcal{I})^\times; (z_1, \ldots, z_m) \mapsto \prod_{j=1}^{m} p_j^{z_j} \pmod{\mathcal{I}}$$

and the set

$$\Lambda = \left\{ (z_1, \ldots, z_m) \in \mathbb{Z}^m \,\middle|\, \prod_{j=1}^{m} p_j^{z_j} \equiv 1_{(\mathcal{R}/\mathcal{I})^\times} \pmod{\mathcal{I}} \right\} \tag{1}$$

of vectors defined by the kernel $\phi^{-1}(1_{(\mathcal{R}/\mathcal{I})^\times})$ of ϕ.

It is clear that Λ is an additive subgroup of $\mathbb{R}^m$ and thus a lattice.

Now we discuss the first minimum $\lambda_1(\Lambda)$ (i.e., Euclidean norm of the shortest vector) of the lattice Λ. We first review the Gaussian estimation [27, 20], which is a heuristic estimation of the first minimum of a lattice Λ. It is defined as

$$gh(\Lambda) = \sqrt{\frac{m}{2\pi e}}\mathrm{vol}(\Lambda)^{\frac{1}{m}},$$

where $\mathrm{vol}(\Lambda)$ is the volume of Λ.

If we fix a real number $\epsilon > 0$, then for all sufficiently large $m \in \mathbb{N}$, we have the following heuristic:

$$(1-\epsilon)\cdot gh(\Lambda) < \lambda_1(\Lambda) < (1+\epsilon)\cdot gh(\Lambda) \tag{2}$$

for a randomly chosen lattice Λ of dimension m [27, p.378], [45], where the parameter $1-\epsilon$ is usually a number close to 1.

We define Gaussian-predictability to describe a situation where the Gaussian estimation is easy to calculate and that it is guaranteed to be a lower bound of the first minimum.

Definition 3.4 (Gaussian-predictability)**.** Let $m, \ell \in \mathbb{N}$ with ℓ polynomial in m. Let $c \in \mathbb{R}_{>1}$. Let $\mathcal{R}$ be a ring and $\mathcal{I}$ be an ideal of $\mathcal{R}$. Let $(\mathcal{R}/\mathcal{I})^\times$ be the multiplicative group of $\mathcal{R}/\mathcal{I}$. Let $P = \{p_1, \ldots, p_m\}$ be a set of irreducible elements of $\mathcal{R}$. Let $\lambda_1(\Lambda)$ be the first minimum of the lattice Λ given by Formula (1) and $gh(\Lambda)$ be the Gaussian estimation of $\lambda_1(\Lambda)$. The pair $(P, (\mathcal{R}/\mathcal{I})^\times)$ is said to be (m, ℓ, c)-Gaussian-predictable if

(1) $(\mathcal{R}/\mathcal{I})^\times = \langle p_1, \ldots, p_m \rangle$;
(2) $|(\mathcal{R}/\mathcal{I})^\times| > (c\ell\sqrt{2\pi e})^m$;
(3) $gh(\Lambda) < c \cdot \lambda_1(\Lambda)$.

We call a Euclidean domain $\mathcal{R}$ an (m, ℓ, c)-Gaussian-predictable Euclidean domain if there exists an (m, ℓ, c)-Gaussian-predictable pair $(P, (\mathcal{R}/\mathcal{I})^\times)$.

Conditions (1) and (2) ensure the easiness of calculating the Gaussian estimation, and Condition (3) guarantees the Gaussian estimation is a lower bound.

Note that in Definition 3.4, the Gaussian estimation is used purely as a convenient formula for defining a lower bound; the usual heuristic assumptions behind it are irrelevant. In particular, satisfying Condition (3) does not require the Gaussian heuristic to hold. Gaussian-predictability may be achieved either probabilistically, by assuming heuristic behavior, or deterministically, through explicit structural guarantees. In this work, we emphasize the latter for theoretical proofs, while noting that the former remains useful in practice.

Now we show how Gaussian-predictability helps to embed a set of products from $\mathcal{R}$ to $(\mathcal{R}/\mathcal{I})^\times$.

Lemma 3.5. *Let $(P, (\mathcal{R}/\mathcal{I})^\times)$ be an (m, ℓ, c)-Gaussian-predictable pair, where $P := \{p_1, \ldots, p_m\}$. Let $T = \{\prod_{j=1}^m p_j^{x_j} | x_j \in \{0, \ldots, \ell\}\}$. Then the morphism*

$$\varphi : T \to (\mathcal{R}/\mathcal{I})^\times; \prod_{j=1}^m p_j^{x_j} \mapsto \prod_{j=1}^m p_j^{x_j} \pmod{\mathcal{I}}$$

is injective.

Proof. Our goal is to show that there does not exist two elements in T mapping to the same element in $(\mathcal{R}/\mathcal{I})^\times$. Consider the lattice given by Formula (1). Since $(P, (\mathcal{R}/\mathcal{I})^\times)$ is (m, ℓ, c)-Gaussian-predictable, we have that

$$\lambda_1(\Lambda) > \frac{1}{c} \cdot gh(\Lambda)$$

where

$$gh(\Lambda) := \sqrt{\frac{m}{2\pi e}} \mathrm{vol}(\Lambda)^{\frac{1}{m}}$$

and the volume $\mathrm{vol}(\Lambda)$ of the lattice Λ is given by the size of the image $\mathrm{im}(\phi)$ of ϕ by the First Group Isomorphism Theorem.

Now since $(P, (\mathcal{R}/\mathcal{I})^\times)$ is (m, ℓ, c)-Gaussian-predictable, P generates $(\mathcal{R}/\mathcal{I})^\times$. Hence ϕ is surjective, and thus $\mathrm{vol}(\Lambda) = \mathrm{im}(\phi) = |(\mathcal{R}/\mathcal{I})^\times|$. Again, by the Gaussian predictability of $(P, (\mathcal{R}/\mathcal{I})^\times)$, we have that $|(\mathcal{R}/\mathcal{I})^\times| > (c\ell\sqrt{2\pi e})^m$. We therefore have a lower bound for the first minimum:

$$\lambda_1(\Lambda) > (1/c) \cdot \sqrt{\frac{m}{2\pi e}} |(\mathcal{R}/\mathcal{I})^\times|^{\frac{1}{m}} > \ell\sqrt{m}.$$

Now suppose for contradiction that there exist two different elements in T mapping to the same element in $(\mathcal{R}/\mathcal{I})^\times$. That is, there exist two different vectors

$$(x_1, \ldots, x_m) \neq (y_1, \ldots, y_m) \in \{0, \ldots, \ell\}^m$$

such that

$$\prod_{j=1}^{m} p_j^{x_j} \neq \prod_{j=1}^{m} p_j^{y_j} \text{ but } \prod_{j=1}^{m} p_j^{x_j} \equiv \prod_{j=1}^{m} p_j^{y_j} \pmod{\mathcal{I}}.$$

Then

$$\prod_{j=1}^{m} p_j^{x_j - y_j} \equiv 1_{(\mathcal{R}/\mathcal{I})^\times} \pmod{\mathcal{I}}$$

with $\boldsymbol{z} := (x_1 - y_1, \ldots, x_m - y_m) \in \{-\ell, \ldots, \ell\}^m$ nonzero. In other words, there exist a nonzero vector $\boldsymbol{z} \in \Lambda$ of Euclidean norm $\leq \ell\sqrt{m}$ in a lattice whose first minimum is $> \ell\sqrt{m}$, a contradiction. □

Now we are ready to prove our theorem of unique solution.

Theorem 3.6. *Let $\mathcal{R}$ be an $(m+k, nB, c)$-Gaussian-predictable qualified Euclidean domain with Gaussian predictable pair $(P = \{p_1, \ldots, p_m\}, \mathcal{R}_q^\times)$, where $m, k, B = poly(n)$ and $1 \leq B < |\mathcal{R}_q| - 2$. Let $L = \{\ell_1, \ldots, \ell_k\}$ be a set of primes of $\mathcal{R}$ such that $P \cap L = \varnothing$, and let $\mathbb{z} = \{0, \ldots, B\}$. Let $\xi \cdot \boldsymbol{a}^{\boldsymbol{x}} = b \pmod q$ be an unknown-coefficient multivariate exponential equation with elements in $\boldsymbol{a}$ factored over P. Then the equation $\xi \cdot \boldsymbol{a}^{\boldsymbol{x}} = b \pmod q$ has at most one solution in $\mathbb{z}^n$ with the same coefficient ξ.*

Proof. Suppose for contradiction that there are two different solutions with the same coefficient. Namely there exists a pair of vectors

$$(u_1, \ldots, u_k, x_1, \ldots, x_n) \neq (u_1, \ldots, u_k, y_1, \ldots, y_n) \in \{0, 1\}^k \times \{0, \ldots, B\}^n$$

such that

$$\prod_{j=1}^{k} \ell_j^{u_j} \cdot \prod_{i=1}^{n} a_i^{x_i} \neq \prod_{j=1}^{k} \ell_j^{u_j} \cdot \prod_{i=1}^{n} a_i^{y_i} \text{ but } \prod_{j=1}^{k} \ell_j^{u_j} \cdot \prod_{i=1}^{n} a_i^{x_i} \equiv \prod_{j=1}^{k} \ell_j^{u_j} \cdot \prod_{i=1}^{n} a_i^{y_i} \pmod q.$$

This means that there exist two vectors

$$(u_1, \dots, u_k, e_1, \dots, e_m) \neq (u_1, \dots, u_k, f_1, \dots, f_m) \in \{0, \dots, nB\}^{k+m}$$

such that

$$\prod_{j=1}^{k} \ell_j^{u_j} \cdot \prod_{j=1}^{m} p_j^{e_j} \neq \prod_{j=1}^{k} \ell_j^{u_j} \cdot \prod_{j=1}^{m} p_j^{f_j} \quad \text{but} \quad \prod_{j=1}^{k} \ell_j^{u_j} \cdot \prod_{j=1}^{m} p_j^{e_j} \equiv \prod_{j=1}^{k} \ell_j^{u_j} \cdot \prod_{j=1}^{m} p_j^{f_j} \pmod{q}.$$

This contradicts Lemma 3.5. □

3.4. **NP-hardness.** Now we investigate the hardness of solving such equations. We show that the equation-solving problem is NP-hard if L is of polynomial size and it is easy to find a Gaussian-predictable pair.

We first define efficient-Gaussian-predictability to describe the easiness of finding a Gaussian-predictable pair $(\{p_1, \dots, p_m\}, (\mathcal{R}/\mathcal{I})^\times)$ given a ring $\mathcal{R}$.

Definition 3.7 (Efficient-Gaussian-predictability)**.** Let $\mathcal{R}$ be a ring. We say that $\mathcal{R}$ is efficient-Gaussian-predictable if for all $m \in \mathbb{N}$ and all $\ell \in \mathbb{N}$ polynomial in m, there exists a (deterministic) polynomial time algorithm that finds an $(m, \ell, 2^m)$-Gaussian-predictable pair $(\{p_1, \dots, p_m\}, (\mathcal{R}/\mathcal{I})^\times)$.

The following lemma shows that $\mathbb{Z}$ is an efficient-Gaussian-predictable ring under heuristic estimates regarding the density of Sophie Germain primes [25] and a bound on the smallest primitive root modulo a prime q [4].

Lemma 3.8. *Assuming the density $\pi_{SG}(x) \sim \frac{ax}{(\log x)^2}$ ($a \approx 1.3203$) of Sophie Germain primes, and the bound $O(\log^6 q)$ on the smallest primitive root g modulo a prime q, then for every $m \in \mathbb{N}$ and every $\ell \in \mathbb{N}$ polynomial in m, there exists a deterministic algorithm, running in time polynomial in m, that constructs an $(m, \ell, 2^m)$-Gaussian-predictable pair $(\{p_1, \dots, p_m\}, \mathbb{Z}_q^\times)$.*

Proof. Fix $m \in \mathbb{N}$ and $\ell \in \mathbb{N}$ with $\ell = O(m^\gamma)$ for some constant γ. Set the Gaussian-predictability parameter $c = 2^m$. We describe a deterministic polynomial-time algorithm that constructs an (m, ℓ, c)-Gaussian-predictable pair $(\{p_1, \dots, p_m\}, \mathbb{Z}_q^\times)$.

First, we select a Sophie Germain prime $r > (2^m \ell \sqrt{2\pi e})^m$ such that $q = 2r + 1$ is a safe prime. Let $x_0 = (2^m \ell \sqrt{2\pi e})^m$. Assuming the density of Sophie Germain primes [25], $\pi_{SG}(x) \sim \frac{ax}{(\log x)^2}$, the probability that an integer $n \approx x_0$ is a Sophie Germain prime is approximately $\frac{a}{(\log x_0)^2}$. Since $\log x_0 \approx m(m \log 2 + \gamma \log m + \frac{1}{2} \log(2\pi e)) \approx m^2 \log m$, the expected number of trials to find a Sophie Germain prime is $O((\log x_0)^2) = O(m^4)$. We enumerate odd integers $n > x_0$, testing each n and $2n + 1$ for primality using the Agrawal–Kayal–Saxena (AKS) primality test [1], which runs in time $O((\log n)^\alpha)$ for some constant α (typically $\alpha \approx 6$). Since $\log r \approx m^2 \log m$, each test takes $O((m^2 \log m)^\alpha)$ time. Testing $O(m^4)$ candidates yields a total time of $O(m^4 \cdot (m^2 \log m)^\alpha) = O(m^{2\alpha+4} (\log m)^\alpha)$, which is polynomial in m.

Next, we compute a generator g of $\mathbb{Z}_q^\times$, a cyclic group of order $q - 1 = 2r$. Since q is a safe prime, $\phi(q - 1) = \phi(2r) = r - 1$, and we need g such that $g^2 \not\equiv 1 \pmod{q}$ and $g^r \not\equiv 1 \pmod{q}$. We test candidates $g = 2, 3, \dots$, checking $g^2 \not\equiv 1 \pmod{q}$ and $g^r \not\equiv 1 \pmod{q}$ using modular exponentiation, which takes $O(\log q) = O(m^2 \log m)$ time per test. Assuming the smallest primitive root is at most $O(\log^6 q) = O((m^2 \log m)^6)$ [4], we test $O((m^2 \log m)^6)$ candidates, for a total time of $O((m^2 \log m)^6 \cdot (m^2 \log m)) = O(m^{14} (\log m)^7)$, which is polynomial in m.

We construct the exponent vector $a = (a_1, \ldots, a_m) \in \mathbb{Z}^m$ by setting $a_j = j$ for $j = 1, \ldots, m$. This ensures the a_j are distinct and linearly independent over $\mathbb{Z}$, taking $O(m \log m)$ time to store m integers of bit length $O(\log m)$.

For each $j \in \{1, \ldots, m\}$, we find a prime $p_j \equiv g^{a_j} \pmod{q}$. Compute $\alpha_j = g^{a_j} \bmod q$ using modular exponentiation, taking $O(\log a_j \cdot \log q) = O(\log m \cdot m^2 \log m)$ time per j, or $O(m^3 (\log m)^2)$ for all m indices. Since $\gcd(\alpha_j, q) = 1$, Dirichlet's theorem [16] guarantees infinitely many primes in the residue class $\alpha_j \pmod{q}$. The prime number theorem for arithmetic progressions [3] implies that the number of primes in the progression $\alpha_j + k_j q$ up to x is approximately $\frac{x}{\phi(q) \log x}$, with $\phi(q) = q - 1 = 2r$. We test candidates $\alpha_j + k_j q$ for $k_j = 1, 2, \ldots$ using the AKS primality test. The first prime is expected at $k_j \approx \phi(q) \log \phi(q)/q \approx m^2 \log m$. Each candidate $\alpha_j + k_j q$ has bit length $O(\log q + \log k_j) = O(m^2 \log m)$, so each AKS test takes $O((m^2 \log m)^\alpha)$ time. Testing $O(m^2 \log m)$ candidates per j yields a time of $O((m^2 \log m) \cdot (m^2 \log m)^\alpha) = O((m^2 \log m)^{\alpha+1})$ per prime. For m primes, the total time is $O(m \cdot (m^2 \log m)^{\alpha+1}) = O(m^{2\alpha+3} (\log m)^{\alpha+1})$, which is polynomial in m. With extra checks to guarantee distinct primes, the time complexity remains polynomial in m. Collect these primes into the set $\{p_1, \ldots, p_m\}$.

Note that each $p_j = \alpha_j + k_j q$ with $k_j \geq 1$, so $p_j \geq q$. Thus, $\prod_{j=1}^m p_j \geq q^m \geq q$, ensuring this is not the trivial case where q is larger than the product of all p_j.

Finally, we verify the Gaussian-predictability conditions. Since $p_j \equiv g^{a_j} \pmod{q}$ and $\{a_j\} = \{1, \ldots, m\}$ are distinct, $\{p_j \mod q\} = \{g^j \mod q\}$ generates $\mathbb{Z}_q^\times$, satisfying condition (1). For condition (2), we need $q - 1 = 2r > (c\ell\sqrt{2\pi e})^m = (2^m \ell \sqrt{2\pi e})^m$. Since $r > (2^m \ell \sqrt{2\pi e})^m$, this holds. For condition (3), consider the lattice $\Lambda = \{z \in \mathbb{Z}^m : \langle a, z \rangle \equiv 0 \pmod{q-1}\}$. The determinant is $\det(\Lambda) = q - 1 = 2r$, so $gh(\Lambda) = \sqrt{\frac{m}{2\pi e}} \cdot (2r)^{1/m}$. For any $z \in \Lambda \setminus \{0\}$, Cauchy–Schwarz gives $|\langle a, z \rangle| \leq \|a\| \cdot \|z\|$, where $\|a\| = \sqrt{\sum_{j=1}^m j^2} = \sqrt{\frac{m(m+1)(2m+1)}{6}}$. Thus, $\lambda_1(\Lambda) = \min \|z\| \geq \frac{q-1}{\|a\|}$. The ratio is:

$$\frac{gh(\Lambda)}{\lambda_1(\Lambda)} \leq \sqrt{\frac{m}{2\pi e}} \cdot (2r)^{\frac{1}{m}} \cdot \sqrt{\frac{m(m+1)(2m+1)}{6}} \cdot \frac{1}{q-1}.$$

Since $q - 1 = 2r$, this simplifies to:

$$\frac{gh(\Lambda)}{\lambda_1(\Lambda)} \leq \sqrt{\frac{m^2(m+1)(2m+1)}{12\pi e}} \cdot 2^{-1+\frac{1}{m}} \cdot r^{-1+\frac{1}{m}}.$$

Since $r > (2^m \ell \sqrt{2\pi e})^m$, we have $r^{-1+\frac{1}{m}} < (2^m \ell \sqrt{2\pi e})^{-(m-1)}$. For $m \geq 2$, since $\sqrt{\frac{m^2(m+1)(2m+1)}{12\pi e}} < \sqrt{\frac{2m^2 \cdot m \cdot 2m}{12\pi e}} = m^2 \sqrt{\frac{1}{3\pi e}} < m^2$, and $2^{-1+1/m} \leq 1$, we have

$$\frac{gh(\Lambda)}{\lambda_1(\Lambda)} < m^2 \cdot 2^{-1+\frac{1}{m}} \cdot (2^m \ell \sqrt{2\pi e})^{-(m-1)} \ll 2^m,$$

as the exponential decay dominates. For $m = 1$, we calculate directly: $\Lambda = \{z_1 \in \mathbb{Z} : z_1 \equiv 0 \pmod{q-1}\}$, so $gh(\Lambda) = \sqrt{\frac{1}{2\pi e}} \cdot (q-1)$, $\lambda_1(\Lambda) = q - 1$, and $\frac{gh(\Lambda)}{\lambda_1(\Lambda)} = \frac{1}{\sqrt{2\pi e}} < 2^m = 2$. Thus, $gh(\Lambda) < 2^m \cdot \lambda_1(\Lambda)$ holds.

The algorithm is deterministic under the stated heuristic assumptions, as no step involves randomness.[3] The total running time is polynomial in m. This completes the proof. □

Now we prove NP-hardness based on Definition 3.7. Note that the proof of the following general theorem does not rely on the concrete example in the proof of Lemma 3.8.

Theorem 3.9. *The problem of solving an unknown-coefficient multivariate exponential equation over an efficient-Gaussian-predictable qualified Euclidean domain $\mathcal{R}$ with a polynomial size coefficient set $L = \{\ell_1, \ldots, \ell_k\}$ is NP-hard.*

Proof. We reduce the exact cover problem to the equation-solving problem. Let $(S = \{s_1, \ldots, s_m\}, C = \{C_1, \ldots, C_n\})$ be an exact cover instance, where S is a set of elements s_i and C is a collection of subsets of S. Define the characteristic matrix of (S, C) to be the binary matrix $A \in \{0,1\}^{m \times n}$ where the i-th column $A_{*,i}$ is the characteristic vector of the i-th subset C_i for $i = 1, \ldots, n$.

Step 1. Find a $(k + m, n, 2^{k+m})$-Gaussian-predictable pair $(P := \{\ell_1, \ldots, \ell_k, p_1, \ldots, p_m\}, \mathcal{R}_q)$, where $\ell_1, \ldots, \ell_k, p_1, \ldots, p_m$ are prime elements of $\mathcal{R}$, and q is a prime of $\mathcal{R}$.

Step 2. Compute $a_1, \ldots, a_n \in \mathcal{R}$ with $a_i = \prod_{j=1}^{m} p_j^{A_{j,i}}$ for all $i \in \{1, \ldots, n\}$. Compute $X = \prod_{j=1}^{m} p_j$. By unique factorization of elements of $\mathcal{R}$, a vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ is a solution to the equation $\boldsymbol{a}^{\boldsymbol{x}} = X$ if and only if it is a solution to the exact cover instance (S, C).

Step 3. Choose any binary vector $(u_1, \ldots, u_k) \in \{0,1\}^k$ with precisely one element being "1". Compute $\xi = \prod_{j=1}^{k} \ell_j^{u_j}$ and $Y = \xi \cdot X$. Again, by unique factorization, a vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ is a solution to the equation $\xi \cdot \boldsymbol{a}^{\boldsymbol{x}} = Y$ if and only if it is a solution to the equation $\boldsymbol{a}^{\boldsymbol{x}} = X$.

Step 4. Now compute $Z := Y \mod q$. We have an equation $\xi \cdot \boldsymbol{a}^{\boldsymbol{x}} = Z \pmod q$. We want to show that it has the same set of solutions as the equation $\xi \cdot \boldsymbol{a}^{\boldsymbol{x}} = Y$. For one direction, it is obvious that all solutions to $\xi \cdot \boldsymbol{a}^{\boldsymbol{x}} = Y$ are solutions to $\xi \cdot \boldsymbol{a}^{\boldsymbol{x}} = Z \pmod q$. For the other direction, suppose for contradiction that there exists a solution to $\xi \cdot \boldsymbol{a}^{\boldsymbol{x}} = Z \pmod q$ that is not a solution to $\xi \cdot \boldsymbol{a}^{\boldsymbol{x}} = Y$. That is, there exists a pair of vectors

$$(u_1, \ldots, u_k, x_1, \ldots, x_n) \neq (v_1, \ldots, v_k, y_1, \ldots, y_n) \in \{0,1\}^{k+n}$$

such that

$$\prod_{j=1}^{k} \ell_j^{u_j} \cdot \prod_{i=1}^{n} a_i^{x_i} \neq \prod_{j=1}^{k} \ell_j^{v_j} \cdot \prod_{i=1}^{n} a_i^{y_i} \quad \text{but} \quad \prod_{j=1}^{k} \ell_j^{u_j} \cdot \prod_{i=1}^{n} a_i^{x_i} \equiv \prod_{j=1}^{k} \ell_j^{v_j} \cdot \prod_{i=1}^{n} a_i^{y_i} \pmod q.$$

This means that there exist two vectors

$$(u_1, \ldots, u_k, e_1, \ldots, e_m) \neq (v_1, \ldots, v_k, f_1, \ldots, f_m) \in \{0, \ldots, n\}^{k+m}$$

[3]Note that the probabilistic arguments (e.g., regarding the density of Sophie Germain primes and the distribution of primes in arithmetic progressions) are used solely for analyzing the time complexity, not for the correctness or determinism of the algorithm itself. They do not introduce randomness to the algorithm.

such that

$$\prod_{j=1}^{k} \ell_j^{u_j} \cdot \prod_{j=1}^{m} p_j^{e_j} \neq \prod_{j=1}^{k} \ell_j^{v_j} \cdot \prod_{j=1}^{m} p_j^{f_j} \text{ but } \prod_{j=1}^{k} \ell_j^{u_j} \cdot \prod_{j=1}^{m} p_j^{e_j} \equiv \prod_{j=1}^{k} \ell_j^{v_j} \cdot \prod_{j=1}^{m} p_j^{f_j} \pmod{q}.$$

This contradicts Lemma 3.5. We conclude that $\boldsymbol{x} = (x_1, \ldots, x_n)$ is a solution to $\xi \cdot \boldsymbol{a}^{\boldsymbol{x}} = Z \pmod{q}$ if and only if it is a solution to $\xi \cdot \boldsymbol{a}^{\boldsymbol{x}} = Y$.

Now we analyze the time complexity of the reduction. By the definitions of Gaussian-predictability and efficient-Gaussian-predictability, Step 1 can be finished in polynomial time. Also, Steps 2, 3 and 4 can be finished in polynomial time. Combine them together, the reduction succeeds in polynomial time. This completes the proof. □

4. Multivariate exponential system with unknown coefficients

Now we consider the problem of solving a system of random equations. This problem gives a wide range of applications in cryptography (see Section 5).

4.1. **Multivariate-exponential-system-solving problem.** We define the *multivariate-exponential-system-solving* (MESS) problem as follows.

Let $\mathcal{R}$ be a qualified Euclidean domain.
Let L be a set of primes of $\mathcal{R}$.[4]
Let q be a prime of $\mathcal{R}$.[5]
Let $\mathbb{z}$ be a subset of $\mathbb{Z}_{\geq 0}$.[6]
Let $\boldsymbol{x} \leftarrow \mathbb{z}^n$.
Let $O_{\boldsymbol{x}}$ be the oracle that outputs pairs $(\boldsymbol{a}, b)$ where $\boldsymbol{a} \leftarrow (\mathcal{R}_q^{\times})^n$, $\xi \leftarrow L$ and $b := \xi \cdot \boldsymbol{a}^{\boldsymbol{x}} \pmod{q}$.
Let O_{ran} be the oracle that outputs uniform pairs $(\boldsymbol{a}, b) \leftarrow (\mathcal{R}_q^{\times})^n \times \mathcal{R}_q^{\times}$.

Definition 4.1 (Search-MESS)**.** Given $O_{\boldsymbol{x}}$, find $\boldsymbol{x}$.

Definition 4.2 (Decision-MESS)**.** Given either $O_{\boldsymbol{x}}$ or O_{ran}, decide which is the case.

We call each pair $(\boldsymbol{a}, b)$ from $O_{\boldsymbol{x}}$ a MESS instance and each pair $(\boldsymbol{a}, b)$ from O_{ran} a random/uniform instance. We also refer to the sequence of b's output by $O_{\boldsymbol{x}}$ the *result vector* of the MESS problem.

4.2. **Examples.** Following are two concrete examples of the MESS problem.
Example (1): **MESS over $\mathbb{Z}$.**

Let $\mathbb{Z}$ be the ring of integers.
Let $L := \{\ell_1, \ldots, \ell_k\}$ be the set of the first k odd primes.
Let q be a safe prime.[7]
Let $\boldsymbol{x} \leftarrow \{0,1\}^n$.
Let $O_{\boldsymbol{x}}$ be the oracle that outputs pairs $(\boldsymbol{a}, b)$ where $\xi \leftarrow L$, $\boldsymbol{a} \leftarrow (\mathbb{Z}_q^{\times})^n$, and $b := \xi \cdot \boldsymbol{a}^{\boldsymbol{x}} \pmod{q}$.
Let O_{ran} be the oracle that outputs uniform instances $(\boldsymbol{a}, b) \leftarrow (\mathbb{Z}_q^{\times})^n \times \mathbb{Z}_q^{\times}$.
– The Search-MESS$_{\mathbb{Z}}$ problem is to find $\boldsymbol{x}$ from $O_{\boldsymbol{x}}$.

[4]The size of L can be polynomial or super-polynomial in n.
[5]The degree of q is often a key parameter for different applications.
[6]The set $\mathbb{z}$ can be finite or infinite.
[7]A safe prime is a prime number of the form $q = 2p + 1$, where p is also a prime, known as a Sophie Germain prime.

– The Decision-MESS$_{\mathbb{Z}}$ problem is to distinguish $O_{\boldsymbol{x}}$ and O_{ran}.

Example (2): **MESS over $\mathbb{F}_2[x]$.**

Let $\mathbb{F}_2[x]$ be the univariate polynomial ring over the finite field $\mathbb{F}_2$.
Let L be a set of primes of $\mathbb{F}_2[x]$ of degree $\leq k$.
Let q be a prime polynomial in $\mathbb{F}_2[x]$.
Let $\boldsymbol{x} \leftarrow \{0,1\}^n$.
Let $O_{\boldsymbol{x}}$ be the oracle that outputs pairs $(\boldsymbol{a}, b)$ where $\xi \leftarrow L$, $\boldsymbol{a} \leftarrow ((\mathbb{F}_2[x]/(q))^\times)^n$, and $b := \xi \cdot \boldsymbol{a}^{\boldsymbol{x}} \pmod q$.
Let O_{ran} be the oracle that outputs uniform instances $(\boldsymbol{a}, b) \leftarrow ((\mathbb{F}_2[x]/(q))^\times)^n \times (\mathbb{F}_2[x]/(q))^\times$.

– The Search-MESS$_{\mathbb{F}_2[x]}$ problem is to find $\boldsymbol{x}$ from $O_{\boldsymbol{x}}$.
– The Decision-MESS$_{\mathbb{F}_2[x]}$ problem is to distinguish $O_{\boldsymbol{x}}$ and O_{ran}.

Note that the product degrees in the two problems grow in very different ways: for $a, b \in \mathbb{Z}$, we have $d(ab) = d(a) \cdot d(b)$; whereas for $a, b \in \mathbb{F}_2[x]$, we have $d(ab) = d(a) + d(b)$. Namely, the degree grows multiplicatively in MESS$_{\mathbb{Z}}$ but additively in MESS$_{\mathbb{F}_2[x]}$. This lower speed of degree growth in MESS$_{\mathbb{F}_2[x]}$ can result in smaller parameters or higher efficiency in applications that involve decryption, for which the degree $d(q)$ of the modulus q is usually required to be greater than the degree of a certain product.

In particular, a lower rate of degree growth can reduce the need of bootstrapping in homomorphic encryption. That is, it allows to perform a larger number of computations on encrypted data. For example, in the case of $\mathbb{F}_\ell[x]$, all degree-1 polynomials of the form $f(x) = ax + b$ with $a, b \in \mathbb{F}_\ell$ and $a \neq 0$ are prime. If each time we multiply by such a degree-1 prime, then each time the degree of the product increases by only 1. This means that a degree d modulus q can allow nearly d computations on encrypted data.

From the above two examples, we observe that different Euclidean domains can endow the MESS problem with distinct algebraic and structural properties. This highlights the theoretical significance and practical relevance of studying MESS over a broad range of Euclidean domains.

Common examples of Euclidean domains include the ring of univariate polynomials over a field, rings of Gaussian, Eisenstein, and quadratic integers, and rings of algebraic integers in general. Further examples arise from cyclotomic and cubic integer rings, function fields, and certain coordinate rings naturally occurring in algebraic geometry. These domains not only reflect the mathematical richness of Euclidean structures, but also suggest diverse directions for exploring the computational and cryptographic potential of MESS.

4.3. **Uniqueness of solutions.**

Theorem 4.3. *MESS over a $(k{+}m, nB, c)$-Gaussian-predictable qualified Euclidean domain $\mathcal{R}$ with $L = \{\ell_1, \ldots \ell_k\}$ for $k = poly(n)$ and $\mathbb{z} = \{0, \ldots, B\}$ for $1 \leq B < |\mathcal{R}_q| - 2$ has a unique solution.*

Proof. By the proof of Theorem 3.6, a single instance of MESS can already fix $\boldsymbol{x}$ hence fixing the coefficients $\xi_1, \ldots, \xi_k$. □

4.4. **Hardness.**

We prove the hardness of MESS with uniform coefficient distribution from the hardness of LWE [40] with uniform error distribution [37, 47].

Definition 4.4. Let S be a subset of $\mathbb{Z}_s$, $\mathbb{z}$ be a subset of $\mathbb{Z}_{\geq 0}$, and $\boldsymbol{s} \leftarrow \mathbb{z}^n$. Let $O_{\boldsymbol{s}}$ be the oracle that outputs pairs $(\boldsymbol{a}, b)$ where $\boldsymbol{a} \leftarrow \mathbb{Z}_s^n$, $e \leftarrow S$ and $b := \boldsymbol{a} \cdot \boldsymbol{s} + e \pmod{s}$. The LWE problem is given $O_{\boldsymbol{s}}$, find $\boldsymbol{s}$.

Theorem 4.5. *LWE $\leq$ MESS.*

Proof. We reduce LWE with parameters (n, s, S) to MESS with parameters (n, q, L), where $s = |\mathcal{R}_q| - 1$ is the size of $\mathcal{R}_q^\times$, $S = \log_g(L)$ is the set of discrete logarithms $\log_g(\ell_i)$ of the elements ℓ_i of L, and g is a generator of $\mathcal{R}_q^\times$. We define a MESS oracle $O_{\boldsymbol{x}}$ from the LWE oracle $O_{\boldsymbol{s}}$ as follows. For every LWE instance $(\boldsymbol{a}, b)$ output by $O_{\boldsymbol{s}}$, the oracle $O_{\boldsymbol{x}}$ simply outputs $(\boldsymbol{c}, d) = (g^{\boldsymbol{a}} \bmod q, g^b \bmod q)$. Note that $d = g^b = g^{\boldsymbol{a}\cdot\boldsymbol{s}+e} = g^e \cdot (g^{\boldsymbol{a}})^{\boldsymbol{s}} = \xi \cdot \boldsymbol{c}^{\boldsymbol{s}} \pmod{q}$, where $\xi = g^e \bmod q$ is a uniform element of L. It is clear that $O_{\boldsymbol{x}}$ is a MESS oracle and that any solution $\boldsymbol{x}$ to the MESS problem is a solution $\boldsymbol{s}$ to the LWE problem. In particular, if the LWE has a unique solution, then $\boldsymbol{x} = \boldsymbol{s}$. □

The other direction is also true if MESS instances $(\boldsymbol{a}, b)$ can be efficiently turned into LWE instances $(\log_g(\boldsymbol{a}) \bmod s, \log_g(b) \bmod s)$ by taking discrete logarithms. A quantum reduction may be obtained by extending Shor's algorithm [43, 44] to work over different cyclic groups $\mathcal{R}_q^\times$ for different Euclidean domains $\mathcal{R}$.

Now we provide some intuitions to explain why the MESS-related LWE problem is hard, and why MESS is not a classical lattice problem. Note that, although LWE is usually phrased as a linear system with "small" errors, its hardness does not depend on the absolute magnitudes of the errors. In fact, if s is a prime, an error set E consisting of small numbers can always be shifted to a set $E' = \{\alpha e + \beta : e \in E\}$ of large numbers via an affine transformation $e \mapsto e' = \alpha e + \beta$, where $\alpha, \beta \in \mathbb{Z}_s$. In this case, given LWE instances $(\boldsymbol{a}, b = \boldsymbol{a} \cdot \boldsymbol{s} + e)$ with errors e in E, one can efficiently transform them to LWE instances $(\boldsymbol{a}', b' = \boldsymbol{a}' \cdot \boldsymbol{s} + e') = (\alpha\mathbf{a}, \alpha b + \beta)$ with errors $e' = \alpha e + \beta$ in E', such that solving the latter immediately solves the former. Even in the case that s is composite and not every element in $\mathbb{Z}_s$ is invertible, we can still shift the error set by setting $\alpha = 1$. For instance, shifting $E = \{-8, \ldots, 8\}$ to $E' = \{992, \ldots, 1008\}$ via $e \mapsto e' = e + 1000$ does not affect the hardness of the LWE problem.

However, in the MESS-related LWE problem, the errors $\log_g(\xi) \bmod s$ are discrete logarithms of random primes $\xi \in L$ of the Euclidean domain $\mathcal{R}$, they generally do not have a simple linear relation with typical LWE errors. Therefore, the error vector of the MESS-related LWE problem do not have a direct correspondence to the shortest or closest vectors in the *shortest vector problem* (SVP) or *closest vector problem* (CVP) [36], which separates MESS from classical lattice problems.

To the best of our knowledge, no efficient quantum algorithms are known for solving the MESS-related LWE problem. This positions MESS as a novel and promising foundation for post-quantum cryptography.

Nonetheless, it is worth noting that some special cases of MESS indeed inherit the worst-case hardness of lattice problems. For example, consider the case where $L = \{g^{-B}, \ldots, g^B\}$ is a set of primes of $\mathcal{R}$, with g a generator of $\mathcal{R}_q^\times$ and $B \geq 2$. In this setting, the corresponding LWE problem with uniform errors drawn from the error set $S = \{-B, \ldots, B\}$ is at least as hard as worst-case lattice problems [37, 47].

As a final remark of this subsection, the reduction between MESS and LWE highlights that MESS extends LWE with additional structural richness, thereby enhancing its theoretical expressiveness and potential applicability without compromising its computational hardness. We will see in Section 5 that some of the

MESS applications are not easy to achieve using LWE, and that they are enabled primarily by the additional structure introduced by MESS.

4.5. **Random self-reducibility.** Now we show that MESS with uniform $\boldsymbol{x}$ is as hard as the problem with worst-case $\boldsymbol{x}$ when $\mathbb{z} = \mathbb{Z}_{|\mathcal{R}_q^\times|}$.

Lemma 4.6. *If $\mathbb{z} = \mathbb{Z}_{|\mathcal{R}_q^\times|}$ then MESS with average-case solution $\boldsymbol{x} \leftarrow \mathbb{z}^n$ is as hard as MESS with worst-case solution $\boldsymbol{x} \in \mathbb{z}^n$.*

Proof. For each instance $(a_1, \ldots, a_n, b)$ from the MESS oracle $O_{\boldsymbol{x}}$ with respect to an arbitrary solution $\boldsymbol{x} \in \mathbb{Z}_{|\mathcal{R}_q^\times|}^n$ such that $\xi \cdot \prod_{i=1}^n a_i^{x_i} \equiv a \pmod q$, sample a random vector $\boldsymbol{y} \leftarrow \mathbb{Z}_{|\mathcal{R}_q^\times|}^n$ and compute $\prod_{i=1}^n a_i^{y_i} \equiv c \pmod q$ and $d \equiv bc \pmod q$. Call the MESS solver with instances of the form $(a_1, \ldots, a_n, d)$. Note that

$$\xi \cdot \prod_{i=1}^n a_i^{x_i+y_i} \equiv d \pmod q$$

and y_i are uniform and thus $x_i + y_i \pmod{|\mathcal{R}_q^\times|}$ are uniform. Hence the solver will return $\boldsymbol{z} \equiv \boldsymbol{x} + \boldsymbol{y} \pmod{|\mathcal{R}_q^\times|}$ and we have $\boldsymbol{x} \equiv \boldsymbol{z} - \boldsymbol{y} \pmod{|\mathcal{R}_q^\times|}$. □

4.6. **Search-to-decision reduction.** Now we show the hardness of Decision-MESS. We show that Search-MESS and Decision-MESS are computationally equivalent when $\mathbb{z} = \{0, 1\}$.

Lemma 4.7. *If $\mathbb{z} = \{0, 1\}$ then Search-MESS = Decision-MESS.*

Proof. The direction Decision-MESS $\leq$ Search-MESS is clear (in fact for all $\mathbb{z} \subseteq \mathbb{Z}$). We prove the direction Search-MESS $\leq$ Decision-MESS (for $\mathbb{z} = \{0, 1\}$).

Assume a noticeable advantage distinguisher $\mathcal{D}$ for Decision-MESS. We show how to use it to learn each entry x_j of $\boldsymbol{x} \in \{0, 1\}^n$ from $poly(n)$ MESS instances as follows.

Ask $O_{\boldsymbol{x}}$ for a MESS instance

$$(a_1, \ldots, a_n, b),$$

where

$$b \equiv \xi \cdot \prod_{i=1}^n a_i^{x_i} \pmod q.$$

Sample a random element

$$r \leftarrow \mathcal{R}_q^\times.$$

Let

$$b_j := a_j \cdot r \pmod q; \text{ and } b_i = a_i \text{ for all } i \neq j, i \in \{1, \ldots, n\}.$$

Let

$$c := b \cdot r \pmod q.$$

Call $\mathcal{D}$ with

$$(b_1, \ldots, b_n, c)$$

and record the output bit of $\mathcal{D}$. Repeat the above process for $poly(n)$ MESS instances and output $x_j = 1$ if $\mathcal{D}$ outputs 1 more than $poly(n)/2$ times, or output $x_j = 0$ otherwise.

We show how this works. We first look at the base vector $(b_1, \ldots, b_n)$. Note that both a_j and r are uniform over $\mathcal{R}_q^\times$. Hence $b_j = a_j \cdot r \pmod q$ is uniform. Thus $(b_1, \ldots, b_n)$ obeys the same distribution as the base vectors of both $O_{\boldsymbol{x}}$ and O_{ran}.

Now we look at the target number c. Note that

$$c \equiv \xi \cdot \prod_{i=1}^{n} a_i^{x_i} \cdot r \equiv \left(\xi \cdot \prod_{i=1}^{n} b_i^{x_i} \right) \cdot r^{1-x_j} \pmod{q}.$$

Hence c is a MESS product with respect to $(b_1, \ldots, b_n)$ if $x_j = 1$; and it is a uniform element if $x_j = 0$ because r is uniform. In other words, the tuple $(b_1, \ldots, b_n, c)$ is a MESS instance from $O_{\boldsymbol{x}}$ if $x_j = 1$; and it is a random instance from O_{ran} if $x_j = 0$. We see that the advantage of learning x_j is the same as the advantage of $\mathcal{D}$ distinguishing D_1 and D_2, which is noticeable by hypothesis. By the typical Chernoff bound argument, with polynomially many trials we can amplify the success probability of learning x_j to approximately 1. □

4.7. **Trapdoorizability.** In cryptographic applications we often use a traopdoorized version of MESS that is (as will be shown in the following) as hard as MESS.

Definition 4.8. A MESS instance is called a *trapdoor instance* if its base vector $\boldsymbol{a}$ is not sampled uniformly but defined as $\boldsymbol{a} := \boldsymbol{v}$ or $\boldsymbol{a} := \boldsymbol{v}^{\circ -1}$ for some result vector $\boldsymbol{v} = (b_1, \ldots, b_n)$, $b_i = \xi_i \cdot \boldsymbol{a}_i^{\boldsymbol{s}} \pmod{q}$ of another MESS problem $O_{\boldsymbol{s}}$ with respect to independent and uniform coefficients $\xi_i \leftarrow L$ and secret $\boldsymbol{s} \leftarrow \{0,1\}^n$. We call the Search-MESS and Decision-MESS with at least one instance being a trapdoor instance the trapdoored MESS problems, and denote them by Search-MESS' and Decision-MESS'.

We have the following lemma.

Lemma 4.9. *Decision-MESS $\leq$ Decision-MESS'.*

Proof. We only prove the case where there is only one trapdoor instance. By a hybrid argument it can be generalized to the case of multiple trapdoor instances. Let k be a positive integer polynomial in n. Let $\boldsymbol{\xi}$ be a coefficient vector, $\boldsymbol{A} \leftarrow (\mathcal{R}_q^\times)^{k \times n}$ be a base matrix, $\boldsymbol{x} \leftarrow \mathbb{z}^n$ be a secret, $\boldsymbol{v} \in (\mathcal{R}_q^\times)^n$ be a result vector with respect to independent and uniform coefficient and secret vector. Let $\boldsymbol{B} := (\boldsymbol{A}, \boldsymbol{v})^\top$, $\boldsymbol{b} := \boldsymbol{\xi} \circ \boldsymbol{B}^{\boldsymbol{x}} \in (\mathcal{R}_q^\times)^{k+1}$, $\boldsymbol{r} \leftarrow (\mathcal{R}_q^\times)^n$, $\boldsymbol{C} := (\boldsymbol{A}, \boldsymbol{r})^\top$, $\boldsymbol{c} = \boldsymbol{\xi} \circ \boldsymbol{C}^{\boldsymbol{x}} \pmod{q} \in (\mathcal{R}_q^\times)^{k+1}$, $\boldsymbol{d} \leftarrow (\mathcal{R}_q^\times)^{k+1}$. The two distributions of Decision-MESS' are $(\boldsymbol{B}, \boldsymbol{b})$ and $(\boldsymbol{C}, \boldsymbol{d})$. Our goal is to show that they are indistinguishable. For this, consider the intermediate distribution $(\boldsymbol{C}, \boldsymbol{c})$. By the hardness of Decision-MESS, it can be verified that (1) $(\boldsymbol{B}, \boldsymbol{b})$ and $(\boldsymbol{C}, \boldsymbol{c})$ are indistinguishable; and (2) $(\boldsymbol{C}, \boldsymbol{c})$ and $(\boldsymbol{C}, \boldsymbol{d})$ are indistinguishable. We therefore conclude that $(\boldsymbol{B}, \boldsymbol{b})$ and $(\boldsymbol{C}, \boldsymbol{d})$ are indistinguishable. □

Combining Lemma 4.7 and 4.9 we have the following theorem.

Theorem 4.10. *If $\mathbb{z} = \{0,1\}$ then Search-MESS = Decision-MESS = Search-MESS' = Decision-MESS'.*

Proof. By Lemma 4.7 we have Search-MESS = Decision-MESS. Also, by Lemma 4.9 we have Decision-MESS $\leq$ Decision-MESS'. Again, it is clear that Decision-MESS' $\leq$ Search-MESS' (obvious) and Search-MESS' $\leq$ Search-MESS (by ignoring the trapdoor instances). Combining the above results, we obtain the desired equivalence chain. □

5. Cryptography from multivariate exponential equations

In this section, we explore the possibilities of MESS in constructing various cryptographic primitives. The resulting primitives are general, in the sense that they work for all qualified Euclidean domains.

Our goal here is to demonstrate the expressive power and flexibility of the new equations. We view the applications as conceptual demonstrations of the potential of the MESS framework, and leave detailed efficiency analysis and concrete parameter instantiations for future work.

Although we do not focus on direct comparisons with existing constructions based on LWE or MQ assumptions, it is worth noting that some of our constructions, such as reusable obfuscation and reusable fuzzy extractor, are highly efficient. Moreover, these primitives are not known to be achievable under LWE or MQ assumptions in such a simple way, due to the lack of structural properties like prime elements in their algebraic settings.

These applications highlight the structural richness of the new equations and their potential as a unifying foundation for a variety of cryptographic tasks.

5.1. **Homomorphic encryption.** Let $\boldsymbol{A} \leftarrow (\mathcal{R}_q^{\times})^{m\times n}$. Define the MESS function as

$$f_{\boldsymbol{A}}(\boldsymbol{\xi}, \boldsymbol{x}) = \boldsymbol{\xi} \circ \boldsymbol{A}^{\boldsymbol{x}} \pmod q.$$

The function is homomorphic in the sense that:

$$(f_{\boldsymbol{A}_1}(\boldsymbol{\xi}_1, \boldsymbol{x})) \circ (f_{\boldsymbol{A}_2}(\boldsymbol{\xi}_2, \boldsymbol{x})) \equiv f_{\boldsymbol{A}_1 \circ \boldsymbol{A}_2}(\boldsymbol{\xi}_1 \circ \boldsymbol{\xi}_2, \boldsymbol{x}) \pmod q.$$

Let p be a prime in $\mathcal{R}\backslash L$ such that the degree $d(\xi \cdot p) < d(q)$ for all $\xi \in L$. To encrypt a message $\boldsymbol{m} = (m_1, \ldots, m_k) \in \{0,1\}^k$ with a symmetric key $\boldsymbol{x}$, compute

$$\boldsymbol{b} = f_{\boldsymbol{A}}(\boldsymbol{\xi}, \boldsymbol{x}) \circ p^{\boldsymbol{m}} \pmod q$$

and the ciphertext is $Enc(\boldsymbol{x}, \boldsymbol{m}) = (\boldsymbol{A}, \boldsymbol{b})$. To decrypt, compute

$$\begin{aligned}
\boldsymbol{d} &= \boldsymbol{b} \circ (\boldsymbol{f}_{\boldsymbol{A}}(\boldsymbol{1}, \boldsymbol{x}))^{\circ -1} \pmod q \\
&\equiv \boldsymbol{\xi} \circ \boldsymbol{A}^{\boldsymbol{x}} \circ p^{\boldsymbol{m}} \circ \boldsymbol{1}^{\circ -1} \circ \boldsymbol{A}^{-\boldsymbol{x}} \pmod q \\
&\equiv \boldsymbol{\xi} \circ p^{\boldsymbol{m}} \pmod q \\
&= \boldsymbol{\xi} \circ p^{\boldsymbol{m}},
\end{aligned}$$

where the last line is due to $d(\xi \cdot p) < d(q)$ for all $\xi \in L$. Denote $\boldsymbol{d} = (d_1, \ldots, d_k)$. Then the message is $\boldsymbol{m} = (m_1, \ldots, m_k)$ where $m_i = 1$ if and only if $p | d_i$.

The semantic security is given by the following theorem.

Theorem 5.1. *If Decision-MESS is hard, then the ciphertext $(\boldsymbol{A}, \boldsymbol{b})$ is indistinguishable from uniform.*

Proof. By the hardness of Decision-MESS, $f_{\boldsymbol{A}}(\boldsymbol{\xi}, \boldsymbol{x})$ is indistinguishable from uniform. Hence $(\boldsymbol{A}, \boldsymbol{b})$ is indistinguishable from uniform regardless of what $p^{\boldsymbol{m}}$ is. □

The homomorphic property is given by the following theorem.

Theorem 5.2. *If $d(\xi_1 \cdot \xi_2 \cdot p^2) < d(q)$ for all $\xi_1, \xi_2 \in L$, then the encryption scheme satisfies*

$$Enc(sk, \boldsymbol{m}_1) \cdot Enc(sk, \boldsymbol{m}_2) \approx Enc(sk, \boldsymbol{m}_1 + \boldsymbol{m}_2),$$

where $\approx$ means there exists an efficient algorithm to decrypt both sides.

Proof. We want to show that both sides are equal, up to two different coefficient factors on each side for every row.

On the one hand, by the homomorphic property of the MESS function, we have

$$Enc(sk, \boldsymbol{m}_1) \cdot Enc(sk, \boldsymbol{m}_2) = (\boldsymbol{A}_1 \circ \boldsymbol{A}_2, f_{\boldsymbol{A}_1 \circ \boldsymbol{A}_2}(\boldsymbol{\xi}_1 \circ \boldsymbol{\xi}_2, \boldsymbol{x}) \circ p^{\boldsymbol{m}_1 + \boldsymbol{m}_2} \pmod q).$$

On the other hand, the encryption of the sum of two messages $\boldsymbol{m}_1, \boldsymbol{m}_2$ is:

$$Enc(sk, \boldsymbol{m}_1 + \boldsymbol{m}_2) = (\boldsymbol{A}, f_{\boldsymbol{A}}(\boldsymbol{\xi}, \boldsymbol{x}) \circ p^{\boldsymbol{m}_1 + \boldsymbol{m}_2} \pmod q).$$

Here, the matrices $\boldsymbol{A}$ and $\boldsymbol{A}_1 \circ \boldsymbol{A}_2$ follow the same distribution, which does not effect decryptability. The two encryption paths differ primarily in the number of coefficient factors in each row: one is $\xi_{1,i} \cdot \xi_{2,i}$ (two factors) and the other is ξ_i (a single factor). Both cases allow decryption because $d(\xi_1 \cdot \xi_2 \cdot p^2) < d(q)$ for all $\xi_1, \xi_2 \in L$. □

Note that after each homomorphic computation $Enc(sk, \boldsymbol{m}_1) \cdot Enc(sk, \boldsymbol{m}_2)$, the number of coefficients ξ_i increases by 1. To allow k homomorphic computations, we require $d(q) > d(\prod_{i=1}^{k} \xi_i \cdot p^2)$ for any coefficients $\xi_i \in L$. An open problem is how to achieve an arbitrary depth of homomorphic computations from unique properties of concrete Euclidean domains. For example, under the same degree $d(q)$, working over polynomials can allow more operations than working over integers. See Section 4.2 for more explanations.

Another open problem is how to make the encryption scheme a public-key encryption scheme. One way to achieve this is by using trapdoors (see Section 5.3). The idea is that the receiver firstly uses the trapdoor (which is the secret key) to recover $\boldsymbol{x}$, then uses $\boldsymbol{x}$ to recover $\boldsymbol{m}$.

5.2. **Hash function.** Let a be an element of $\mathcal{R}$. Denote $bin(a)$ as the binary representation of a. Let $\boldsymbol{v}$ be a binary vector. Denote $int(\boldsymbol{v})$ as the integer whose binary representation is $\boldsymbol{v}$. Denote ℓ_i as the i-th element of L. Denote $l := \lceil \log_2(|L|) \rceil$ and $r := \lceil \log_2(|\mathcal{R}_q^\times|) \rceil$.

Define the MESS hash function family as

$$\mathcal{H} : (\mathcal{R}_q^\times)^{m \times n} \times \{0,1\}^{n+ml} \to \{0,1\}^r.$$

A function $H_{\boldsymbol{A}}$ in $\mathcal{H}$ is of the form

$$H_{\boldsymbol{A}}(\boldsymbol{z}, \boldsymbol{x}) = bin(\boldsymbol{\xi} \circ \boldsymbol{A}^{\boldsymbol{x}} \pmod q),$$

where $\boldsymbol{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m)$ and $\boldsymbol{e} = (e_1, \ldots, e_r)$ with $\xi_i = \ell_{int(\boldsymbol{z}_i)}$.

The MESS hash family achieves the following three properties of hash functions.

Definition 5.3 ([42, 21])**.** Let $\mathcal{H} : \mathcal{K} \times \mathcal{M} \to \mathcal{Y}$ be a hash function family where $|\mathcal{M}| = 2^r$. Then $\mathcal{H}$ is said to be

(1) preimage-resistant: if for every probabilistic polynomial time algorithm $\mathcal{A}$ there exists a negligible function $\mu(r)$ such that

$$\Pr[K \leftarrow \mathcal{K}, a \leftarrow \mathcal{M}, h \leftarrow H_K(a), a' \leftarrow \mathcal{A}(H_K, h) : H_K(a') = h] \leq \mu(r);$$

(2) *collision-resistant*: if for every probabilistic polynomial time algorithm $\mathcal{A}$ there exists a negligible function $\mu(r)$ such that

$$\Pr[K \leftarrow \mathcal{K}, (a, a') \leftarrow \mathcal{A}(K) : (a \neq a') \wedge (H_K(a) = H_K(a'))] \leq \mu(r);$$

(3) *collision-free*: if there exists a negligible function $\mu(r)$ such that

$$\Pr_{K \leftarrow \mathcal{K}}[\exists a, b \in \mathcal{M} : a \neq b, H_K(a) = H_K(b)] \leq \mu(r).$$

It is clear that collision-freeness implies collision-resistance. It can be shown that for a random function, if $|\mathcal{M}| = 2^r$ then $|\mathcal{Y}| = 2^{3r}$ is enough to ensure that the hash function family is collision-free [21, Section 2.4].

Theorem 5.4. *If computing $\ell_{int(\boldsymbol{z}_i)}$ is efficient and MESS is hard, then the MESS hash function family is preimage-resistant. If the MESS further has a unique solution $(\boldsymbol{\xi}, \boldsymbol{x})$, then the MESS hash function family is collision-resistant and collision-free.*

Proof. For the first part of the theorem, if computing $\ell_{int(\boldsymbol{z}_i)}$ is efficient, then computing $\boldsymbol{\xi}$ from $\boldsymbol{z}$ is efficient. Again, if MESS is hard, then it is hard to find $(\boldsymbol{\xi}, \boldsymbol{x})$ from $\boldsymbol{\xi} \circ \boldsymbol{A}^{\boldsymbol{x}} \pmod q$. Therefore it must be hard to find $(\boldsymbol{z}, \boldsymbol{x})$ from $bin(\boldsymbol{\xi} \circ \boldsymbol{A}^{\boldsymbol{x}} \pmod q)$ since finding $(\boldsymbol{z}, \boldsymbol{x})$ immediately yields $(\boldsymbol{\xi}, \boldsymbol{x})$.

For the second part of the theorem, collision-resistance is implied by collision-free, and collision-free is clear since $\boldsymbol{z}$ and $\boldsymbol{\xi}$ are one-to-one corresponding. □

By Theorem 3.6 and 4.3, as well as the easiness of finding an efficient-Gaussian-predictable pair as illustrated by the example in Lemma 3.8, it is not hard to find a qualified Euclidean domain where the MESS variant has a unique solution, and thus it is not hard to find a collision-free MESS hash function family.

5.3. **Trapdoor function.** Let L_1 and L_2 be two disjoint sets of primes of $\mathcal{R}$. Let q be a prime of $\mathcal{R}$ of degree greater than the degree of the product of any n primes in L_1 and any $n+1$ primes in L_2. Let $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_k, \boldsymbol{x} \in \{0,1\}^n$ be secrets. Let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ and $\boldsymbol{\xi} := (\boldsymbol{v}, v)$ be coefficient vectors, where each $\boldsymbol{u}$ is a vector of distinct primes sampled from L_1, and $\boldsymbol{\xi}$ is a vector of distinct primes sampled from L_2. Let $\boldsymbol{a}_i = \boldsymbol{u}_i \circ \boldsymbol{A}^{\boldsymbol{s}_i} \pmod q \in (\mathcal{R}_q^{\times})^n$ for $i = 1, \ldots, k$. Append $\boldsymbol{a}_i$ to $\boldsymbol{A}^{\circ -1}$ and denote $\bar{\boldsymbol{A}}_i := (\boldsymbol{A}^{\circ -1}, \boldsymbol{a}_i^{\top})^{\top}$.

Define the MESS trapdoor function family as:

$$\mathcal{F} = \{f_{\bar{\boldsymbol{A}}_i}(\boldsymbol{\xi}, \boldsymbol{x}) = \boldsymbol{\xi} \circ \bar{\boldsymbol{A}}_i^{\boldsymbol{x}} \pmod q\}_{i=1,\ldots,k}$$

with the set of trapdoors to be $T = \{(\boldsymbol{u}_i, \boldsymbol{s}_i)\}_{i=1,\ldots,k}$.

Theorem 5.5. *If MESS with $\mathbb{z} = \{0,1\}$ is hard then the function family $\mathcal{F}$ along with the set T is a trapdoor function family.*

Proof. We first show one-wayness. Note that $f_{\bar{\boldsymbol{A}}_i}(\boldsymbol{\xi}, \boldsymbol{x})$ is a trapdoored MESS with base $(\boldsymbol{A}^{\circ -1}, \boldsymbol{a}_i^{\top})^{\top}$. By Theorem 4.10, this trapdoored MESS is as hard as MESS. The one-wayness of $\mathcal{F}$ follows.

We then show trapdoorness. We want to show that $(\boldsymbol{u}_i, \boldsymbol{s}_i)$ is a trapdoor, namely it is efficient to find $(\boldsymbol{\xi}, \boldsymbol{x})$ from $f_{\bar{\boldsymbol{A}}_i}(\boldsymbol{\xi}, \boldsymbol{x})$ and $(\boldsymbol{u}_i, \boldsymbol{s}_i)$. To recover $\boldsymbol{x}$ with $(\boldsymbol{u}_i, \boldsymbol{s}_i)$, we simply compute

$$\begin{aligned} E &\equiv (v \cdot \boldsymbol{a}_i^{\boldsymbol{x}}) \cdot (\boldsymbol{v} \circ ((\boldsymbol{A}^{\circ -1})^{\top})^{\boldsymbol{x}})^{\boldsymbol{s}_i} \pmod q \\ &\equiv v \cdot \boldsymbol{u}^{\boldsymbol{x}} \cdot \boldsymbol{v}^{\boldsymbol{s}_i} \pmod q \\ &\equiv v \cdot \boldsymbol{u}^{\boldsymbol{x}} \cdot \boldsymbol{v}^{\boldsymbol{s}_i}. \end{aligned}$$

Recall that the degree of q is greater than the degree of the product of n primes in L_1 and $n+1$ primes in L_2. Hence in the last step of the above equation, we can ignore "mod q" by the definition of qualified Euclidean domains, and we have that $E = v \cdot u_{i,1}^{x_1} \cdots u_{i,n}^{x_n} \cdot v_1^{s_{i,1}} \cdots v_n^{s_{i,n}}$. We can then recover $\boldsymbol{x}$ by testing which primes $u_{i,j}$ in $\boldsymbol{u}_i$ divide E. We have that $x_j = 1$ if and only if $u_{i,j} | E$. □

5.4. **Signature scheme.** We first construct an analogy of the Schnorr identification scheme by replacing the discrete logarithm problem by MESS, then use the Fiat-Shamir transformation to turn it into a signature scheme.

An identification scheme involves two parties, the prover and the verifier. Both of them have access to the same public key. The prover wants to convince the verifier that she owns the corresponding private key of the public key. They interact as the following. The prover first sends a commitment to the verifier; the verifier then sends a challenge to the prover; the prover then sends a response to the verifier; in the end the verifier accepts or rejects the prover's claim about her possession of the secret key. The MESS identification scheme is as follows.

Let $m, n \in \mathbb{N}$ be positive integers with $m \geq 2n$, and $q \in \mathcal{R}$ be a prime. Let $\boldsymbol{A} \in (\mathcal{R}_q^\times)^{m \times n}$ be a uniform base matrix. The prover's private key is $(\boldsymbol{u}, \boldsymbol{s}) \leftarrow L^m \times \mathbb{Z}_p^n$. Her public key is $(\boldsymbol{A}, \boldsymbol{b} := \boldsymbol{u} \circ \boldsymbol{A}^{\boldsymbol{s}} \pmod q)$.

- The prover samples $(\boldsymbol{\xi}, \boldsymbol{x}) \leftarrow L^m \times \mathbb{Z}_p^n$; computes $\boldsymbol{a} = \boldsymbol{\xi} \circ \boldsymbol{A}^{\boldsymbol{x}} \pmod q$; and sends $\boldsymbol{a}$ to the verifier as the commitment;
- The verifier samples $c \leftarrow \mathbb{Z}_p$ and sends it to the prover as the challenge;
- The prover computes $\boldsymbol{y} = \boldsymbol{x} - c\boldsymbol{s} \pmod p$ and $\boldsymbol{v} = \boldsymbol{\xi} \circ \boldsymbol{u}^{\circ -c} \pmod q$, and sends $(\boldsymbol{y}, \boldsymbol{v})$ to the verifier as the response;
- The verifier computes $\boldsymbol{a}' = \boldsymbol{v} \circ \boldsymbol{b}^{\circ c} \circ \boldsymbol{A}^{\boldsymbol{y}} \pmod q$; and accepts if $\boldsymbol{a}' = \boldsymbol{a}$ or rejects if $\boldsymbol{a}' \neq \boldsymbol{a}$.

The correctness is the following.

Theorem 5.6. *If every party in the scheme is honest then $\boldsymbol{a}' = \boldsymbol{a}$.*

Proof. It is clear that

$$\begin{aligned}
\boldsymbol{a}' &= \boldsymbol{v} \circ \boldsymbol{b}^{\circ c} \circ \boldsymbol{A}^{\boldsymbol{y}} \mod q \\
&\equiv (\boldsymbol{\xi} \circ \boldsymbol{u}^{\circ -c}) \circ (\boldsymbol{u} \circ \boldsymbol{A}^{\boldsymbol{s}})^{\circ c} \circ \boldsymbol{A}^{\boldsymbol{y}} \mod q \\
&\equiv (\boldsymbol{\xi} \circ \boldsymbol{u}^{\circ -c}) \circ \boldsymbol{u}^{\circ c} \circ \boldsymbol{A}^{\boldsymbol{y} + c\boldsymbol{s}} \mod q \\
&\equiv \boldsymbol{\xi} \circ \boldsymbol{A}^{\boldsymbol{x}} \mod q \\
&= \boldsymbol{a}.
\end{aligned}$$

□

The security of the identification scheme reduces to the hardness of the concrete MESS by a similar proof logic as in [30, Section 13.5.2], with a twist of the underlying computational problem. Specifically, we assume that the adversary is given the public key pk and can eavesdrop previous executions of the protocol with respect to the same private key sk. Let o_{sk} be the oracle that each time invokes a fresh execution of the protocol and returns the full transcript $(\boldsymbol{a}, c, \boldsymbol{y})$ of the execution. Then what we assume is that the adversary is given pk and o_{sk}.

Definition 5.7. An identification or signature scheme is said to be secure (against impersonation) if for all probabilistic polynomial time adversaries $\mathcal{A}$, there is a negligible function μ such that the probability that $\mathcal{A}$ (given the public key pk and can eavesdrop previous executions of the protocol with respect to the same private key sk) convinces the verifier is $\leq \mu(n)$.

Theorem 5.8. *If the MESS variant is hard and it has a unique solution with overwhelming probability, then the MESS identification scheme is secure against impersonation.*

Proof. Let $\mathcal{A}$ be any probabilistic polynomial time adversary, which is given pk and o_{sk}. Define a MESS solver $\mathcal{B}$ as follows. $\mathcal{B}$ takes as input a MESS instance $(\boldsymbol{A}, \boldsymbol{b})$. It runs $\mathcal{A}(pk) = \mathcal{A}(\boldsymbol{A}, \boldsymbol{b})$. When $\mathcal{A}$ outputs $\boldsymbol{a}$, $\mathcal{B}$ chooses a uniform $c_1 \leftarrow \mathbb{Z}_p$ as the challenge and gives it to $\mathcal{A}$; $\mathcal{A}$ responses with $(\boldsymbol{y}_1, \boldsymbol{v}_1)$. $\mathcal{B}$ then runs $\mathcal{A}(pk)$ a second time with c_1 replaced by an independent $c_2 \leftarrow \mathbb{Z}_p$; $\mathcal{A}$ responses with $(\boldsymbol{y}_2, \boldsymbol{v}_2)$. If

$$\boldsymbol{v}_1 \circ \boldsymbol{b}^{\circ c_1} \circ \boldsymbol{A}^{\boldsymbol{y}_1} = \boldsymbol{a}$$

and

$$\boldsymbol{v}_2 \circ \boldsymbol{b}^{\circ c_2} \circ \boldsymbol{A}^{\boldsymbol{y}_2} = \boldsymbol{a}$$

and that

$$c_1 \neq c_2$$

then $\mathcal{B}$ outputs $(\boldsymbol{y}_1 - \boldsymbol{y}_2)/(c_1 - c_2) \pmod p$.

Let ω be the randomness during the execution. Define $V(\omega, c) = 1$ if and only if the problem $(\boldsymbol{A}, \boldsymbol{b})$ has a unique solution and $\mathcal{A}$ correctly responds to challenge c when randomness ω is used in the rest of the execution; define $V'(\omega, c) = 1$ if and only if the problem $(\boldsymbol{A}, \boldsymbol{b})$ has nonunique solutions and $\mathcal{A}$ correctly responds to challenge c when randomness ω is used in the rest of the execution. For any fixed ω, define $\delta_\omega := \Pr_c[V(\omega, c) = 1]$ and $\delta'_\omega := \Pr_c[V'(\omega, c) = 1]$. With ω fixed, δ_ω and δ'_ω are the probabilities over c that $\mathcal{A}$ responds correctly under the two situations of unique and nonunique solutions of $(\boldsymbol{A}, \boldsymbol{b})$ respectively.

Denote $\delta(n)$ as the probability that $\mathcal{A}$ succeeds when $(\boldsymbol{A}, \boldsymbol{b})$ has a unique solution. We have

$$\delta(n) = \Pr_{\omega,c}[V(\omega, c) = 1] = \sum_\omega \Pr[\omega] \cdot \delta_\omega.$$

Denote $\delta'(n)$ as the probability that $\mathcal{A}$ succeeds when $(\boldsymbol{A}, \boldsymbol{b})$ has nonunique solutions. We have

$$\delta'(n) = \Pr_{\omega,c}[V'(\omega, c) = 1] = \sum_\omega \Pr[\omega] \cdot \delta'_\omega.$$

Denote $\bar{\delta}(n)$ as the probability that $\mathcal{A}$ succeeds. We have

$$\bar{\delta}(n) = \mathrm{P} \cdot \delta(n) + (1 - \mathrm{P}) \cdot \delta'(n).$$

In the following we show that this probability is negligible.

Denote P as the probability that $(\boldsymbol{A}, \boldsymbol{b})$ has a unique solution. By assumption, P is overwhelming.

Denote $\tilde{\delta}(n)$ as the probability that $\mathcal{B}$ succeeds. Note that $\mathcal{B}$ successfully solves $(\boldsymbol{A}, \boldsymbol{b})$ if (1) $(\boldsymbol{A}, \boldsymbol{b})$ has a unique solution and $\mathcal{A}$ succeeds twice with $c_1 \neq c_2$; or (2) $(\boldsymbol{A}, \boldsymbol{b})$ has nonunique solutions and $\mathcal{A}$ succeeds with twice with $c_1 \neq c_2$ and that the two times that $\mathcal{A}$ succeeds are with respect to the same solution $\boldsymbol{x}_1 = \boldsymbol{x}_2$ to $(\boldsymbol{A}, \boldsymbol{b})$. Hence

$$\begin{aligned}
\tilde{\delta}(n) =& \mathrm{P} \cdot \Pr_{\omega,c_1,c_2}[V(\omega, c_1) \wedge V(\omega, c_2) \wedge c_1 \neq c_2] \\
&+ (1 - \mathrm{P}) \cdot \Pr_{\omega,c_1,c_2}[V'(\omega, c_1) \wedge V'(\omega, c_2) \wedge c_1 \neq c_2 \wedge \boldsymbol{x}_1 = \boldsymbol{x}_2] \\
\geq& \mathrm{P} \cdot \Pr_{\omega,c_1,c_2}[V(\omega, c_1) \wedge V(\omega, c_2) \wedge c_1 \neq c_2] \\
\geq& \mathrm{P} \cdot \left(\Pr_{\omega,c_1,c_2}[V(\omega, c_1) \wedge V(\omega, c_2)] - \Pr_{\omega,c_1,c_2}[c_1 = c_2] \right) \\
=& \mathrm{P} \cdot \left(\sum_\omega \Pr[\omega] \cdot (\delta_\omega)^2 - 1/p \right)
\end{aligned}$$

$$\begin{aligned}&\geq \mathrm{P} \cdot \left(\left(\sum_{\omega} \Pr[\omega] \cdot \delta_{\omega} \right)^2 - 1/p \right) \\ &= \mathrm{P} \cdot \left(\delta(n)^2 - 1/p \right),\end{aligned}$$

where the second-to-last step uses Jensen's inequality.

Now by the assumption that the MESS $(\boldsymbol{A}, \boldsymbol{b})$ is hard, $\mathcal{B}$ succeeds with negligible probability. That is, $\tilde{\delta}(n)$ is negligible. Also note that P is overwhelming and $1/p$ is negligible. Hence $\delta(n)$ is negligible. Again, $1 - \mathrm{P}$ is negligible since P is overwhelming. Therefore $\bar{\delta}(n) = \mathrm{P} \cdot \delta(n) + (1 - \mathrm{P}) \cdot \delta'(n)$ is negligible. That is, $\mathcal{A}$ succeeds with negligible probability. Hence the scheme is secure. □

Now we present the MESS signature scheme as follows.

$\mathsf{KeyGen}(n, m, q)$:
- Sample $\boldsymbol{A} \leftarrow (\mathcal{R}_q^{\times})^{m \times n}$;
- Sample $(\boldsymbol{u}, \boldsymbol{s}) \leftarrow L^m \times \mathbb{Z}_p^n$;
- Compute $\boldsymbol{b} = \boldsymbol{u} \circ \boldsymbol{A}^{\boldsymbol{s}} \pmod{q}$;
- Output (sk, pk) with $sk := (\boldsymbol{u}, \boldsymbol{s})$, $pk := (\boldsymbol{A}, \boldsymbol{b})$.

$\mathsf{Sign}(sk, \boldsymbol{m})$:
- Sample $(\boldsymbol{\xi}, \boldsymbol{x}) \leftarrow L^m \times \mathbb{Z}_p^n$ and compute $\boldsymbol{a} = \boldsymbol{\xi} \circ \boldsymbol{A}^{\boldsymbol{x}} \pmod{q}$;
- Compute $c = H(\boldsymbol{a}, \boldsymbol{m})$, where H is a cryptographic hash function;
- Compute $\boldsymbol{y} = \boldsymbol{x} - c\boldsymbol{s} \pmod{p}$ and $v = \boldsymbol{\xi} \circ \boldsymbol{u}^{\circ -c} \pmod{q}$;
- Output $(\boldsymbol{y}, \boldsymbol{v}, c)$ as the signature.

$\mathsf{Verify}(pk, \boldsymbol{m}, \boldsymbol{y}, \boldsymbol{v}, c)$:
- Compute $\boldsymbol{a}' = \boldsymbol{v} \circ \boldsymbol{b}^{\circ c} \circ \boldsymbol{A}^{\boldsymbol{y}} \pmod{q}$;
- Compute $c' = H(\boldsymbol{a}', \boldsymbol{m})$;
- Accept if $c' = c$ or rejects if $c' \neq c$.

The correctness of the MESS signature scheme is as follows.

Theorem 5.9. *If every party in the identification scheme is honest then $c' = c$ in the MESS signature scheme.*

Proof. By a similar argument to the proof of Theorem 5.6, we have $\boldsymbol{a}' = \boldsymbol{a}$. Then $c' = H(\boldsymbol{a}', \boldsymbol{m}) = H(\boldsymbol{a}, \boldsymbol{m}) = c$. □

The security is from Theorem 5.8 and the following well-known theorem.

Theorem 5.10 ([30, p.478, Theorem 13.10])**.** *If an identification scheme is secure against impersonation and the hash function is modeled as a random oracle, then the signature scheme that results by applying the Fiat-Shamir transform is secure against impersonation.*

The security of the MESS signature scheme is as follows.

Theorem 5.11. *If MESS is hard and has a unique solution with overwhelming probability and that the hash function H is modeled as a random oracle, then the MESS signature scheme is secure against impersonation.*

Proof. Immediate from Theorem 5.8 and 5.10. □

5.5. **Multilinear map with coefficients.** Cryptographic multilinear map was envisioned by Boneh and Silverberg [10]. It has enormous applications in cryptography [48]. Several candidates [24, 33, 15, 14] have been proposed but followed by cryptanalysis [13, 12]. We define multilinear map with coefficients, which can hopefully achieve many goals that a traditional multilinear map achieves.

To explain our ideas, we generalize our notations in Section 2.2 to handle hypermatrices and define related operations.

Hypermatrices. A *hypermatrix* is a generalization of a matrix to a multidimensional array. The *order* or *dimension* of a hypermatrix is the number of indices required to specify its elements. In particular, an order-1 hypermatrix is a vector, an order-2 hypermatrix is a matrix. We denote hypermatrices by bold uppercase letters such as $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, etc. When they are clearly vectors, we denote them by bold lowercase letters such as $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$, etc. We denote elements of a hypermatrix by light lowercase letters such as a, b, c, etc. We only consider hypermatrices over a Euclidean domain.

Common operations. Let $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n)$ be two vectors. Denote $\cdot$ as inner product between two vectors or normal multiplication between two elements, i.e., $\boldsymbol{a} \cdot \boldsymbol{b} := \sum_{i=1}^{n} a_i \cdot b_i$. Let $\boldsymbol{A} = (a_{\alpha_1,\ldots,\alpha_d})$ and $\boldsymbol{B} = (b_{\alpha_1,\ldots,\alpha_d})$ be two order-d hypermatrices. Denote $\circ$ as Hadamard product (or entry-wise product), i.e., $\boldsymbol{A} \circ \boldsymbol{B} := (a_{\alpha_1,\ldots,\alpha_d} \cdot b_{\alpha_1,\ldots,\alpha_d})$. Let x be an element. Denote $x\boldsymbol{A} := (x a_{\alpha_1,\ldots,\alpha_d})$, $\boldsymbol{A}^{\circ x} := (a^x_{\alpha_1,\ldots,\alpha_d})$ and $x^{\boldsymbol{A}} := (x^{a_{\alpha_1,\ldots,\alpha_d}})$. Namely applying x to $\boldsymbol{A}$ entry-wise. Denote $\boldsymbol{A}|x := (\text{“}a_{\alpha_1,\ldots,\alpha_d}|x\text{”})$ to be the binary hypermatrix $\boldsymbol{B} \in \{0,1\}^{\alpha_1 \times \cdots \times \alpha_d}$ such that each entry $b_{\alpha_1,\ldots,\alpha_d} = 1$ if and only if $a_{\alpha_1,\ldots,\alpha_d}|x$.

Actions. Let $m_1, \ldots, m_d \geq n$ be positive integers. Let $\boldsymbol{A} = (a_{\alpha_1,\ldots,\alpha_d})$ be an $m_1 \times \cdots \times m_d$ hypermatrix, where $1 \leq \alpha_i \leq m_i$. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d \in \{0,1\}^n$ be d binary vectors.[8] Define an *action* on $\boldsymbol{A}$ to be $\boldsymbol{x}_i : \boldsymbol{A} \mapsto \boldsymbol{A}^{\boldsymbol{x}_i}$ such that $\boldsymbol{A}^{\boldsymbol{x}_i} = \boldsymbol{A}$ if $\boldsymbol{x}_i = \boldsymbol{0}$ and $\boldsymbol{A}^{\boldsymbol{x}_i} = (a_{\alpha_1,\ldots,\alpha_{i-1},\alpha_{i+1},\ldots,\alpha_d} = \prod_{j=1}^{n} a^{x_{i,j}}_{\alpha_1,\ldots,\alpha_{i-1},\alpha_j,\alpha_{i+1},\ldots,\alpha_d})$ if $\boldsymbol{x}_i \neq \boldsymbol{0}$. Namely if $\boldsymbol{x}_i = \boldsymbol{0}$ then $\boldsymbol{x}_i$ preserves $\boldsymbol{A}$ as it is as an order-d hypermatrix and if $\boldsymbol{x}_i \neq \boldsymbol{0}$ then it shrinks the i-th dimension of $\boldsymbol{A}$ by multiplying together the elements along the i-th direction that are indicated by the 1's in $\boldsymbol{x}_i$. Note that along the i-th direction there are m_i elements and $\boldsymbol{x}_i$ only applies on the first n elements and ignores the rest. When we use $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ to denote actions without specifying dimensions, we assume that they apply to the first, second, and third dimensions respectively. For example, let $\boldsymbol{a} = (a_1, \ldots, a_m)$ be an order-1 hypermatrix (i.e. a vector) over $\mathbb{Z}$ and $\boldsymbol{x} = (x_1, \ldots, x_n)$ be a nonzero vector over $\{0,1\}$, where $n \leq m$. Then applying $\boldsymbol{x}$ on $\boldsymbol{a}$ shrinks $\boldsymbol{a}$ into an order-0 hypermatrix (i.e. a number) $\boldsymbol{a}^{\boldsymbol{x}} = \prod_{i=1}^{n} a_i^{x_i}$.

Action compositions. Define the composition of two actions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ to be $\boldsymbol{x}_1\boldsymbol{x}_2 : \boldsymbol{A} \mapsto \boldsymbol{A}^{\boldsymbol{x}_1\boldsymbol{x}_2} = (\boldsymbol{A}^{\boldsymbol{x}_1})^{\boldsymbol{x}_2}$. This is just to apply $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ on $\boldsymbol{A}$ successively. Clearly $\boldsymbol{A}^{\boldsymbol{x}_1\boldsymbol{x}_2} = \boldsymbol{A}^{\boldsymbol{x}_2\boldsymbol{x}_1}$ (commutativity) and $\boldsymbol{A}^{\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3} = \boldsymbol{A}^{(\boldsymbol{x}_1\boldsymbol{x}_2)\boldsymbol{x}_3} = \boldsymbol{A}^{\boldsymbol{x}_1(\boldsymbol{x}_2\boldsymbol{x}_3)}$ (associativity). If we apply d nonzero actions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$ on an order-d hypermatrix $\boldsymbol{A}$ then $\boldsymbol{A}$ collapses to an order-0 hypermatrix $\boldsymbol{A}^{\boldsymbol{x}_1\cdots\boldsymbol{x}_d}$ which is just an element of the Euclidean domain $\mathcal{R}$. Hence at most d nonzero actions can be applied on an order-d hypermatrix. In that case, the resulting element is the product of elements of $\boldsymbol{A}$ that are simultaneously chosen by $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$ from d different directions – Imagine slicing a cube of cheese from three directions: top, right, and front, each with three slices, yielding a total of 27 intersecting points within the cube. We

[8] It is convenient to extend the definition to handle more general integral vectors in $\mathbb{Z}^n$.

therefore call $\boldsymbol{A}^{\boldsymbol{x}_1\cdots\boldsymbol{x}_d}$ the *product of intersection* of $\boldsymbol{A}$ with respect to $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_d$. For example, let $\boldsymbol{A}$ be an $m_1\times m_2$ matrix over $\mathbb{Z}$, and $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ be two nonzero vectors over $\{0,1\}$, where $m_1,m_2>n$. Then $\boldsymbol{x}_1$ shrinks $\boldsymbol{A}$ to a vector $\boldsymbol{A}^{\boldsymbol{x}_1}=(a_1,\ldots,a_{m_2})$ where $a_j=\prod_{i=1}^n a_{i,j}^{x_{1,i}}$; and $\boldsymbol{x}_2$ further shrinks $\boldsymbol{A}^{\boldsymbol{x}_1}$ to a number $\boldsymbol{A}^{\boldsymbol{x}_1\boldsymbol{x}_2}=\prod_{j=1}^n a_j^{x_{2,j}}$ which is the product of elements of $\boldsymbol{A}$ that are in the intersection of the rows and columns chosen by $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ respectively.

For more general vectors $\boldsymbol{x}_i\in\mathbb{Z}^n$, the definitions of action and action composition are generalized in the natural sense. Namely the binary exponents $x_{i,j}\in\{0,1\}$ in $a_{\alpha_1,\ldots,\alpha_{i-1},\alpha_j,\alpha_{i+1},\ldots,\alpha_d}^{x_{i,j}}$ are now allowed to be any integers.

Coefficient multilinear map. Now we define our new type of multilinear maps.

Definition 5.12. A map $\varphi:H_1^d\to H_2$ is a coefficient d-linear map if it satisfies the following properties:

(1) H_1 and H_2 are two sets of $(n+d)\times\cdots\times(n+d)$ hypermatrices of order $1\le k\le d$;
(2) there exists a set L such that if $\boldsymbol{G}\in H_1$, $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\in\{0,1\}^n$, and $\boldsymbol{C}_1,\ldots,\boldsymbol{C}_d\in L^{(n+d)\times\cdots\times(n+d)}$ ($\boldsymbol{E}_i$ is of order-d if $\boldsymbol{x}_i=\boldsymbol{0}$ and of order-$(d-1)$ if $\boldsymbol{x}_i\neq\boldsymbol{0}$), then
$$\varphi(\boldsymbol{C}_1\circ\boldsymbol{G}^{\boldsymbol{x}_1},\ldots,\boldsymbol{C}_d\circ\boldsymbol{G}^{\boldsymbol{x}_d})=\varphi(\boldsymbol{G},\ldots,\boldsymbol{G})^{\boldsymbol{x}_1\cdots\boldsymbol{x}_d};$$
(3) the map φ is non-degenerate in the sense that $\varphi(\boldsymbol{G},\ldots,\boldsymbol{G})=\boldsymbol{G}$, where $\boldsymbol{G}$ is called the base of the map.

Our construction is based on MESS with $|L|$ super-polynomial in n and $d(q)\gtrapprox d_0^{2n+1}$, where $d_0:=\max(d(L))$.

Construction from MESS. We first construct a base $\boldsymbol{G}\in\mathcal{R}_q^{(n+d)\times\cdots\times(n+d)}$ for our map. Let $\boldsymbol{A}=\{a_{\alpha_1,\ldots,\alpha_d}\}$ be an $n\times\cdots\times n$ hypermatrix over $\mathcal{R}_q^\times$. Let $\boldsymbol{A}_1,\ldots,\boldsymbol{A}_d$ be d $n\times n$ matrices in $\boldsymbol{A}$ such that their rows are parallel to the directions $\alpha_1,\ldots,\alpha_d$ respectively. Namely for every row of $\boldsymbol{A}_i$ the i-th indexes α_i of the elements of the row ascend from 1 to n.

Let $\boldsymbol{s}_1,\ldots,\boldsymbol{s}_d\in\{0,1\}^n$ be d binary vectors and $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_d\in L^n$ be d coefficient vectors. Let $\bar{\boldsymbol{a}}_1,\ldots,\bar{\boldsymbol{a}}_d$ be the result vectors with respect to $\boldsymbol{A}_1,\ldots,\boldsymbol{A}_d$ and $(\boldsymbol{u}_1,\boldsymbol{s}_1),\ldots,(\boldsymbol{u}_d,\boldsymbol{s}_d)$, i.e., $\bar{\boldsymbol{a}}_i=\boldsymbol{u}_i\circ\boldsymbol{A}_i^{\boldsymbol{s}_i}\pmod q$ for $i=1,\ldots,d$. Each pair $(\boldsymbol{u}_i,\boldsymbol{s}_i)$ is a *trapdoor* and each vector $\bar{\boldsymbol{a}}_i$ is called a *trapdoor result sequence*.

Let $\bar{\boldsymbol{A}}$ be the $d\times n$ matrix whose rows are the d trapdoor result sequences $\bar{\boldsymbol{a}}_1,\ldots,\bar{\boldsymbol{a}}_d$. Let $\boldsymbol{A}_1^{\circ-1},\ldots,\boldsymbol{A}_d^{\circ-1}$ be $\boldsymbol{A}_1,\ldots,\boldsymbol{A}_d$ with inversed entries. Append $\bar{\boldsymbol{A}}$ to $\boldsymbol{A}_1^{\circ-1},\ldots,\boldsymbol{A}_d^{\circ-1}$ to get d extended matrices $(\boldsymbol{A}_1^{\circ-1},\bar{\boldsymbol{A}})^\top,\ldots,(\boldsymbol{A}_d^{\circ-1},\bar{\boldsymbol{A}})^\top$.

Within $\boldsymbol{A}$, replace $\boldsymbol{A}_1,\ldots,\boldsymbol{A}_d$ by $(\boldsymbol{A}_1^{\circ-1},\bar{\boldsymbol{A}})^\top,\ldots,(\boldsymbol{A}_d^{\circ-1},\bar{\boldsymbol{A}})^\top$ respectively. This is to employ trapdoor instances to ensure that our coefficient multilinear map is efficiently computable. Now $\boldsymbol{A}$ becomes an object $\boldsymbol{O}$ where most vectors are of dimension[9] n but some are extended to dimension $n+d$. Complete $\boldsymbol{O}$ by extending all vectors to $(n+d)$-dimension by filling with 1's (or any element in $\mathcal{R}_q^\times$ which does not matter, as the actions $\boldsymbol{x}_i$ will only apply on the first n entries and ignore the rest), resulting in an $(n+d)\times\cdots\times(n+d)$ hypermatrix $\boldsymbol{G}$.

The base of our map is $\boldsymbol{G}$. In particular, if $\boldsymbol{G}$ is computed from random $\boldsymbol{u}_i\leftarrow L^n$, $\boldsymbol{A}\leftarrow(\mathcal{R}_q^\times)^{n\times\cdots\times n}$, and $\boldsymbol{s}_i\leftarrow\{0,1\}^n$, we call it an *average base*.

Let $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_d\in\{0,1\}^n$ be d binary vectors. Let $\boldsymbol{C}_1,\ldots,\boldsymbol{C}_d$ be d hypermatrices over L, where the order of $\boldsymbol{C}_i$ is d if $\boldsymbol{x}_i=\boldsymbol{0}$ and it is $d-1$ if $\boldsymbol{x}_i\neq\boldsymbol{0}$. Let H_1 be

[9]Here, dimension means the number of entries of a vector and not the order of a vector as a hypermatrix.

the set of $(n+d)\times\cdots\times(n+d)$ order-k hypermatrices over $\mathcal{R}_q^\times$ for all integers $d-1\le k\le d$. Let H_2 be the set of $(n+d)\times\cdots\times(n+d)$ order-k hypermatrices over $\mathcal{R}_q^\times$ for all integers $1\le k\le d$. Our coefficient multilinear map is the following.

Definition 5.13. The *MESS coefficient d-linear map* is $\varphi: H_1^d\to H_2$ such that

$$\varphi(\boldsymbol{C}_1\circ\boldsymbol{G}^{\boldsymbol{x}_1} \pmod q,\ldots,\boldsymbol{C}_d\circ\boldsymbol{G}^{\boldsymbol{x}_d} \pmod q)=\boldsymbol{G}^{\boldsymbol{x}_1\cdots\boldsymbol{x}_d} \pmod q. \tag{3}$$

Note that $\boldsymbol{C}_i\circ\boldsymbol{G}^{\boldsymbol{x}_i} \pmod q$ is a hypermatrix of order d if $\boldsymbol{x}=\boldsymbol{0}$ and it is of order $(d-1)$ if $\boldsymbol{x}_i\neq\boldsymbol{0}$. Also, $\boldsymbol{G}^{\boldsymbol{x}_1\cdots\boldsymbol{x}_d} \pmod q$ is a hypermatrix of order n_0 where n_0 is the number of i's such that $\boldsymbol{x}_i=\boldsymbol{0}$. Also notice that if $\boldsymbol{x}_i\neq\boldsymbol{0}$ then $\boldsymbol{C}_i\circ\boldsymbol{G}^{\boldsymbol{x}_i} \pmod q$ (in the average case) is in fact a set of n^{d-1} many MESS instances with respect to $\boldsymbol{x}_i$.

The following lemma says that the map can be computed efficiently if a trapdoor is given.

Lemma 5.14. *There exists a probabilistic polynomial time algorithm $\mathcal{A}$ that given $(\boldsymbol{u}_i,\boldsymbol{s}_i)$ for any $i\in\{1,\ldots,d\}$ computes φ with overwhelming probability.*

Proof. Suppose $\mathcal{A}$ has $(\boldsymbol{u}_i,\boldsymbol{s}_i)$ for some $i\in\{1,\ldots,d\}$. Let j be any of the other indexes. $\mathcal{A}$ only needs to use $\boldsymbol{v}:=\boldsymbol{f}_j\circ\boldsymbol{A}_i^{-\boldsymbol{x}_j} \pmod q$ and $v:=f_j\cdot\bar{\boldsymbol{a}}_i^{\boldsymbol{x}_j} \pmod q$ to compute the map, where $\boldsymbol{f}_j$ and f_j are the corresponding coefficient vector and coefficient of $\boldsymbol{v}$ and v respectively. $\mathcal{A}$ computes

$$\begin{aligned}
w &\equiv v\cdot\boldsymbol{v}^{\boldsymbol{s}_i} \pmod q\\
&\equiv v\cdot(\boldsymbol{f}_j\circ\boldsymbol{A}_i^{-\boldsymbol{x}_j})^{\boldsymbol{s}_i} \pmod q\\
&\equiv f_j\cdot(\boldsymbol{u}_i\circ\boldsymbol{A}_i^{\boldsymbol{s}_i})^{\boldsymbol{x}_j}\cdot(\boldsymbol{f}_j\circ\boldsymbol{A}_i^{-\boldsymbol{x}_j})^{\boldsymbol{s}_i} \pmod q\\
&\equiv f_j\cdot\boldsymbol{f}_j^{\boldsymbol{s}_i}\cdot\boldsymbol{u}_i^{\boldsymbol{x}_j} \pmod q\\
&= f_j\cdot\boldsymbol{f}_j^{\boldsymbol{s}_i}\cdot\boldsymbol{u}_i^{\boldsymbol{x}_j}.
\end{aligned}$$

Note that the last line is due to the degree $d(q)\gtrapprox d_0^{2n+1}=\max(d(L))^{2n+1}$, and that the Hamming weights of $\boldsymbol{s}_i$ and $\boldsymbol{x}_j$ are at most n. Then $\mathcal{A}$ can find $x_{i,k}$ efficiently by testing if $u_k|w$, for $k=1,\ldots,n$. Since all (but polynomially many) coefficients in use are sampled from an exponential size coefficient set L, they are all different with overwhelming probability. Hence we expect that no f is a u with overwhelming probability and that the decoding is correct with overwhelming probability. □

The security of the map is from the hardness of the following *computational and decisional coefficient multilinear problems*, denoted CCM and DCM, respectively.

Definition 5.15 (CCM)**.** Given $\boldsymbol{G},\boldsymbol{C}_1\circ\boldsymbol{G}^{\boldsymbol{x}_1} \pmod q,\ldots,\boldsymbol{C}_d\circ\boldsymbol{G}^{\boldsymbol{x}_d} \pmod q$, find $\boldsymbol{G}^{\boldsymbol{x}_1\cdots\boldsymbol{x}_d} \pmod q$.

Definition 5.16 (DCM)**.** Given distribution

$$D_0:=(\boldsymbol{C}_1\circ\boldsymbol{G}^{\boldsymbol{x}_1} \pmod q,\ldots,\boldsymbol{C}_d\circ\boldsymbol{G}^{\boldsymbol{x}_d} \pmod q,\boldsymbol{G}^{\boldsymbol{x}_1\cdots\boldsymbol{x}_d} \pmod q)$$

or

$$D_1:=(\boldsymbol{C}_1\circ\boldsymbol{G}^{\boldsymbol{x}_1} \pmod q,\ldots,\boldsymbol{C}_d\circ\boldsymbol{G}^{\boldsymbol{x}_d} \pmod q,r),$$

where $\boldsymbol{G}$ is a prefixed average base, $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_d\leftarrow\{0,1\}^n$, and $r\leftarrow\mathcal{R}_q^\times$, decide which distribution is given.

It is clear that CCM $\le$ DCM. We have the following theorem.

Theorem 5.17. *If the map given by Definition 5.13 is build upon a qualified Euclidean domain $\mathcal{R}$ where DCM is hard, then the map has the following properties.*

(1) quasi-multilinearity:

$$\varphi(\boldsymbol{C}_1 \circ \boldsymbol{G}^{\boldsymbol{x}_1} \pmod q, \ldots, \boldsymbol{C}_d \circ \boldsymbol{G}^{\boldsymbol{x}_d} \pmod q) = \varphi(\boldsymbol{G}, \ldots, \boldsymbol{G})^{\boldsymbol{x}_1 \cdots \boldsymbol{x}_d} \pmod q;$$

(2) non-degeneration:

$$\varphi(\boldsymbol{G}, \ldots, \boldsymbol{G}) = \boldsymbol{G}.$$

(3) efficiency: computing $\boldsymbol{G}^{\boldsymbol{x}_1 \cdots \boldsymbol{x}_d} \pmod q$ from $\boldsymbol{G}$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$ is efficient;

(4) trapdoorness: if $\boldsymbol{G}$ is an average base, $\boldsymbol{C}_i \leftarrow L^{n \times \cdots \times n}$ and $\boldsymbol{x}_i \leftarrow \{0,1\}^n$ then:

(4.1) inefficiency: computing $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$ from $\boldsymbol{G}, \boldsymbol{C}_1 \circ \boldsymbol{G}^{\boldsymbol{x}_1} \pmod q, \ldots, \boldsymbol{C}_d \circ \boldsymbol{G}^{\boldsymbol{x}_d} \pmod q$ is inefficient;

(4.2) efficiency: computing $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$ from $\boldsymbol{G}, \boldsymbol{C}_1 \circ \boldsymbol{G}^{\boldsymbol{x}_1} \pmod q, \ldots, \boldsymbol{C}_d \circ \boldsymbol{G}^{\boldsymbol{x}_d} \pmod q$ together with any of the trapdoors $(\boldsymbol{u}_1, \boldsymbol{s}_1), \ldots, (\boldsymbol{u}_d, \boldsymbol{s}_d)$ is efficient;

(5) randomness: if $\boldsymbol{G}$ is an average base, $\boldsymbol{x}_i \leftarrow \{0,1\}^n$ and $\boldsymbol{C}_i \leftarrow L^{n \times \cdots \times n}$ then the product $\boldsymbol{G}^{\boldsymbol{x}_1 \cdots \boldsymbol{x}_d} \pmod q$ is pseudorandom over $\mathcal{R}_q^\times$.

Proof. Quasi-multilinearity can be seen by plugging in non-degeneration; and non-degeneration is true because $\varphi(\boldsymbol{G}, \ldots, \boldsymbol{G}) = \varphi(\boldsymbol{G}^{\boldsymbol{0}}, \ldots, \boldsymbol{G}^{\boldsymbol{0}}) = \boldsymbol{G}^{\boldsymbol{0} \cdots \boldsymbol{0}} = \boldsymbol{G}^{\boldsymbol{0}} = \boldsymbol{G} \pmod q$. Efficiency is also clear since computing $\boldsymbol{G}^{\boldsymbol{x}_1 \cdots \boldsymbol{x}_d}$ only takes polynomially many multiplications in $\mathcal{R}_q^\times$ and each multiplication takes polynomial time by the definition of qualified Euclidean domains. The inefficiency aspect of trapdoorness is from the hardness of MESS. The efficiency aspect of trapdoorness is from Lemma 5.14. The randomness is from the hardness of DCM according to Theorem 4.10. □

At the end of this section, we show how the MESS coefficient multilinear map achieves two important applications of traditional multilinear maps [10]: multiparty key exchange and broadcast encryption.

Multiparty key exchange. A d-party ($d \geq 2$) key exchange scheme allows $d \geq 2$ parties to safely share a common secret. A general d-party key exchange scheme ($d \geq 2$) is immediate from the general coefficient d-multilinear map presented above. That is, d parties publish $\boldsymbol{\xi}_i \circ \boldsymbol{G}^{\boldsymbol{x}_i} \pmod q$ respectively and use their own secret keys $(\boldsymbol{u}_i, \boldsymbol{s}_i)$ to recover other parties' $\boldsymbol{x}_j$ and share $\boldsymbol{G}^{\boldsymbol{x}_1 \cdots \boldsymbol{x}_d} \pmod q$. The randomness of the shared secret is from the hardness of DCM.

Broadcast encryption. A broadcast encryption scheme allows Alice to secretly broadcast a message to a specified subset $S = \{B_i \mid i \in I, I \subset \{1, \ldots, d\}\}$ of Bobs $B_1, \ldots, B_d$. We achieve this scheme as follows. Alice embeds the public keys $\bar{\boldsymbol{a}}_i = \boldsymbol{u}_i \circ \boldsymbol{A}^{\boldsymbol{x}_i}$, $i \in I$ of the target Bobs into $\boldsymbol{G}$. Alice then uses her ephemeral key $(\boldsymbol{C}, \boldsymbol{x})$ to encrypt the message $\boldsymbol{m}$ to get a ciphertext c, and broadcasts c as well as $\boldsymbol{C} \circ \boldsymbol{G}^{\boldsymbol{x}}$. The target Bobs whose public keys are in $\boldsymbol{G}$ will be able to recover $\boldsymbol{x}$ and thus $\boldsymbol{C} = \boldsymbol{C} \circ \boldsymbol{G}^{\boldsymbol{x}} \circ \boldsymbol{G}^{-\boldsymbol{x}}$ and decrypt the ciphertext using $(\boldsymbol{C}, \boldsymbol{x})$. Meanwhile, the other Bobs are not able to recover $\boldsymbol{x}$ due to the hardness of MESS.

5.6. **Obfuscation.** For this application (i.e. obfuscation) and the next application (i.e. reusable fuzzy extractor), we will be using multivariate exponential equations

$$\xi \cdot \prod_{i=1}^{n} a_i^{x_i} = b \pmod{\mathfrak{p}}$$

with a_i primes. This special case of equations is of particular interest as it shows a clear difference from other computational problems. For these two applications, we only sketch the construction ideas and leave their full details and proofs to future work.

5.6.1. *Small superset and big subset functions.* This application uses a MESS variant where (1) $\mathbb{z} = \{0,1\}$; (2) the base vector $\boldsymbol{a}$ is a vector of primes a_i of $\mathcal{R}$; (3) $\{a_i\} \cap L = \varnothing$; and (4) the degree of the modulus q is greater than the degree of the product of any $t - |\boldsymbol{x}|$ primes in $\boldsymbol{a}$ and any prime in L, where t and $|\boldsymbol{x}|$ are parameters of the function family described below.

A small superset function (SSF) is a boolean function $f_{\boldsymbol{x},t} : \{0,1\}^n \to \{0,1\}; \boldsymbol{y} \mapsto f_{\boldsymbol{x},t}(\boldsymbol{y})$ with respect to a secret vector $\boldsymbol{x} \in \{0,1\}^n$ and a threshold $0 < t < n \in \mathbb{N}$ evaluates to 1 if and only if $\boldsymbol{y}$ is a "small superset" of $\boldsymbol{x}$, i.e., the 1's in $\boldsymbol{y}$ "cover" all the 1's in $\boldsymbol{x}$ and the number of 1's in $\boldsymbol{y}$ is not greater than t.

A big subset function (BSF) is a boolean function $f_{\boldsymbol{x},t} : \{0,1\}^n \to \{0,1\}; \boldsymbol{y} \mapsto f_{\boldsymbol{x},t}(\boldsymbol{y})$ with respect to a secret vector $\boldsymbol{x} \in \{0,1\}^n$ and a threshold $0 < t < n \in \mathbb{N}$ evaluates to 1 if and only if $\boldsymbol{y}$ is a "big subset" of $\boldsymbol{x}$, i.e., the 1's in $\boldsymbol{x}$ "cover" all the 1's in $\boldsymbol{y}$ and the number of 1's in $\boldsymbol{y}$ is greater than t.

Note that in a BSF, if we consider sets indicated by the 0's instead of 1's of $\boldsymbol{x}$ and $\boldsymbol{y}$, or equivalently, if we flip the bits of $\boldsymbol{x}$ and $\boldsymbol{y}$, we have an SSF [22] with threshold $n - t$. Therefore an obfuscator for SSF implies an obfuscator for BSF. Hence we only consider obfuscation of SSF. In particular, we only consider an SSF family with uniform $\boldsymbol{x}$ with fixed Hamming weight $|\boldsymbol{x}|$. It should be able to extend to uniform $\boldsymbol{x}$ without weight limitation.

The idea of this obfuscation is similar to [22] except that we employ a noise e to make a function "reobfuscatable". This extra property is not required by traditional obfuscation but it is meaningful in practice. Now back to the obfuscation, we simply compute a MESS instance $(\boldsymbol{a}, X = \boldsymbol{a}^{\boldsymbol{x}} \cdot e \pmod q)$ and publish it together with the obfuscated point function $O_P(f_{\boldsymbol{x}})$ as the obfuscated SSF, where $\boldsymbol{a}$ is a vector of random primes that are not in the noise set L, q is a prime modulus chosen to be of degree "just" greater than the degree of the product of any $t - |\boldsymbol{x}|$ primes in $\boldsymbol{a}$ and any prime in L. To evaluate, compute $E = X \cdot \boldsymbol{a}^{-\boldsymbol{y}} = \boldsymbol{a}^{\boldsymbol{x}-\boldsymbol{y}} \cdot e \pmod q$. Now if $\boldsymbol{y}$ is really a "small superset" of $\boldsymbol{x}$, then $\boldsymbol{a}^{\boldsymbol{x}-\boldsymbol{y}} \cdot e$ is smaller than q and "(mod q)" can be removed and that we have $E = \boldsymbol{a}^{\boldsymbol{x}-\boldsymbol{y}} \cdot e$. We can then recover $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{y}$ by testing divisibilities "$(a_i|E)$?" using the primes a_i in $\boldsymbol{a}$. Having $\boldsymbol{y}$ and $\boldsymbol{z}$, we can recover $\boldsymbol{x}$; and the noise e follows. In this case, $\boldsymbol{y}$ will be accepted by testing $O_P(f_{\boldsymbol{x}}(\boldsymbol{x})) = 1$. Otherwise, if $\boldsymbol{y}$ is not a "small superset" of $\boldsymbol{x}$, then the above decoding process will fail with high probability, meaning that we will encounter one or more of the following situations with high probability: (1) obtaining a non-binary $\boldsymbol{x}' := \boldsymbol{z} + \boldsymbol{y}$; (2) the prime factors of E are not all in $\boldsymbol{a}$ or L; or (3) there are more than one prime factor that are in L. In that case, $\boldsymbol{y}$ will be rejected. There is still a chance that $\boldsymbol{y}$ is a false positive, namely it is not an accepting input but it avoids all the above three conditions. These kinds of inputs can be identified by checking if $O_P(f_{\boldsymbol{x}})(\boldsymbol{x}') = 1$.

The advantage of this SSF obfuscator compared with [22] is that the same function $f_{\boldsymbol{x},t}$ can be obfuscated many times using this obfuscator, while in the case of [22], a function $f_{\boldsymbol{x},t}$ can only be obfuscated once, and that obfuscating it many times is not secure. Also, the conjunction obfuscation from our SSF obfuscation does not rely on any extra assumption apart from the hardness of MESS, while the conjunction obfuscation in [22, Appendix B] is based on an extra assumption which hypothesizes that the secret $\boldsymbol{x}$ is safe if it is only used twice.

5.6.2. *Conjunctions.* This application uses the same MESS variant as in Section 5.6.1. Conjunctions [8, 6] are also called pattern matching with wildcards. As suggested by [22, Appendix B], in order to obfuscate conjunctions, it is sufficient to obfuscate SSF and BSF. As mentioned earlier, obfuscation of BSF can be reduced to obfuscation of SSF. Therefore it is sufficient to obfuscate SSF, which is done in Section 5.6.1.

5.6.3. *Fuzzy Hamming distance matching.* This application uses a MESS variant where (1) $\mathbb{z} = \{0, 1\}$; (2) the base vectors $\boldsymbol{a}$ are vectors of distinct primes a_i of small norm such that their inverses b_i in $\mathcal{R}_q^\times$ are also primes of $\mathcal{R}$ of small norms and that no a_i is a b_j for $i \neq j \in \{1, \ldots, n\}$;[10] (3) $\{a_i\} \cap L = \varnothing$; and (4) the degree of the modulus q is greater than the degree of the product of any t primes in $\boldsymbol{a}$ and any prime in L, and smaller than the degree of the product of any $t+1$ primes in $\boldsymbol{a}$ and any prime in L, where t is the threshold of the fuzzy Hamming distance matching functions.

A fuzzy Hamming distance matching function is a boolean function $f_{\boldsymbol{x},t} : \{0,1\}^n \to \{0,1\}; \boldsymbol{y} \mapsto f_{\boldsymbol{x},t}(\boldsymbol{y})$ with respect to a secret vector $\boldsymbol{x} \in \{0,1\}^n$ and a threshold $0 < t < n \in \mathbb{N}$ evaluates to 1 if and only if the Hamming distance $|\boldsymbol{x} - \boldsymbol{y}|$ between $\boldsymbol{x}$ and $\boldsymbol{y}$ is not greater than t.

The obfuscator is as follows. Given a fuzzy Hamming distance matching function $f_{\boldsymbol{x},t}$, we compute an MESS instance $(\boldsymbol{a}, X = \boldsymbol{a}^{\boldsymbol{x}} \cdot e \pmod q)$ and publish it together with the obfuscated point function $O_P(f_{\boldsymbol{x}})$ as the obfuscated fuzzy Hamming distance matching function. To evaluate, compute $E = X \cdot a^{-\boldsymbol{y}} = \boldsymbol{a}^{\boldsymbol{x}-\boldsymbol{y}} \cdot e \pmod q$. Now if the Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ is not greater than t, i.e., $|\boldsymbol{x} - \boldsymbol{y}| \leq t$, then "$\pmod q$" can be removed and we have $E = \boldsymbol{a}^{\boldsymbol{x}-\boldsymbol{y}} \cdot e$. Denote $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{y} \in \{-1, 0, 1\}^n$. Note also that the inverses b_i's of the a_i's are all primes of $\mathcal{R}$. We can therefore denote the i-th term in E to be $c_i := a_i$ if $z_i = 0$ or 1, and $c_i := b_i$ if $z_i = -1$, and replace the exponential vector $\boldsymbol{z}$ by $\boldsymbol{w}$ which is $\boldsymbol{z}$ with all -1 replaced by 1. Denote $\boldsymbol{b} = (b_1, \ldots, b_n)$ and $\boldsymbol{c} = (c_1, \ldots, c_n)$. Then $E = \boldsymbol{c}^{\boldsymbol{w}} \cdot e$. We can then recover $\boldsymbol{z}$ by testing divisibilities "$(a_i|E)$" or "$(b_i|E)$" for $i = 1, \ldots, n$ using the primes $a_i \in \boldsymbol{a}$ and $b_i \in \boldsymbol{b}$. Having $\boldsymbol{y}$ and $\boldsymbol{z}$ we can recover $\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{z}$; and the noise e follows. In this case, $\boldsymbol{y}$ will be accepted by testing $O_P(f_{\boldsymbol{x}}(\boldsymbol{x})) = 1$. Otherwise, if the Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ is greater than t, then "$\pmod q$" cannot be removed, and the above decoding process will either fail or resulting in a wrong vector $\boldsymbol{x}' \neq \boldsymbol{x}$. Then $\boldsymbol{y}$ will be rejected by the fact that $O_P(f_{\boldsymbol{x}}(\boldsymbol{x}')) \neq 1$.

The advantages of this obfuscator compared with [23] are that (1) it drops the use of continuous fractions and Diophantine approximation (or extended Legendre theorem) for decoding and thus is adaptive to more general Euclidean domains rather than just the ring $\mathbb{Z}$ of integers; (2) in the case of $\mathbb{Z}$ the lack of using Diophantine approximation (or extended Legendre theorem) in decoding is expected to gain considerable efficiency; and (3) the same function $f_{\boldsymbol{x},t}$ can be obfuscated many times using this obfuscator, while in the case of [23], a function $f_{\boldsymbol{x},t}$ can only be obfuscated once, and that obfuscating it many times is not secure.

5.7. **Reusable fuzzy extractor.** This application uses the same MESS variant as in Section 5.6.3.

[10] To see the existence of many coinverse prime pairs (a_i, b_i) (ignoring their norms), let us take $\mathcal{R} := \mathbb{Z}$ as an example. By a simple program we can find out the smallest prime q such that $\mathbb{Z}_q^\times$ has at least 256 coinverse prime pairs (including both (a_i, b_i) and (b_i, a_i) for $a_i \neq b_i$ and those pairs (a_i, b_i) with $a_i = b_i$) is 16831. The actual number of coinverse prime pairs in $\mathbb{Z}_{16831}^\times$ is 266.

Secure sketches and fuzzy extractors were firstly introduced by Dodis, Reyzin and Smith in [19]. The idea of secure sketch is to encode a metric point $\boldsymbol{x}$ such that only a point $\boldsymbol{y}$ that is close to $\boldsymbol{x}$ can recover $\boldsymbol{x}$. The idea of fuzzy extractor is to encode a pseudo-random string $\boldsymbol{s}$ using a metric point $\boldsymbol{x}$ such that only a point $\boldsymbol{y}$ close to $\boldsymbol{x}$ can recover $\boldsymbol{s}$ with the help of some helper information $\boldsymbol{h}$. Reusable fuzzy extractor [11] further requires that the secret $\boldsymbol{x}$ is protected even given many encodings of it.

Note that there is a generic construction of fuzzy extractors from secure sketches [19]. The idea is to create $\boldsymbol{s}$ from $\boldsymbol{x}$ together with a pseudo-random string $\boldsymbol{h}$ (e.g. take XOR as: $\boldsymbol{s} = \boldsymbol{x} \oplus \boldsymbol{h}$), sketch $\boldsymbol{x}$ as $\mathsf{SK}(\boldsymbol{x})$, and publish $\mathsf{SK}(\boldsymbol{x})$ and $\boldsymbol{h}$. Then with a point $\boldsymbol{y}$ close to $\boldsymbol{x}$ one could recover $\boldsymbol{x}$ from $\mathsf{SK}(\boldsymbol{x})$ and then recover $\boldsymbol{s} = \boldsymbol{x} \oplus \boldsymbol{h}$. Therefore in the following we only construct reusable secure sketch, and that reusable fuzzy extractor is implied.

In fact, the encoding and decoding algorithms of the obfuscation of fuzzy Hamming distance functions given in Section 5.6.3 are naturally a reusable secure sketch and there is nothing else we need to do.

6. Final remarks

We introduced a new class of abstract equation over Euclidean domains. We proved its computational properties and shown cryptographic applications. The new equations introduce properties not present in existing computational problems, often related to primes in Euclidean domains, as reflected in our applications. The applications presented here represent only the tip of the iceberg. The theoretical significance and practical potential of this new class of equations, with their rich structural implications, remain to be fully uncovered through case studies across diverse Euclidean domains. These include domains of central interest in algebraic number theory and algebraic geometry, such as rings of algebraic integers (e.g., Gaussian, Eisenstein, and quadratic integers), univariate polynomial rings over fields, function fields, certain cyclotomic and cubic rings, and coordinate rings associated with algebraic varieties and arithmetic schemes.

References

[1] M. Agrawal, N. Kayal and N. Saxena, PRIMES is in P, *Annals of Mathematics*, **160** (2004), 781-793.

[2] M. Ajtai, Generating hard instances of lattice problems, *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, (1996), 99-108.

[3] T. M. Apostol, *Introduction to Analytic Number Theory*, Springer, New York, (2013).

[4] E. Bach, Explicit bounds for primality testing and related problems, *Mathematics of Computation*, **55** (1990), 355-380.

[5] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan and K. Yang, On the (im) possibility of obfuscating programs, *Annual International Cryptology Conference (CRYPTO)*, (2001), 1-18.

[6] J. Bartusek, T. Lepoint, F. Ma and M. Zhandry, New Techniques for Obfuscating Conjunctions, *Advances in Cryptology – EUROCRYPT 2019*, (2019), 636-666.

[7] E. Berlekamp, R. McEliece and H. van Tilborg, On the inherent intractability of certain coding problems (corresp.), *IEEE Transactions on Information Theory*, **24** (1978), 384-386.

[8] A. Bishop, L. Kowalczyk, T. Malkin, V. Pastro, M. Raykova and K. Shi, A simple obfuscation scheme for pattern-matching with wildcards, *Annual International Cryptology Conference (CRYPTO)*, (2018), 731-752.

[9] A. Blum, M. Furst, M. Kearns and R. J. Lipton, Cryptographic primitives based on hard learning problems, *Advances in Cryptology—CRYPTO'93*, (1994), 278-291.

[10] D. Boneh and A. Silverberg, Applications of multilinear forms to cryptography, *Contemporary Mathematics*, **324** (2003), 71-90.
[11] R. Canetti, B. Fuller, O. Paneth, L. Reyzin and A. Smith, Reusable fuzzy extractors for low-entropy distributions, *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, (2016), 117-146.
[12] J. H. Cheon, K. Han, C. Lee, H. Ryu and D. Stehlé, Cryptanalysis of the multilinear map over the integers, *Advances in Cryptology–EUROCRYPT 2015*, (2015), 3-12.
[13] J.-S. Coron, M. S. Lee, T. Lepoint and M. Tibouchi, Cryptanalysis of GGH15 multilinear maps, *Advances in Cryptology–CRYPTO 2016*, (2016), 607-628.
[14] J.-S. Coron, T. Lepoint and M. Tibouchi, New multilinear maps over the integers, *Annual Cryptology Conference*, (2015), 267-286.
[15] J.-S. Coron, T. Lepoint and M. Tibouchi, Practical multilinear maps over the integers, *Annual Cryptology Conference*, (2013), 476-493.
[16] H. Davenport, *Multiplicative Number Theory*, 3rd ed., edited by H. L. Montgomery, *Graduate Texts in Mathematics*, **74** (2000), Springer–Verlag, New York.
[17] J. Ding and A. Petzoldt, Current state of multivariate cryptography, *IEEE Security & Privacy*, **15** (2017), 28-36.
[18] J. Ding and B.-Y. Yang, Multivariate public key cryptography, *Post-Quantum Cryptography*, (2009), 193-241.
[19] Y. Dodis, L. Reyzin and A. Smith, Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, *Advances in Cryptology - EUROCRYPT 2004*, (2004), 523-540.
[20] L. Ducas, Shortest vector from lattice sieving: A few dimensions for free, *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, (2018), 125-145.
[21] S. D. Galbraith and T. Li, Obfuscation of evasive algebraic set membership, *Advances in Mathematics of Communications*, **19** (2025), 604-627.
[22] S. D. Galbraith and T. Li, Small superset and big subset obfuscation, *Information Security and Privacy*, (2021), 68-87.
[23] S. D. Galbraith and L. Zobernig, Obfuscated Fuzzy Hamming distance and conjunctions from subset product problems, *Theory of Cryptography*, (2019), 81-110.
[24] S. Garg, C. Gentry and S. Halevi, Candidate multilinear maps from ideal lattices, *Advances in Cryptology–EUROCRYPT 2013*, (2013), 1-17.
[25] G. H. Hardy and J. E. Littlewood, Some problems of 'Partitio numerorum'; III: On the expression of a number as a sum of primes, *Acta Mathematica*, **44** (1923), 1-70.
[26] J. Hoffstein, NTRU: A Ring Based Public Key Cryptosystem, *Algorithmic Number Theory (ANTS III)*, (1998).
[27] J. Hoffstein, J. Pipher and J. H. Silverman, *An Introduction to Mathematical Cryptography*, Springer, (2008).
[28] I. D. Ion and G. Militaru, Rings with finite quotients, *Bulletin Mathématique de la Société des Sciences Mathématiques de Roumanie*, **37** (1993), 29-39.
[29] R. Jozsa, Quantum factoring, discrete logarithms, and the hidden subgroup problem, *Computing in Science & Engineering*, **3** (2001), 34-43.
[30] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 3rd ed., Chapman & Hall/CRC, (2020).
[31] M. Kearns, Efficient noise-tolerant learning from statistical queries, *Journal of the ACM (JACM)*, **45** (1998), 983-1006.
[32] A. Langlois and D. Stehlé, Worst-case to average-case reductions for module lattices, *Designs, Codes and Cryptography*, **75** (2015), 565-599.
[33] A. Langlois, D. Stehlé and R. Steinfeld, GGHLite: More efficient multilinear maps from ideal lattices, *Advances in Cryptology–EUROCRYPT 2014*, (2014), 239-256.
[34] V. Lyubashevsky, C. Peikert and O. Regev, On ideal lattices and learning with errors over rings, *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, (2010), 1-23.
[35] D. Micciancio, Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions, *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, (2002), 356-365.
[36] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems: A Cryptographic Perspective*, *The Springer International Series in Engineering and Computer Science*, vol. 671, Springer-Verlag, New York, (2002).

[37] D. Micciancio and C. Peikert, Hardness of SIS and LWE with small parameters, *Advances in Cryptology–CRYPTO 2013*, (2013), 21-39.
[38] C. Peikert and A. Rosen, Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices, *Theory of Cryptography Conference*, (2006), 145-166.
[39] K. Pietrzak, Cryptography from learning parity with noise, *International Conference on Current Trends in Theory and Practice of Computer Science*, (2012), 99-114.
[40] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, *Journal of the ACM (JACM)*, **56** (2009), Art. 34, 40 pp.
[41] R. L. Rivest, L. Adleman and M. L. Dertouzos, On data banks and privacy homomorphisms, *Foundations of Secure Computation*, **4** (1978), 169-180.
[42] P. Rogaway and T. Shrimpton, Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance, *Fast Software Encryption: 11th International Workshop, FSE 2004*, (2004), 371-388.
[43] P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, *Proceedings 35th Annual Symposium on Foundations of Computer Science*, (1994), 124-134.
[44] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Review*, **41** (1999), 303-332.
[45] C. L. Siegel, A mean value theorem in geometry of numbers, *Annals of Mathematics*, **46** (1945), 340-347. `http://www.jstor.org/stable/1969027`.
[46] D. Stehlé, R. Steinfeld, K. Tanaka and K. Xagawa, Efficient public key encryption based on ideal lattices, *International Conference on the Theory and Application of Cryptology and Information Security*, (2009), 617-635.
[47] C. Sun, M. Tibouchi and M. Abe, Revisiting the hardness of binary error LWE, *Information Security and Privacy: 25th Australasian Conference, ACISP 2020*, (2020), 425-444.
[48] M. Tibouchi, Cryptographic multilinear maps: A status report, *Tech. Rep.*, CRYPTREC-EX-2603-2016, `http://www.cryptrec.go.jp/report/ex/cryptrec-ex-2603-2016.pdf`, (2016).