Improved Constructions of PRFs Secure Against Related-Key Attacks

Kevin Lewi[*] Hart Montgomery[*] Ananth Raghunathan[*]

May 8, 2014

Abstract

Building cryptographic primitives that are secure against related-key attacks (RKAs) is a well-studied problem by practitioners and theoreticians alike. Practical implementations of block ciphers take into account RKA security to mitigate fault injection attacks. The theoretical study of RKA security was initiated by Bellare and Kohno (Eurocrypt '03). In Crypto 2010, Bellare and Cash introduce a framework for building RKA-secure pseudorandom functions (PRFs) and use this framework to construct RKA-secure PRFs based on the decision linear and DDH assumptions.

We build RKA-secure PRFs by working with the Bellare-Cash framework and the LWE- and DLIN-based PRFs recently constructed by Boneh, Lewi, Montgomery, and Raghunathan (Crypto '13). As a result, we achieve the first PRFs from lattices secure against an (almost) linear class of related-key functions. In addition, we note that our DLIN-based PRF (based on multilinear maps) is the first RKA-secure PRF for affine classes under the DLIN assumption, and the first RKA-secure PRF against a large class of polynomial functions under a natural generalization of the DLIN assumption. Previously, RKA security for higher-level primitives (such as signatures and IBEs) were studied in Bellare, Paterson, and Thomson (Asiacrypt '12) for affine and polynomial classes, but the question of RKA-secure PRFs for such classes remained open.

Although our RKA-secure LWE-based PRF only applies to a restricted linear class, we show that by weakening the notion of RKA security, we can handle a significantly larger class of affine functions. Finally, the results of Bellare, Cash, and Miller (Asiacrypt '11) show that all of our RKA-secure PRFs can be used as building blocks for a wide variety of public-key primitives.

Keywords: related-key attacks, pseudorandom functions, learning with errors.

1 Introduction

The usual notions of security for cryptographic primitives do not address the possibility that an attacker could adversarially modify the internal state of hardware devices that implement the primitive. Indeed, fault injection attacks (and other types of side-channel attacks including cold-boot attacks [21], timing attacks [23, 15], and power analysis attacks [26]) have shown that our traditional security definitions are not sufficient for most practical implementations of provably secure cryptographic primitives [11, 12, 32, 5].

[*] Stanford University. Email: {klewi,hartm,ananthr}@cs.stanford.edu

To deal with fault injection attacks, cryptographers have developed the notion of related-key attack (RKA) security. RKA security definitions [8] capture the following notion: in addition to allowing the adversary to make input queries on the primitive for a randomly chosen secret key, the adversary is allowed to make input queries on the primitive for adversarially chosen "related-key deriving" functions $\phi \in \Phi$ of a randomly chosen secret key (where Φ is a function family specified in advance). This notion can be used to show that certain classes of tampering attacks are ineffective against primitives proven secure in the presence of RKAs.

In the past few years, there has been much work in constructing RKA-secure primitives [6, 7, 2, 10, 34, 9]. In addition, RKA security is also of interest to practitioners, particularly in the design of block ciphers [18, 22, 35]. In this work, we will focus our attention on building one of the most basic of the RKA primitives—pseudorandom functions (PRFs). Not only do PRFs find applications in many real-world implementations where side-channel attacks are possible (and hence RKA security becomes relevant) [5], but RKA-secure PRFs are also known to imply RKA security for a wide range of more advanced primitives, including signatures, identity-based encryption, and both public-key and private-key chosen ciphertext secure encryption [7].

1.1 Background and Related Work

Bellare and Cash [6] developed the first RKA-secure PRF for a non-trivial class of functions. Instantiations prior to [6] on RKA-secure PRFs required ideal ciphers, random oracles, or non-standard assumptions [25, 8]. In addition, Bellare and Cash develop a novel framework (which we call the BC framework) for building RKA-secure PRFs, and show how the DDH assumption implies an RKA-secure PRF for the class $\Phi_{\mathsf{prod}} = \{\phi_{\mathbf{a}} : \mathbb{Z}_q^m \to \mathbb{Z}_q^m \mid \phi_{\mathbf{a}}(\mathbf{k}) = \mathbf{k} \times \mathbf{a}\}_{\mathbf{a} \in \mathbb{Z}_q^m}$, the class of all Hadamard product (component-wise product) transformations to the key. Additionally, they construct an RKA-secure PRF under the DLIN assumption [33, 29] for an interesting multiplicative class Φ (where related keys are derived from scalar multiples of components of the key).

Bellare *et al.* [7] explore the possibilities of transferring RKA security from one primitive to another (while preserving the class Φ of related-key deriving functions). In particular, they show that RKA-secure PRFs can be used to construct a wide variety of higher-level RKA-secure primitives. Thus, improvements in building RKA-secure PRFs have wide applicability to RKA-secure public-key cryptographic primitives.

Applebaum *et al.* [2] show how to build RKA-secure symmetric encryption from a variety of hardness assumptions for linear related-key attacks. Wee [34] presents chosen ciphertext RKA-secure public-key encryption scheme constructions from the DBDH and LWE assumptions for linear related-key attacks. Finally, Bellare *et al.* [10] show how to build RKA-secure variants from a variety of primitives discussed in [7] for more expressive classes Φ including affine and polynomial function families. However, constructing RKA-secure PRFs for linear, affine, or polynomial Φ is notably left open. Concurrently, Bellare *et al.* [9] build RKA-secure signature schemes against related-key deriving functions drawn from such classes of polynomials. Their construction relies on RKA-secure one-way functions which appear to be easier to build under standard assumptions (as opposed to RKA-secure PRFs).

PRFs are extremely well-studied primitives and have been built from a wide variety of assumptions [28, 17, 24, 14, 4, 13]. Currently known RKA-secure PRFs only consider the Naor-Reingold [28] and Lewko-Waters [24] PRFs. We note that PRFs constructed by Boneh *et al.* [13] satisfy an additional "key homomorphism" property which we find useful in constructing RKA-secure PRFs. Our constructions are based on the PRFs considered in this work.

1.2 Our Contributions

Lattice-based RKA-secure PRFs. We present the first lattice-based PRFs secure against related-key attacks. Our construction achieves RKA security under the standard LWE assumption against the class of related-key functions $\Phi_{\mathsf{lin}^*} = \{\phi_{\mathbf{a}} : \mathbb{Z}_q^m \to \mathbb{Z}_q^m \mid \phi_{\mathbf{a}}(\mathbf{k}) = \mathbf{k} + \mathbf{a}\}_{\mathbf{a} \in (\frac{q}{p})\mathbb{Z}_q^m}$ over the key space $\mathcal{K} = \mathbb{Z}_q^m$. The class $(\frac{q}{p})\mathbb{Z}_q^m$ here denotes the vectors in $\mathbb{Z}_q^m$ whose entries are all multiples of q/p (where p divides q). Ideally we would like to address RKA security for the entire class of linear key shifts, but we only achieve a weaker notion of security. However, these restrictions are quite plausible as they translate to an adversary that can inject faults into the higher order bits of the key.[1]

RKA security against an affine class of related keys. Next, we show how the powerful multilinear map abstraction by Garg *et al.* [19] along with the DLIN assumption in this abstraction can be used to construct PRFs with RKA security against a very large and natural class of affine key transformations $\Phi_{\mathsf{aff}} = \{\phi_{\mathbf{C},\mathbf{B}} : \mathbb{Z}_p^{m\times\ell} \to \mathbb{Z}_p^{m\times\ell} \mid \phi_{\mathbf{C},\mathbf{B}}(\mathbf{K}) = \mathbf{C}\mathbf{K} + \mathbf{B}\}$ over the key space $\mathcal{K} = \mathbb{Z}_p^{m\times\ell}$. For Φ_{aff}, we require that $\mathbf{C}$ comes from a family of *invertible* matrices and that Φ_{aff} be claw-free—for all $\phi_1, \phi_2 \in \Phi_{\mathsf{aff}}$ and $\mathbf{K} \in \mathcal{K}$, $\phi_1(\mathbf{K}) \neq \phi_2(\mathbf{K})$.

Both restrictions arise from a technical requirement under the BC framework. As noted in [6, 10], some restrictions must be placed on Φ_{aff} in order for PRFs to achieve RKA security against them (for example, Φ_{aff} cannot include constant functions $\phi(\mathbf{K}) = \mathbf{B}$). Hence, our class Φ_{aff} is essentially the most expressive affine class of transformations for which RKA PRF security is still attainable under the Bellare-Cash framework. In fact, there are no known PRFs which are RKA-secure against a class which does not have the claw-free restriction. Bellare *et al.* [10] constructed higher-level primitives RKA-secure against affine classes, but left open the problem of constructing such a PRF (for which we provide an answer).

Unique-input RKA security against an affine class. We note, however, that the assumption that there exists an instantiation of the Garg *et al.* multilinear map abstraction [19] for which DLIN holds is a fairly strong assumption. This raises the following question: Can we achieve a similar result for RKA PRF security against affine transformations from a more standard assumption? We answer this question in the affirmative by considering a slightly weaker notion of RKA security, denoted *unique-input* RKA security, where adversary queries are restricted to unique inputs. We build RKA-secure PRFs from the LWE assumption that can handle the class of transformations $\Phi_{\mathsf{ln\text{-}aff}} = \{\phi_{\mathbf{C},\mathbf{B}} : \phi_{\mathbf{C},\mathbf{B}}(\mathbf{K}) = \mathbf{C}\mathbf{K} + \mathbf{B}\}$, where $\mathbf{C}$ is a full-rank "low-norm" matrix and $\mathbf{B}$ is an arbitrary matrix in $\mathbb{Z}_q^{m\times m}$ from the LWE assumption. We observe that under this weaker notion of security, our class is significantly more expressive than our first result from lattices because it allows for the addition of arbitrary vectors. However, this requires us to work outside the Bellare-Cash framework. We leave it as an open problem to construct "truly" RKA-secure PRFs from LWE (or other standard assumptions, such as DDH) for an affine class of key transformations.

Unique-input RKA security against a class of polynomials. We further explore the connection between key homomorphism and unique-input RKA security by using the multilinear map abstraction to tackle a polynomial class of related-key functions. More specifically, we consider

[1] We note that when q and p are powers of 2, Φ_{lin^*} captures all functions that perform linear shifts on the entries of the key that do not modify the $\log(q/p)$-least significant bits of each entry.

the class of polynomials $\Phi_{\mathsf{poly}(d)}$ of bounded degree d over matrices $\mathbb{Z}_q^{m\times m}$ and consider a natural exponent assumption over multilinear maps called the Multilinear Diffie-Hellman Exponent (MDHE) assumption. For technical reasons, we require that at least one of the polynomial's non-constant coefficient matrices is full-rank. This natural restriction simply ensures that the output of the polynomial is sufficiently random given a uniformly drawn input of a special form. We note that the MDHE assumption is a natural and fairly plausible generalization of the DLIN assumption.

Finally, we can apply the results of [7] to get Φ-RKA security for signatures, identity-based encryption, and public and private key CCA encryption from our Φ-RKA-secure PRFs.

1.3 Our Techniques

At a high level, we use the Bellare-Cash framework with the (LWE- and DLIN-based) key homomorphic PRFs from Boneh *et al.* [13] to construct RKA-secure PRFs against the classes Φ_{lin^*} and Φ_{aff}. Below, we give an outline of the framework and note that key homomorphic PRFs are a natural starting point due to the malleability requirement of the framework.

Bellare-Cash framework. The only known construction of RKA-secure PRFs to date is that of Bellare and Cash [6]. In their framework, Bellare and Cash identify sufficient properties for constructing an RKA-secure PRF. They first consider PRFs $F\colon \mathcal{K}\times\mathcal{X}\to\mathcal{Y}$ that are *key malleable*—PRFs which have an efficient algorithm (denoted a transformer T) that when given an input $(\phi, x)\in\Phi\times\mathcal{X}$ and oracle access to $F(k,\cdot)$ computes $F(\phi(k),x)$. In addition, T must satisfy a *uniformity* property, namely, when $F(k,\cdot)$ is replaced with a random function $f(\cdot)$, the outputs of T on inputs $(\phi_1,x_1),\ldots,(\phi_Q,x_Q)$ for distinct $x_1,\ldots,x_Q$ are uniform and independently distributed. The framework also requires the existence of a *key fingerprint*—an input $w\in\mathcal{X}$ such that for all $k\in\mathcal{K}$ and distinct $\phi_1,\phi_2\in\Phi$, $F(\phi_1(k),w)\neq F(\phi_2(k),w)$.

For a class Φ with a suitable key malleable PRF, a fingerprint w, and a collision-resistant hash function that satisfies a simple *compatiblity* property H_{com} (see Definition 2.9), under the Bellare-Cash framework, the authors show that the PRF $F_{\mathsf{rka}}(k,x) = F\left(k, H_{\mathsf{com}}(x, F\left(k,w\right))\right)$ is Φ-RKA-secure.

Applying the BC framework to the DLIN-based PRF. Our starting point is the construction of a DLIN-based key homomorphic PRF by Boneh *et al.* [13], who note that key homomorphic PRFs are key malleable. In this work, we generalize this PRF to operate with the key space $\mathcal{K}=\mathbb{Z}_p^{m\times\ell}$ instead of $\mathbb{Z}_p^{\ell}$. The PRF has public parameters $\mathbf{A}_0,\mathbf{A}_1\in\mathbb{Z}_q^{\ell\times\ell}$. On input x, the PRF is of the form $(g_\ell)^{\mathbf{W}}$ for $\mathbf{W}=\mathbf{KP}$ where $\mathbf{P}\in\mathbb{Z}_p^{\ell\times\ell}$ is the publicly computable matrix $\mathbf{A}_{x_\ell}\mathbf{A}_{x_{\ell-1}}\cdots\mathbf{A}_{x_1}$ (that only depends on the bits of x) and g_ℓ is the generator of a group with a multilinear map. This additional algebraic structure allows us to consider the class of affine related-key deriving functions of the form $\mathbf{CK}+\mathbf{B}$ for matrices $\mathbf{C}\in\mathbb{Z}_q^{m\times m}$ and $\mathbf{B}\in\mathbb{Z}_q^{m\times\ell}$. The pseudorandomness of the PRF holds by a straightforward hybrid argument, noting that the rows of $\mathbf{K}$ are now identical to independent keys of the original PRF.

Working in the exponent, given access to an oracle that computes $\mathbf{W}$ and an input $\phi_{\mathbf{C},\mathbf{B}}$, it is easy to construct a transformer that computes $\mathbf{W}'=\mathbf{CW}+\mathbf{BP}$. From some simple algebra, one can verify that this indeed computes the exponent $\mathbf{W}'$ corresponding to $F_{\text{DLIN}}(\phi(\mathbf{K}),x)$. In addition, as long as $\mathbf{C}$ is restricted to the set of full-rank matrices, it follows that the transformer described above outputs uniform matrices if $\mathbf{W}$ corresponds to the outputs of a random function. From this, the rest of the BC framework can be applied and is shown in Section 3.2. We note here that the restriction that Φ is claw-free seems to be inherently required in applying the BC framework (here, we require it in constructing a suitable fingerprint), and we do not overcome this limitation in our construction either.[2]

Applying the BC framework to the LWE-based PRF. Recollect that Boneh *et al.* construct an "almost" key homomorphic LWE-based PRF F which on input x is of the form $\lfloor \mathbf{P}\mathbf{k} \rceil_p$, where $\mathbf{P} = \mathbf{A}_{x_\ell}\mathbf{A}_{x_{\ell-1}} \cdots \mathbf{A}_{x_1}$. (Here, $\lfloor x \rceil_p$ for $x \in \mathbb{Z}_q$ denotes multiplying x by p/q and rounding the result to $\mathbb{Z}_p$.) Unfortunately, the "almost"-ness of the key homomorphism disallows a direct argument of key malleability. Furthermore, a transformer which is "almost" key malleable (in the same sense) is still insufficient for instantiating the BC framework.

This limitation can be overcome by observing that $F(\mathbf{k}_1, x) + F(\mathbf{k}_2, x) = F(\mathbf{k}_1 + \mathbf{k}_2, x)$ if the entries of either $\mathbf{k}_1$ or $\mathbf{k}_2$ are all multiples of q/p. This property is sufficient to show that F is key malleable with respect to the class Φ_{lin^*}, where $\mathbf{k}_2$ is required to be an element of $(\frac{q}{p})\mathbb{Z}_q^m$. Additionally, this restriction is needed show that any fixed input $w \in \{0,1\}^\ell$ acts as a key fingerprint for F under the class Φ_{lin^*}. It seems likely that this restriction is in fact necessary for applying the BC framework, leaving this the most expressive class achievable for the LWE-based PRF F.

One natural question to ask is whether the Banerjee *et al.* [4] LWE-based PRF can be used instead of F. We note that their PRF is not key homomorphic and hence the above approach does not apply. However, we leave open the question of achieving unique-input RKA security for their PRF (see Section 6).

Unique-input adversaries. As was observed by Bellare and Cash, key malleability is intuitively useful in constructing RKA security because it allows us to simulate $F(\phi(k), \cdot)$ without access to the key k but also leads to a simple related-key attack against any class that contains the functions ϕ_{id} (the identity function) and any $\phi' \neq \phi_{\mathrm{id}}$. The difficulty in achieving security lies in the adversary's ability to request multiple related-key deriving functions on the same input x. Given ϕ_{id}, to attack the pseudorandomness, the adversary can run the transformer for ϕ' himself and compare the output of the transformer to the output of the oracle on (ϕ', x). Thus, Bellare and Cash require additional tools.

However, the notion of key malleability suffices to show security against unique-input adversaries, where the adversary's queries are restricted to distinct x's. In extending the RKA-secure LWE-based PRF to a class of affine functions, as discussed earlier in this section, the presence of the rounding does not directly imply key malleability. However, in Section 4, we work through the proof of security of the pseudorandomness of F, along the lines of the proof in [13], to consider its RKA security against the larger class $\Phi_{\mathsf{ln\text{-}aff}}$. We show that the structure of the PRF allows us to simulate, in addition to PRF queries on input x, RKA queries for functions $\phi \in \Phi_{\mathsf{ln\text{-}aff}}$. As in [13], the proof works through several hybrid arguments that modify a challenger from a truly random function to a pseudorandom function that also provides answers to RKA queries $(\phi, x) \in \Phi_{\mathsf{ln\text{-}aff}} \times \{0,1\}^\ell$.

The low-norm restriction on the matrix $\mathbf{C}$ in $\phi_{\mathbf{C},\mathbf{B}} \in \Phi_{\mathsf{ln\text{-}aff}}$ is required to ensure that when using LWE challenges in the hybrids, the noise does not grow larger than what the rounding allows. In the final hybrid, the adversary interacts with uniform and independently chosen outputs corresponding to inputs x_i. As long as the adversary is restricted to unique inputs, this interaction is identical to

[2] However, in [7], the authors overcome this barrier and achieve RKA security for PRGs, not PRFs, against a class Φ which is not claw-free.

the game where the adversary receives uniform and independent (consistent) values on queries (ϕ, x). This is sufficient to show RKA security. Whether we can take advantage of the algebraic structure of other pseudorandom functions to directly prove unique-input RKA security is an interesting question.

Unique-input security against a class of polynomials. We have shown how under the DLIN and LWE assumptions we can build RKA-secure PRFs for classes of affine functions, but unfortunately we do know how to extend these results to handle classes of polynomials. However, in Section 5, we show that the PRF F_{DLIN} (defined in Section 3.2) is RKA-secure against unique-input adversaries under the (new) d-MDHE assumption (see Definition 2.6) for a class of degree-d polynomials.

For integers ℓ, d, and a prime p, we consider the class $\Phi_{\mathsf{poly}(d)}$ consisting of all degree-d polynomials over $\mathbb{Z}_p^{\ell \times \ell}$ of the form $P(\mathbf{K}) = \sum_{i=0}^{d} \mathbf{C}_i \cdot \mathbf{K}^i$, where $\mathbf{C}_0, \ldots, \mathbf{C}_d, \mathbf{K} \in \mathbb{Z}_p^{\ell \times \ell}$ and at least one of $\mathbf{C}_1, \ldots, \mathbf{C}_d$ is full rank. To prove the RKA security of F_{DLIN} against unique-input adversaries, we consider a series of hybrid experiments which respond to queries $(\phi_{P(\cdot)}, x) \in \Phi_{\mathsf{poly}(d)} \times \{0,1\}^{\ell}$, where $P(\mathbf{S}) = \sum_{i=0}^{d} \mathbf{C}_i \cdot \mathbf{S}^i$, by choosing d uniformly random, *independent* secrets $\mathbf{K}_1, \ldots, \mathbf{K}_d$ and computing the weighted sum $\mathbf{C}_0 + \sum_{i=1}^{d} \mathbf{C}_i \cdot \mathbf{K}_i$, as opposed to choosing a single uniformly random secret $\mathbf{S}$ and computing $P(\mathbf{S})$. We show how an adversary which distinguishes between these two cases can be used to break the d-MDHE assumption, and then we use the techniques used to prove the pseudorandomness of F_{DLIN} to complete the argument.

The additional requirement of at least one of $\mathbf{C}_1, \ldots, \mathbf{C}_d$ being full rank is only needed to ensure that a sufficient amount of entropy from the secret key will remain in the output of the PRF. Note that this restriction on $\Phi_{\mathsf{poly}(d)}$ rules out polynomials P for which the output of P on randomly chosen key can be predicted (as an example consider *constant* polynomials $P(\mathbf{K}) = \mathbf{C}$ for some fixed $\mathbf{C} \in \mathbb{Z}_p^{\ell \times \ell}$), for which achieving RKA security is impossible. We believe $\Phi_{\mathsf{poly}(d)}$ captures what is essentially the most expressive class of bounded-degree polynomials for RKA-secure PRFs.

Organization. In Section 2 we introduce preliminary notation and definitions. In Section 3 we construct RKA-secure LWE- and DLIN-based PRFs using the BC framework. Then, in Section 4, we give an LWE-based RKA-secure PRF against unique-input adversaries for an affine class of transformations. In Section 5, we show how the DLIN-based PRF is secure against unique-input adversaries where the related-key attacks come from a class of bounded-degree polynomials. We conclude in Section 6. In Appendix A we give a security proof of the d-MDHE assumption in the generic group model.

2 Preliminaries

2.1 Notation

Rounding. We define $\lfloor \cdot \rfloor$ to round a real number to the largest integer which does not exceed it. For integers q and p where $q \geq p \geq 2$, we define the function $\lfloor \cdot \rceil_p : \mathbb{Z}_q \to \mathbb{Z}_p$ as $\lfloor x \rceil_p = i$ where $i \cdot \lfloor q/p \rfloor$ is the largest multiple of $\lfloor q/p \rfloor$ which does not exceed x. For a vector $\mathbf{v} \in \mathbb{Z}_q^m$, we define $\lfloor \mathbf{v} \rceil_p$ as the vector in $\mathbb{Z}_p^m$ obtained by rounding each coordinate of the vector individually.

When $p \,|\, q$, we let $(\frac{q}{p})\mathbb{Z}_q$ denote the subgroup of $\mathbb{Z}_q$ comprising the set $\{(q/p) \cdot x \mid x \in \mathbb{Z}_q\}$. The following lemma follows from some elementary arithmetic.

Lemma 2.1. *For any $u \in (\frac{q}{p})\mathbb{Z}_q$ and $x \in \mathbb{Z}_q$ such that $u \equiv x(q/p) \mod q$ and any $y \in \mathbb{Z}_q$,*

$$\lfloor y + u \rfloor_p = \lfloor y \rfloor_p + \lfloor u \rfloor_p = \lfloor y \rfloor_p + x \pmod{p}.$$

Groups. For a matrix $\mathbf{M}$, we let the component-wise exponentiation $g^{\mathbf{M}}$ denote a matrix with entries $g^{\mathbf{M}_{i,j}}$. We let $\left(g^{\mathbf{A}}\right)^{\mathbf{B}}$ denote the matrix with entries $g^{(\mathbf{AB})_{i,j}}$. We let $\mathbf{Rk}_i(\mathbb{Z}_p^{a \times b})$ denote the set of all $a \times b$ matrices over $\mathbb{Z}_p$ of rank i.

Collision Resistance. The advantage of an efficient algorithm $\mathcal{C}$ in attacking the collision-resistance security of a hash function $H : \mathcal{D} \to \mathcal{R}$ is $\mathbf{Adv}_H^{\mathsf{cr}}(\mathcal{C}) = \Pr[x \neq x' \text{ and } H(x) = H(x')]$ where the probability is taken over $(x, x') \leftarrow \mathcal{C}$. For simplicity of exposition, we leave out the necessary syntax for function families that takes into account the non-uniformity of the above definition.

2.2 Pseudorandom Functions

We briefly review the definition of pseudorandom functions [20]. Informally, a pseudorandom function is an efficiently computable function such that no efficient adversary can distinguish the function from a truly random function given only black-box access.

More precisely, a PRF is an efficiently computable function $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ where $\mathcal{K}$ is called the key space, $\mathcal{X}$ is called the domain, and $\mathcal{Y}$ is called the range. In this paper, we allow the PRF to take additionally public parameters pp and use $F_{pp} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ to denote such a PRF. Security for a PRF is defined using two experiments between a challenger and an adversary $\mathcal{A}$. For $b \in \{0, 1\}$ define the following experiment $\mathsf{Expt}_b^{\mathsf{prf}}$:

1. Given security parameter λ, the challenger samples and publishes public parameters pp to the adversary. Next, if $b = 0$ the challenger chooses a random key $k \in \mathcal{K}$ and sets $f(\cdot) \stackrel{\mathsf{def}}{=} F_{pp}(k, \cdot)$. If $b = 1$ the challenger chooses a random function $f : \mathcal{X} \to \mathcal{Y}$.
2. The adversary (adaptively) sends input queries $x_1, \ldots, x_Q$ in $\mathcal{X}$ and receives back $f(x_1), \ldots, f(x_Q)$.
3. Eventually the adversary outputs a bit $b' \in \{0, 1\}$, which the experiment also outputs.

Definition 2.2 (Pseudorandom Function)**.** A PRF $F_{pp} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is secure if for all efficient adversaries $\mathcal{A}$ the quantity

$$\mathbf{Adv}_F^{\mathsf{prf}}(\mathcal{A}) \stackrel{\mathsf{def}}{=} \left| \Pr\left[\mathsf{Expt}_0^{\mathsf{prf}} = 1\right] - \Pr\left[\mathsf{Expt}_1^{\mathsf{prf}} = 1\right] \right|$$

is negligible.

2.3 RKA-secure PRFs

For a class of related-key deriving functions $\Phi = \{\phi : \mathcal{K} \to \mathcal{K}\}$, the notion of Φ-RKA security for a PRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is defined using an experiments between a challenger and an adversary $\mathcal{A}$. For $b \in \{0, 1\}$ define the following experiment $\mathsf{Expt}_b^{\mathsf{prf\text{-}rka}}$:

1. Given security parameter λ, the challenger samples and publishes public parameters pp to the adversary. Next, the challenger chooses a random key $k \in \mathcal{K}$ and if $b = 0$, sets $f(\cdot) \stackrel{\mathsf{def}}{=} F(k, \cdot)$. Otherwise, if $b = 1$, the challenger chooses a random keyed function $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$.

2. The adversary (adaptively) sends input queries $(\phi_1, x_1), \ldots, (\phi_Q, x_Q)$ in $\Phi \times \mathcal{X}$ and receives back $f(\phi_1(k), x_1), \ldots, f(\phi_Q(k), x_Q)$.
3. The adversary outputs a bit $b' \in \{0,1\}$, and the experiment also outputs b'.

Definition 2.3 (RKA-secure PRF for Φ)**.** A PRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is RKA-secure with respect to class Φ if for all efficient adversaries $\mathcal{A}$ the quantity

$$\mathbf{Adv}^{\mathsf{prf\text{-}rka}}_{\Phi,F}(\mathcal{A}) \overset{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Expt}^{\mathsf{prf\text{-}rka}}_0 = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{prf\text{-}rka}}_1 = 1\right]\right|$$

is negligible.

Unique-input RKA security (cf. [6]). We say that an adversary is *unique-input* in the above security game if the input queries $(\phi_1, x_1), \ldots, (\phi_Q, x_Q) \in \Phi \times \mathcal{X}$ are such that $x_1, \ldots, x_Q$ are distinct. A PRF is *unique-input* RKA-secure if it is RKA secure against unique-input adversaries.

2.4 Security Assumptions

Learning with errors (LWE) assumption. The LWE problem was introduced by Regev [31] who showed that solving the LWE problem on average is as hard as (quantumly) solving several standard lattice problems in the worst case.

Definition 2.4 (Learning With Errors)**.** For integers $q > 2$ and a noise distribution χ over $\mathbb{Z}_q$, the learning with errors problem (LWE) over n-dimensional vectors is to distinguish between the distributions $\{\mathbf{A}, \mathbf{A}^\mathsf{T}\mathbf{s} + \boldsymbol{\chi}\}$ and $\{\mathbf{A}, \mathbf{u}\}$, where $m = \text{poly}(n)$, $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\boldsymbol{\chi} \leftarrow \chi^m$, and $\mathbf{u} \leftarrow \mathbb{Z}_q^m$.

Regev [31] shows that for a certain noise distribution $\chi = \overline{\Psi}_\alpha$,[3] for n polynomial in λ and $q > 2\sqrt{n}/\alpha$, the LWE problem is as hard as the worst-case SIVP and GapSVP under a quantum reduction (see also [30, 16] for classical reductions). These results have been extended to show that $\mathbf{s}$ can be sampled from a low-norm distribution (in particular, from the noise distribution $\boldsymbol{\chi}$) and the resulting problem is as hard as the basic LWE problem [1]. Similarly, the noise distribution $\boldsymbol{\chi}$ can be a simple low-norm distribution [27]. Boneh *et al.* [13] show that the variant of LWE where the entries of $\mathbf{A}$ are binary and $m > n \log q$ is equivalent (modulo a $\log q$-factor loss in dimension) to LWE over n-dimensional vectors. In this work, we let $B \in \mathbb{R}$ be an error bound such that for $\chi \leftarrow \overline{\Psi}_\alpha$, $|\chi| \leq B$ with overwhelming probability.

Low-norm matrix LWE. We work with the right-multiplied matrix form of (low-norm) LWE, namely, that for a uniformly drawn $\mathbf{A} \leftarrow \{0,1\}^{m \times 2m}$, $\mathbf{U} \leftarrow \mathbb{Z}_q^{m \times 2m}$, $\mathbf{S} \leftarrow \mathbb{Z}_q^{m \times m}$, and $\mathbf{X} \leftarrow \chi^{m \times 2m}$, the problem is to distinguish between the distributions $\{\mathbf{A}, \mathbf{SA} + \mathbf{X}\}$ and $\{\mathbf{A}, \mathbf{U}\}$.

To compare it to the low-norm LWE variant in [13], we note that $\{\mathbf{A}, \mathbf{SA}+\mathbf{X}\}$ and $\{\mathbf{A}, \mathbf{A}^\mathsf{T}\mathbf{S}+\mathbf{X}^\mathsf{T}\}$ are distributed identically, and a standard hybrid argument shows that any adversary which can distinguish $\{\mathbf{A}, \mathbf{A}^\mathsf{T}\mathbf{S} + \mathbf{X}^\mathsf{T}\}$ from $\{\mathbf{A}, \mathbf{U}\}$ can be used to distinguish $\{\mathbf{A}, \mathbf{A}^\mathsf{T}\mathbf{s} + \boldsymbol{\chi}\}$ from $\{\mathbf{A}, \mathbf{u}\}$ with only a $(1/m)$-factor loss in advantage.

[3] For an $\alpha \in (0,1)$ and a prime q, let $\overline{\Psi}_\alpha$ denote the distribution over $\mathbb{Z}_q$ of the random variable $\lceil qX \rfloor \pmod{q}$ where X is a normal random variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$.

The DLIN assumption in multilinear groups. In Section 3.2, we rely on the decisional linear (DLIN) assumption (as stated in Boneh *et al.* [13]) for the Garg *et al.* abstraction of graded multilinear maps [19]. Consider a sequence of groups $\vec{\mathbb{G}} = (\mathbb{G}_1, \ldots, \mathbb{G}_\ell)$ with a set of bilinear maps $\hat{e}_i$ for $i \in [1, \ell - 1]$, and a generator g of $\mathbb{G}_1$.

Definition 2.5 (Decisional Linear)**.** The κ-decisional linear (κ-DLIN) assumption in the presence of a graded ℓ-linear map states that for any integers $a, b \geq \kappa$, and for any $\ell \leq j < \kappa$ the distributions

$$\left\{g, g^{\mathbf{X}}\right\}_{\mathbf{X} \leftarrow \mathbf{Rk}_j(\mathbb{Z}_p^{a \times b})} \quad \text{and} \quad \left\{g, g^{\mathbf{Y}}\right\}_{\mathbf{Y} \leftarrow \mathbf{Rk}_\kappa(\mathbb{Z}_p^{a \times b})}$$

are computationally indistinguishable, in the presence of $\vec{\mathbb{G}}$ and $\{\hat{e}_i\}_{i \in [1, \ell-1]}$.

The Multilinear Diffie-Hellman Exponent assumption. In Section 5, we will use the Multilinear Diffie-Hellman Exponent (MDHE) assumption, defined as follows. Consider a sequence of groups $\vec{\mathbb{G}} = (\mathbb{G}_1, \ldots, \mathbb{G}_\ell)$ with a set of bilinear maps $\hat{e}_i$ for $i \in [1, \ell - 1]$, and a generator g of $\mathbb{G}_1$.

Definition 2.6 (Multilinear Diffie-Hellman Exponent)**.** The d-Multilinear Diffie-Hellman Exponent (d-MDHE) assumption in the presence of a graded ℓ-linear map (as abstracted by [19]) states that, in the presence of $\vec{\mathbb{G}}$ and $\{\hat{e}_i\}_{i \in [1, \ell-1]}$, for any integer $j \geq \ell$, the distribution

$$\left\{g^{\mathbf{A}}, \left\langle g^{\mathbf{S}^i \cdot \mathbf{A}} \right\rangle_{i \in [d]}, g^{\mathbf{B}}, \left\langle g^{\mathbf{S}^i \cdot \mathbf{B}} \right\rangle_{i \in [d]} \right\}_{\mathbf{A}, \mathbf{B} \leftarrow \mathbf{Rk}_j(\mathbb{Z}_p^{j \times j}), \mathbf{S} \leftarrow \mathbb{Z}_p^{j \times j}}$$

is computationally indistinguishable from the distribution

$$\left\{g^{\mathbf{A}}, \left\langle g^{\mathbf{U}_i} \right\rangle_{i \in [d]}, g^{\mathbf{B}}, \left\langle g^{\mathbf{V}_i} \right\rangle_{i \in [d]} \right\}_{\mathbf{A}, \mathbf{B} \leftarrow \mathbf{Rk}_j(\mathbb{Z}_p^{j \times j}), \forall i \in [d], \mathbf{U}_i, \mathbf{V}_i \leftarrow \mathbb{Z}_p^{j \times j}}.$$

We note that the 1-MDHE assumption is essentially equivalent to the 2ℓ-DLIN assumption (where $j = \ell$ and $\kappa = 2\ell$ as in [13]), and hence the d-MDHE assumption can be seen as a generalization of DLIN assumption to the d^{th} exponent of the secret.

2.5 The Bellare-Cash Framework

Bellare and Cash [6] give a general framework (denoted the BC framework) for constructing RKA-secure PRFs for a class Φ using a key malleable PRF, a key fingerprint, and a collision-resistant hash function. We review their definitions and main theorem here.

Definition 2.7 (Key Malleable PRF)**.** A PRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is *key malleable* if there exists an efficient algorithm T, which on input $\phi \in \Phi$ and $x \in \mathcal{X}$ and with oracle access to $F(k, \cdot)$, which satisfies $\mathsf{T}^{F(k,\cdot)}(\phi, x) = F(\phi(k), x)$, for all $k \in \mathcal{K}$. Also, we require that for any distinct $x_1, \ldots, x_Q \in \mathcal{X}$, if $f : \mathcal{X} \to \mathcal{Y}$ is a truly random function, then $\mathsf{T}^{f(\cdot)}(\phi, x_1), \ldots, \mathsf{T}^{f(\cdot)}(\phi, x_Q)$ are distributed independently and uniformly in $\mathcal{Y}$.

Definition 2.8 (Key Fingerprint)**.** An element $w \in \mathcal{X}$ is a *key fingerprint* if for all $k \in \mathcal{K}$ and *distinct* $\phi_1, \phi_2 \in \Phi$, $F(\phi_1(k), w) \neq F(\phi_2(k), w)$.

Definition 2.9 (Compatible Hash Function)**.** For a fingerprint w, a hash function $H_{\mathsf{com}} : \mathcal{X} \times \mathcal{Y} \to \mathcal{R}$ is *compatible* if the set of oracle queries made by $\mathsf{T}^{F(k,\cdot)}(\phi, w)$ over all $\phi \in \Phi$ is disjoint from the set of oracle queries made by $\mathsf{T}^{F(k,\cdot)}(\phi, z)$ over all $z \in \mathcal{R}$ and $\phi \in \Phi$.

Theorem 2.10 (c.f. [6, Theorem 3.1], paraphrased). *For a fixed class Φ of related-key deriving functions, let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a key malleable PRF for Φ, $w \in \mathcal{X}$ a key fingerprint for F and Φ, and $H_{\mathsf{com}} : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$ a compatible hash function. Define $F_{\mathsf{rka}} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ as*

$$F_{\mathsf{rka}}(k, x) = F(k, H_{\mathsf{com}}(x, F(k, w))).$$

For any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ against the RKA PRF F_{rka} for the class Φ, there exist PPT adversaries $\mathcal{B}$ against the PRF security of F_{LWE} and $\mathcal{C}$ against the collision-resistance of the hash function H_{com} such that

$$\mathbf{Adv}^{\mathsf{prf\text{-}rka}}_{\Phi, F_{\mathsf{rka}}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{prf}}_{F}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{cr}}_{H_{\mathsf{com}}}(\mathcal{C}).$$

3 New RKA-secure PRFs Using the BC Framework

In this section, we use the BC framework [6] to construct new RKA-secure PRFs. We introduce two classes of related-key functions, a linear (Φ_{lin^*}) and an affine (Φ_{aff}) class, and show that the key homomorphic PRFs from Boneh *et al.* [13] can be used to instantiate the BC framework. The main technical challenge requires using the key homomorphism property to construct appropriate *transformers* required in the BC framework.

3.1 RKA-secure PRFs for a Restricted Linear Class Φ_{lin^*}

Boneh, Lewi, Montgomery, and Raghunathan [13] constructed the following PRF that is *almost* key homomorphic and showed its pseudorandomness under the LWE assumption.

The PRF F_{LWE}. For parameters m, p, and $q \in \mathbb{N}$ such that $p \mid q$, the public parameters of the PRF are binary matrices $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{Z}_p^{m \times m}$. The PRF key is a vector $\mathbf{k} \in \mathbb{Z}_q^m$. The PRF $F_{\mathrm{LWE}} : \mathbb{Z}_q^m \to \mathbb{Z}_p^m$ is defined as follows:

$$F_{\mathrm{LWE}}(\mathbf{k}, x) = \left\lfloor \prod_{i=1}^{\ell} \mathbf{A}_{x_i} \cdot \mathbf{k} \right\rfloor_p . \tag{3.1}$$

Theorem 3.1 (cf. [13], paraphrased). *The function F_{LWE} is pseudorandom under the LWE assumption for suitable choices of the parameters.*

The class Φ_{lin^*}. Recall the definition of $(\frac{q}{p})\mathbb{Z}_q$. We consider a class of linear RKA functions defined as follows:

$$\Phi_{\mathsf{lin}^*} = \{\phi_{\mathbf{a}} : \mathbb{Z}_q^m \to \mathbb{Z}_q^m \mid \phi_{\mathbf{a}}(\mathbf{k}) = \mathbf{k} + \mathbf{a}\}_{\mathbf{a} \in (\frac{q}{p})\mathbb{Z}_q^m}. \tag{3.2}$$

We use the homomorphic property of the PRF to construct a transformer, that we denote $\mathsf{T}_{\mathsf{lin}}^{f(\cdot)}$, in a straightforward manner: $\mathsf{T}_{\mathsf{lin}}^{f(\cdot)}(\phi_{\mathbf{a}}, x) := f(x) + F_{\mathrm{LWE}}(\mathbf{a}, x)$. To use the BC framework, it is necessary to show that for the class of RKA functions Φ_{lin^*}, the PRF and the transformer satisfy the malleability and uniformity properties.

Lemma 3.2 (Malleability). *For all $\mathbf{k} \in \mathbb{Z}_q^m$, $\phi \in \Phi_{\mathsf{lin}^*}$, and $x \in \{0,1\}^\ell$, it holds that*

$$\mathsf{T}_{\mathsf{lin}}^{F_{\mathrm{LWE}}(\mathbf{k},\cdot)}(\phi, x) = F_{\mathrm{LWE}}(\phi(\mathbf{k}), x). \tag{3.3}$$

Proof. Fix a key $\mathbf{k} \in \mathbb{Z}_q^m$ and $x \in \{0,1\}^\ell$. Let $\phi_{\mathbf{a}}$ denote a function in Φ_{lin^*} corresponding to $\mathbf{a} \in (\frac{q}{p})\mathbb{Z}_q^m$. Define the product of matrices $\mathbf{P} = \prod_{i=1}^{\ell} \mathbf{A}_{x_i}$. From the definition of the transformer $\mathsf{T}_{\mathsf{lin}}^{F_{\mathrm{LWE}}(\mathbf{k},\cdot)}$ the left side of equation (3.3) equals $\lfloor \mathbf{Pk} \rfloor_p + \lfloor \mathbf{Pa} \rfloor_p$. The right side of the equation is $\lfloor \mathbf{P}(\mathbf{k}+\mathbf{a}) \rfloor_p = \lfloor \mathbf{Pk} + \mathbf{Pa} \rfloor_p$. As $\mathbf{a} \in (\frac{q}{p})\mathbb{Z}_q^m$, it holds that $\mathbf{Pa} \in (\frac{q}{p})\mathbb{Z}_q^m$. Applying Lemma 2.1 on each coordinate, it holds that $\lfloor \mathbf{Pk} + \mathbf{Pa} \rfloor_p = \lfloor \mathbf{Pk} \rfloor_p + \lfloor \mathbf{Pa} \rfloor_p$, as required. ∎

The following lemma follows straightforwardly from the definition of $\mathsf{T}_{\mathsf{lin}}^{f(\cdot)}$.

Lemma 3.3 (Uniformity). *If $f : \{0,1\}^\ell \to \mathbb{Z}_p^m$ is a random function and $x_1, \ldots, x_Q \in \{0,1\}^\ell$ are distinct, for any functions $\phi_1, \ldots, \phi_Q \in \Phi_{\mathsf{lin}^*}$, the values $\mathsf{T}_{\mathsf{lin}}^{f(\cdot)}(\phi_i, x_i)$ are independently and uniformly distributed in $\mathbb{Z}_p^m$.*

Next, we show that any $w \in \{0,1\}^\ell$ is a key fingerprint for Φ_{lin^*}.

Lemma 3.4 (Fingerprint). *For any $w \in \{0,1\}^\ell$, $\mathbf{k} \in \mathbb{Z}_q^m$, for any distinct $\phi_1, \phi_2 \in \Phi_{\mathsf{lin}^*}$, it holds that $F_{\mathrm{LWE}}(\phi_1(\mathbf{k}), w) \neq F_{\mathrm{LWE}}(\phi_2(\mathbf{k}), w)$.*

Proof. For $i \in \{1,2\}$, let $\phi_i = \phi_{\mathbf{a}_i}$ for vectors $\mathbf{a}_i \in (\frac{q}{p})\mathbb{Z}_q^m$. Let $\mathbf{P} = \prod_{i=1}^{\ell} \mathbf{A}_{w_i}$, the product of full-rank matrices. As ϕ_1 and ϕ_2 are *distinct* and $\mathbf{P}$ is full-rank over $\mathbb{Z}_q$, it holds that $\mathbf{P}(\mathbf{a}_1 - \mathbf{a}_2) = \mathbf{u}$ for some *non-zero* $\mathbf{u}$. Moreover, as $\mathbf{a}_1$ and $\mathbf{a}_2$ are in $(\frac{q}{p})\mathbb{Z}_q^m$, the difference $(\mathbf{a}_1 - \mathbf{a}_2)$ and therefore $\mathbf{u}$ are in $(\frac{q}{p})\mathbb{Z}_q^m$. Now, note that $F_{\mathrm{LWE}}(\phi_1(\mathbf{k}), w) = \lfloor \mathbf{P} \cdot \mathbf{k} + \mathbf{P} \cdot \mathbf{a}_1 \rfloor_p = \lfloor \mathbf{P} \cdot \mathbf{k} + \mathbf{P} \cdot \mathbf{a}_2 + \mathbf{u} \rfloor_p$. Applying Lemma 2.1, this in turn equals $\lfloor \mathbf{P} \cdot \mathbf{k} + \mathbf{P} \cdot \mathbf{a}_2 \rfloor_p + \lfloor \mathbf{u} \rfloor_p = F_{\mathrm{LWE}}(\phi_2(\mathbf{k}), w) + \lfloor \mathbf{u} \rfloor_p$. As $\mathbf{u} \in (\frac{q}{p})\mathbb{Z}_q^m$ and is non-zero, $\lfloor \mathbf{u} \rfloor_p$ is also non-zero in $\mathbb{Z}_p^m$ concluding the proof of the lemma. ∎

Consider a collision-resistant hash function $H \colon \{0,1\}^\ell \times \mathbb{Z}_q^m \to \{0,1\}^{\ell-1}$ and the fingerprint $w = 0^\ell$. We define $H_{\mathsf{com}}^{(\Phi_{\mathsf{lin}^*})} \colon \{0,1\}^\ell \times \mathbb{Z}_q^m \to \{0,1\}^\ell$ as $H_{\mathsf{com}}^{(\Phi_{\mathsf{lin}^*})}(x,y) = 1 \| H(x,y)$ and note that it is a compatible hash function. Applying Lemmas 3.2–3.4 and Theorem 3.1 to the BC framework, Theorem 2.10 implies the following result.

Theorem 3.5. *Under the LWE assumption and the collision-resistance of the hash function H, the function $F_{\mathsf{rka\text{-}lin}} \colon \mathbb{Z}_q^m \times \{0,1\}^\ell \to \mathbb{Z}_p^m$ defined as:*

$$F_{\mathsf{rka\text{-}lin}}(\mathbf{k}, x) = F_{\mathrm{LWE}}\left(\mathbf{k}, H_{\mathsf{com}}^{(\Phi_{\mathsf{lin}^*})}\left(x, F_{\mathrm{LWE}}\left(\mathbf{k}, 0^\ell\right)\right)\right)$$

is an RKA-secure PRF with respect to Φ_{lin^*}.

3.2 RKA-secure PRFs for an Affine Class Φ_{aff}

In addition to the LWE-based almost key homomorphic PRF, Boneh *et al.* [13] also constructed a "fully" homomorphic PRF under the DLIN assumption over groups equipped with a multilinear map.

The PRF F_{DLIN}. For parameters m and $\ell \in \mathbb{N}$, let $\vec{\mathbb{G}} = (\mathbb{G}_1, \ldots, \mathbb{G}_\ell)$ be a sequence of groups equipped with a graded ℓ-multilinear map $\{\hat{e}_i\}_{i \in [\ell-1]}$. The public parameters comprise $pp = \left(g^{\mathbf{A}_0}, g^{\mathbf{A}_1}\right)$, where $\mathbf{A}_0, \mathbf{A}_1 \leftarrow \mathbf{Rk}_\ell\left(\mathbb{Z}_p^{\ell \times \ell}\right)$. The PRF key $\mathbf{K}$ is a matrix in $\mathbb{Z}_p^{m \times \ell}$. Define $F_{\mathrm{DLIN}} \colon \mathbb{Z}_p^{m \times \ell} \times \{0,1\}^\ell \to (\mathbb{G}_\ell)^{m \times \ell}$ as follows:

$$F_{\mathrm{DLIN}}(\mathbf{K}, x) = (g_\ell)^{\mathbf{W}}, \text{ where } \mathbf{W} = \mathbf{K} \cdot \left(\prod_{i=1}^{\ell} \mathbf{A}_{x_i}\right). \tag{3.4}$$

Theorem 3.6 (cf. [13], paraphrased)**.** *The function F_{DLIN} is pseudorandom under the DLIN assumption for suitable choices of parameters.*

As noted by Boneh *et al.*, the PRF can be evaluated at a point $x = x_1 \dots x_\ell \in \{0,1\}^\ell$ given the the public parameters *pp* and secret key $\mathbf{k} \in \mathbb{Z}_p^\ell$ using the graded bilinear maps $\hat{e}_i : \mathbb{G}_1 \times \mathbb{G}_i \to \mathbb{G}_{i+1}$. The matrix multiplication is carried out one step at a time by nesting these bilinear maps as follows:

$$F_{\text{DLIN}}(\mathbf{K}, x) = \hat{e}_{\ell-1}\left(g^{\mathbf{K}\mathbf{A}_{x_1}},\ \hat{e}_{\ell-2}\left(g^{\mathbf{A}_{x_2}},\ \dots \hat{e}_2\left(g^{\mathbf{A}_{x_{\ell-2}}},\ \hat{e}_1\left(g^{\mathbf{A}_{x_{\ell-1}}},\ g^{\mathbf{A}_{x_\ell}}\right)\right)\right)\right),$$

where $g^{\mathbf{K}\mathbf{A}_{x_1}}$ is computed "in the exponent" given $\mathbf{K}$ and $g^{\mathbf{A}_{x_1}}$. A pairing $\hat{e}\left(g^{\mathbf{A}_0}, g^{\mathbf{A}_1}\right)$ of matrices given in the exponent is done by computing the component-wise dot products of rows of $\mathbf{A}_0$ with columns of $\mathbf{A}_1$ using the bilinear map $\hat{e}$.

Observe that this PRF is identical to the DLIN-based PRF in [13] except that the key $\mathbf{K}$ is now a matrix. This is required to define a meaningful affine class over the key space. The pseudorandomness extends to the case where $\mathbf{K}$ is a matrix by considering the rows of $\mathbf{K}$, $\mathbf{k}_1{}^\intercal, \dots, \mathbf{k}_m{}^\intercal$ to be m independent keys of the original DLIN-based PRF. The key homomorphism also extends in a straightforward manner.

The affine class Φ_{aff}. With the above DLIN-based PRF, we can consider the following affine class of related-key deriving functions. We define

$$\Phi_{\mathsf{aff}} = \{\phi_{\mathbf{C},\mathbf{B}} : \mathbb{Z}_p^{m\times\ell} \to \mathbb{Z}_p^{m\times\ell} \mid \phi_{\mathbf{C},\mathbf{B}}(\mathbf{K}) = \mathbf{C}\mathbf{K} + \mathbf{B}\}, \tag{3.5}$$

for matrices $\mathbf{C} \in \mathbb{Z}_p^{m\times m}$ and $\mathbf{B} \in \mathbb{Z}_p^{m\times\ell}$ constrained as follows: (a) the class Φ_{aff} is *claw-free*, and (b) $\mathbf{C}$ is a *full-rank* matrix.

As in Section 3.1, the key homomorphism of F_{DLIN} allows us to construct a transformer, denoted $\mathsf{T}_{\mathsf{aff}}^{f(\cdot)}$, in the following manner: $\mathsf{T}_{\mathsf{aff}}^{f(\cdot)}(\phi_{\mathbf{C},\mathbf{B}}, x)$ sets $f(x) = (g_\ell)^{\mathbf{F}}$ and computes $(g_\ell)^{\mathbf{CF}} \cdot F_{\text{DLIN}}(\mathbf{B}, x)$. In other words, we left-multiply (in the exponent) the output of $f(\cdot)$ with entries from $\mathbf{C}$ and then use the homomorphism of F_{DLIN} to incorporate $\mathbf{B}$. We use the BC framework and show that for the class of related-key functions Φ_{aff}, the PRF and the transformer satisfy the malleability and uniformity properties.

Lemma 3.7 (Malleability)**.** *For all $\mathbf{K} \in \mathbb{Z}_p^{m\times\ell}$, $\phi \in \Phi_{\mathsf{aff}}$, and $x \in \{0,1\}^\ell$, it holds that*

$$\mathsf{T}_{\mathsf{aff}}^{f(\cdot)}(\phi, x) = F_{\text{DLIN}}(\phi(\mathbf{k}), x). \tag{3.6}$$

Proof. The proof follows from elementary algebra in the exponent. Let $\phi = \phi_{\mathbf{C},\mathbf{B}}$ for arbitrary $\mathbf{C}$ and $\mathbf{B}$. For a key $\mathbf{K}$ and input x, let $\mathbf{W}$ be the matrix in equation (3.4). By definition, $\mathsf{T}_{\mathsf{aff}}^{f(\cdot)}(\phi, x) = (g_\ell)^{\mathbf{C}\cdot\mathbf{W}} \cdot F_{\text{DLIN}}(\mathbf{B}, x) = F_{\text{DLIN}}(\mathbf{CK} + \mathbf{B}, x)$ as required. The last equality follows from the key homomorphism of F_{DLIN}. ∎

The following lemma follows straightforwardly from the definition of $\mathsf{T}_{\mathsf{aff}}^{f(\cdot)}$.

Lemma 3.8 (Uniformity)**.** *If $f : \{0,1\}^\ell \to (\mathbb{G}_\ell)^{m\times\ell}$ is a random function and $x_1, \dots, x_Q \in \{0,1\}^\ell$ are distinct, for any functions $\phi_1, \dots, \phi_Q \in \Phi_{\mathsf{aff}}$, the values $\mathsf{T}_{\mathsf{aff}}^{f(\cdot)}(\phi_i, x_i)$ are independently and uniformly distributed in $(\mathbb{G}_\ell)^{m\times\ell}$.*

Next, we show that any $w \in \{0,1\}^\ell$ is a key fingerprint for Φ_{lin^*}.

Lemma 3.9 (Fingerprint). *For any $w \in \{0,1\}^\ell$, for any $\mathbf{K} \in \mathbb{Z}_q^{m\times\ell}$, and for any two distinct $\phi_1, \phi_2 \in \Phi_{\mathsf{aff}}$, it holds that $F_{\text{DLIN}}(\phi_1(\mathbf{K}), w) \neq F_{\text{DLIN}}(\phi_2(\mathbf{K}), w)$.*

Proof. We use the fact that the family Φ_{aff} is claw-free. For any key $\mathbf{K}$, this implies that $\phi_1(\mathbf{K}) \neq \phi_2(\mathbf{K})$. For $i \in \{1,2\}$, let $\mathbf{W}_i$ denote the matrix $\phi_i(\mathbf{K}) \cdot \left(\prod_{i=1}^{\ell} \mathbf{A}_{w_i}\right)$. The product of full-rank matrices $\mathbf{A}_{w_i}$ is full-rank and as $\phi_1(\mathbf{K}) \neq \phi_2(\mathbf{K})$, it follows that $\mathbf{W}_1 \neq \mathbf{W}_2$. As F_{DLIN} is defined as $(g_\ell)^{\mathbf{W}}$ for generator g_ℓ, it holds that if $\mathbf{W}_1 \neq \mathbf{W}_2$, then $(g_\ell)^{\mathbf{W}_1} \neq (g_\ell)^{\mathbf{W}_2}$ concluding the proof of the lemma. ∎

Consider a collision-resistant hash function $H\colon \{0,1\}^\ell \times (\mathbb{G}_\ell)^{m\times\ell} \to \{0,1\}^{\ell-1}$ and the fingerprint $w = 0^\ell$. We define $H_{\mathsf{com}}^{(\Phi_{\mathsf{aff}})}\colon \{0,1\}^\ell \times (\mathbb{G}_\ell)^{m\times\ell} \to \{0,1\}^\ell$ as $H_{\mathsf{com}}^{(\Phi_{\mathsf{aff}})}(x,y) = 1\|H(x,y)$ and note that it is a compatible hash function. Applying Lemmas 3.7–3.9 and Theorem 3.6 to the BC framework, Theorem 2.10 implies the following result.

Theorem 3.10. *Under the DLIN assumption and the collision-resistance of the hash function H, the function $F_{\mathsf{rka\text{-}aff}}\colon \mathbb{Z}_p^{m\times\ell} \times \{0,1\}^\ell \to (\mathbb{G}_\ell)^{m\times\ell}$ defined as:*

$$F_{\mathsf{rka\text{-}aff}}(\mathbf{K}, x) = F_{\text{DLIN}}\left(\mathbf{K}, H_{\mathsf{com}}^{(\Phi_{\mathsf{aff}})}\left(x, F_{\text{DLIN}}\left(\mathbf{K}, 0^\ell\right)\right)\right)$$

is an RKA-secure PRF with respect to Φ_{aff}.

4 Unique-Input RKA-secure PRFs for an Affine Class

In this section, we construct RKA-secure PRFs from the LWE assumption for a slightly more restricted notion of RKA security, denoted unique-input RKA security. As explained in Section 1.3, we work directly with the pseudorandomness proof of F_{LWE} to show unique-input RKA security against a larger class of affine related-key functions rather than the restricted linear class Φ_{lin^*} from Section 3.1. To do this, we use the algebraic structure that suits the key homomorphism of F_{LWE} to overcome the restrictions of Φ_{lin^*} required in order to apply the Bellare-Cash framework. We prove unique-input RKA security for the affine class $\Phi_{\mathsf{ln\text{-}aff}} = \{\phi_{\mathbf{C},\mathbf{B}} : \phi_{\mathbf{C},\mathbf{B}}(\mathbf{K}) = \mathbf{C}\mathbf{K} + \mathbf{B}\}$, where $\mathbf{C}$ is a full rank matrix in $[-c,c]^{m\times m}$ for a small constant c, and $\mathbf{B}$ is an arbitrary matrix in $\mathbb{Z}_q^{m\times m}$.

We consider the PRF F_{LWE} where the key $\mathbf{k}$, originally a vector, is replaced by a matrix $\mathbf{K}$ in order to obtain the algebraic structure required for $\Phi_{\mathsf{ln\text{-}aff}}$. Recollect the definition of F_{LWE} from Equation (3.1). For parameters $m, p, q \in \mathbb{N}$ such that $p \,|\, q$, the public parameters of the PRF are binary matrices $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{Z}_p^{m\times m}$. The key is now a *matrix* $\mathbf{K} \in \mathbb{Z}_q^{m\times m}$, and the PRF $F_{\text{LWE}}\colon \mathbb{Z}_q^{m\times m} \times \{0,1\}^\ell \to \mathbb{Z}_p^{m\times m}$ is defined as follows:

$$F_{\text{LWE}}(\mathbf{K}, x) = \left\lfloor \mathbf{K} \cdot \prod_{i=1}^{\ell} \mathbf{A}_{x_i} \right\rfloor_p . \tag{4.1}$$

Recollect the bound B for samples drawn from the LWE error distribution $\overline{\Psi}_\alpha$. In the rest of the section, we set the parameters of the system $q, p, m, c, B, \lambda, \ell > 0$ such that the quantity $(2m)^\ell cBp/q$ is negligible in the security parameter λ. This is along the lines of the parameters chosen in [13]. We state the following theorem for this choice of parameters:

Theorem 4.1. *Under the LWE assumption, the PRF F_{LWE} defined in Equation* (4.1) *is RKA-secure against unique-input adversaries for the class $\Phi_{\mathsf{ln\text{-}aff}}$.*

Proof of Theorem 4.1. In what follows, for a bit string x on ℓ bits, we use $x|_j$ to denote the bit string comprising bits j through ℓ of x. Let $x|_{\ell+1}$ denote the empty string ε^*. Let $\mathcal{A}$ be a probabilistic polynomial time unique-input RKA adversary. We consider the following experiments interacting with $\mathcal{A}$.

Experiment G_j for $j \in [1, \ell+1]$.

1. The challenger samples as public parameters full-rank matrices $\mathbf{A}_0, \mathbf{A}_1 \in \{0,1\}^{m\times m} \subset \mathbb{Z}_q^{m\times m}$ which are sent to the adversary.
2. The challenger creates a lookup table L of pairs $(w, \mathbf{Z}) \in \{0,1\}^{\ell-j+1} \times \mathbb{Z}_q^{m\times m}$, and initializes L to contain only the pair $(\varepsilon^*, \mathbf{R})$ for some randomly chosen $\mathbf{R} \in \mathbb{Z}_q^{m\times m}$.
3. For $k \in [Q]$, the adversary (adaptively) sends input queries $\left(\phi^{(k)}_{\mathbf{C},\mathbf{B}}, x^{(k)}\right) \in \Phi_{\mathsf{ln\text{-}aff}} \times \{0,1\}^\ell$ to the challenger. For each input query, the challenger checks to see if there is a pair $\left(x^{(k)}|_j, \mathbf{Z}\right)$ in L for some $\mathbf{Z} \in \mathbb{Z}_q^{m\times m}$. If there is no such pair, then the challenger chooses a random $\mathbf{Y} \in \mathbb{Z}_q^{m\times m}$, adds the pair $\left(x^{(k)}|_j, \mathbf{Y}\right)$ to L, and sets $\mathbf{Z} = \mathbf{Y}$. The challenger returns $\mathbf{N} = \left\lfloor \mathbf{CZ}\prod_{i=1}^{j-1}\mathbf{A}_{x_i^{(k)}} + \mathbf{B}\prod_{i=1}^{\ell}\mathbf{A}_{x_i^{(k)}} \right\rfloor_p$ to the adversary.
4. The adversary outputs a bit $b' \in \{0,1\}$, which the experiment also outputs.

Experiment H_j for $j \in [1, \ell+1]$.

1. The challenger samples as public parameters full-rank matrices $\mathbf{A}_0, \mathbf{A}_1 \in \{0,1\}^{m\times m} \subset \mathbb{Z}_q^{m\times m}$ which are sent to the adversary.
2. The challenger creates a lookup table L of triples $(w, \mathbf{Y}, \mathbf{Z}) \in \{0,1\}^{\ell-j+1} \times \mathbb{Z}_q^{m\times m} \times \mathbb{Z}_q^{m\times m}$, and initializes L to contain only the triple $(\varepsilon^*, \mathbf{R}, \mathbf{\Delta})$ for some randomly chosen $\mathbf{R} \in \mathbb{Z}_q^{m\times m}$ and $\mathbf{\Delta} \leftarrow \overline{\Psi}_\alpha^{m\times m}$.
3. For $k \in [Q]$, the adversary (adaptively) sends input queries $\left(\phi^{(k)}_{\mathbf{C},\mathbf{B}}, x^{(k)}\right) \in \Phi_{\mathsf{ln\text{-}aff}} \times \{0,1\}^\ell$ to the challenger. For each input query, the challenger checks to see if there is a triple $(x^{(k)}|_{j-1}, \mathbf{Z}, \mathbf{\Delta})$ in L for some $\mathbf{Z} \in \mathbb{Z}_q^m$ and $\mathbf{\Delta} \leftarrow \overline{\Psi}_\alpha^{m\times m}$. If there is no such triple, then the challenger chooses a random $\mathbf{Y} \in \mathbb{Z}_q^{m\times m}$ and random $\mathbf{V}_0, \mathbf{V}_1 \leftarrow \overline{\Psi}_\alpha^{m\times m}$, adds the triples $\left(0 \parallel \left(x^{(k)}|_j\right), \mathbf{Y}, \mathbf{V}_0\right)$ and $\left(1 \parallel \left(x^{(k)}|_j\right), \mathbf{Y}, \mathbf{V}_1\right)$ to L, and sets $\mathbf{Z} = \mathbf{Y}$ and $\mathbf{\Delta} = \mathbf{V}_{x^{(k)}_{j-1}}$ (i.e., $\mathbf{V}_0$ or $\mathbf{V}_1$ depending on the $j - 1^{\text{th}}$ bit of $x^{(k)}$). The challenger returns to the adversary the value:
$$\mathbf{N} = \left\lfloor \mathbf{C}\left(\mathbf{ZA}_{x^{(k)}_{j-1}} + \mathbf{\Delta}\right) \cdot \prod_{i=1}^{j-2}\mathbf{A}_{x_i^{(k)}} + \mathbf{B} \cdot \prod_{i=1}^{\ell}\mathbf{A}_{x_i^{(k)}} \right\rfloor_p .$$
4. The adversary outputs a bit $b' \in \{0,1\}$, which the experiment also outputs.

Observe that $\mathsf{G}_{\ell+1}$ responds to the adversary's queries identically as in $\mathsf{Expt}_0^{\mathsf{prf\text{-}rka}}$. Hence, $\Pr\left[\mathsf{Expt}_0^{\mathsf{prf\text{-}rka}} = 1\right] = \Pr[\mathsf{G}_{\ell+1} = 1]$.

Lemma 4.2. *For all $j \in [2, \ell+1]$, it holds that $|\Pr[\mathsf{G}_j = 1] - \Pr[\mathsf{H}_j = 1]|$ is negligible.*

Proof. In Experiment H_j, let $\mathbf{M}_k = \mathbf{CZA}_{x^{(k)}_{j-1}} \cdot \prod_{i=1}^{j-2}\mathbf{A}_{x_i^{(k)}}$ and $\mathbf{W}_k = \mathbf{C\Delta} \cdot \prod_{i=1}^{j-2}\mathbf{A}_{x_i^{(k)}}$. Since the entries of $\mathbf{C}$ lie within $[-c, c]$, the entries of $\mathbf{\Delta}$ lie within $[-B, B]$, and the entries of each of the $j-2$ matrices $A_{x_i^{(k)}}$ lie within $\{0,1\}$, the entries of $\mathbf{W}_k$ must lie within $[-cBm^{j-2}, cBm^{j-2}]$.[4] Since $\mathbf{A}_0$ and $\mathbf{A}_1$ are full rank, the product of these matrices is also full rank. Since $\mathbf{Z}$ is drawn uniformly at random from $\mathbb{Z}_q^{m\times m}$, the matrix $\mathbf{M}_k$ is distributed uniformly in $\mathbb{Z}_q^{m\times m}$. Thus, the probability that $\lfloor \mathbf{M}_k + \mathbf{W}_k \rfloor_p \neq \lfloor \mathbf{M}_k \rfloor_p$ is at most $m^2(cBm^{j-2})p/q$. By taking a union bound over all $x \in \{0,1\}^\ell$, we have that the probability that there exists some input $x \in \{0,1\}^\ell$ for which $\lfloor \mathbf{M}_k + \mathbf{W}_k \rfloor_p \neq \lfloor \mathbf{M}_k \rfloor_p$ is at most $(2m)^\ell cBp/q$. Conditioned on the above event not occurring, it holds that for all x, $\lfloor \mathbf{M}_k + \mathbf{W}_k \rfloor_p = \lfloor \mathbf{M}_k \rfloor_p$ which implies that G_j and H_j respond identically to adversary queries. Therefore $|\Pr[\mathsf{G}_j = 1] - \Pr[\mathsf{H}_j = 1]|$ is bounded by the probability of the above "bad" event, which is negligible for a suitable choice of parameters. ∎

Lemma 4.3. *Under the LWE assumption, for all* $j \in [2, \ell+1]$, *it holds that the quantity* $|\Pr[\mathsf{G}_{j-1} = 1] - \Pr[\mathsf{H}_j = 1]|$ *is negligible.*

Proof. Recollect the definition of the low-norm matrix LWE problem defined in Section 2.4. We construct a simulator Sim against a low-norm matrix LWE challenger which gives Q LWE challenges rather than just one. Note that a standard hybrid argument can be used to show that an adversary against an LWE challenger which gives Q LWE challenges can be used to construct an adversary against an LWE challenger for only a single LWE challenge, with a $1/Q$ loss in advantage. When the LWE challenger gives "real" challenges to the simulator, Sim behaves as a challenger for Experiment H_j, and when the LWE challenger gives "random" challenges, Sim behaves as a challenger for Experiment G_j.

The simulator Sim queries the LWE challenger to receive Q samples, each of the form $(\mathbf{U}, \mathbf{V}_i) \in \mathbb{Z}_q^{2m\times m} \times \mathbb{Z}_q^{2m\times m}$ for $i \in [Q]$. We will refer to $\mathbf{U}^{(0)}, \mathbf{V}_i^{(0)} \in \mathbb{Z}_q^{m\times m}$ as the first m rows of $\mathbf{U}$ and $\mathbf{V}_i$, and $\mathbf{U}^{(1)}, \mathbf{V}_i^{(1)} \in \mathbb{Z}_q^{m\times m}$ as the last m rows of $\mathbf{U}$ and $\mathbf{V}_i$, respectively. Sim creates two lists of matrices $\mathsf{List}^{(0)}, \mathsf{List}^{(1)} \in (\mathbb{Z}_q^{m\times m})^Q$ such that $\mathsf{List}^{(0)} = \left\langle \mathbf{V}_i^{(0)} \right\rangle_{i\in[1,Q]}$ and $\mathsf{List}^{(1)} = \left\langle \mathbf{V}_i^{(1)} \right\rangle_{i\in[1,Q]}$. Sim then sets $\mathbf{A}_i = \mathbf{U}^{(i)}$ for $i \in \{0,1\}$. Sim then creates a lookup table of pairs $\mathsf{L} : \{0,1\}^{\ell-j} \times \mathbb{Z}_q^{m\times m}$, initializing the table to contain the pair $(\varepsilon^*, \mathbf{R})$ for a randomly chosen $\mathbf{R} \leftarrow \mathbb{Z}_q^{m\times m}$. Sim also keeps a counter $\mathsf{k} \in \mathbb{Z}$, initialized to 1. Sim sends $pp = (\mathbf{A}_0, \mathbf{A}_1)$ to the adversary.

Now, when the adversary $\mathcal{A}$ makes a query $(\phi_{\mathbf{C},\mathbf{B}}, \hat{x}) \in \Phi_{\mathsf{ln\text{-}aff}} \times \{0,1\}^\ell$, Sim first checks if the pair $(\hat{x}|_{j-1}, \mathbf{Z})$ exists in L, for some $\mathbf{Z} \in \mathbb{Z}_q^{m\times m}$. If not, he adds the pairs $\left(0 \parallel (\hat{x}|_j), \mathsf{List}_\mathsf{k}^{(0)}\right)$ and $\left(1 \parallel (\hat{x}|_j), \mathsf{List}_\mathsf{k}^{(1)}\right)$ to the table L, and sets $\mathbf{Z} = \mathsf{List}_\mathsf{k}^{(\hat{x}_{j-1})}$, and increments k by 1. Then, Sim responds to the adversary's query by returning $\left\lfloor \mathbf{CZ} \prod_{i=1}^{j-2} \mathbf{A}_{\hat{x}_i} + \mathbf{B} \prod_{i=1}^{\ell} \mathbf{A}_{\hat{x}_i} \right\rfloor_p$. Finally, when $\mathcal{A}$ outputs a bit b', Sim also outputs b'. Note that the counter k will never exceed Q, since $\mathcal{A}$ makes at most Q queries, and therefore the simulation is well-defined.

If the LWE challenges are of the form $(\mathbf{U}, \mathbf{R}_i)$ for each $i \in [Q]$, then each $\mathbf{Z}$ is distributed uniformly and independently across queries which differ on bits $j-1$ through ℓ, which means that Sim responds to queries x with $\left\lfloor \mathbf{CZ} \prod_{i=1}^{j-2} \mathbf{A}_{x_i} + \mathbf{B} \prod_{i=1}^{\ell} \mathbf{A}_{x_i} \right\rfloor_p$, and therefore Sim has simulated G_{j-1}. If instead the LWE challenges are of the form $(\mathbf{U}, \mathbf{K}_i\mathbf{U} + \mathbf{\Delta}_i)$ for each $i \in [Q]$, then $\mathbf{Z}$ is of the form $\mathbf{K}\mathbf{A}_{x_{j-2}} + \mathbf{\Delta}$ for each query, which means that Sim responds to queries x with

[4] The fact that entries of $\mathbf{\Delta}$ lie within $[-B, B]$ holds only with overwhelming probability, but we will ignore this detail for ease of presentation, as it does not affect the final theorem.

$\left\lfloor \mathbf{C}\left(\mathbf{K}\mathbf{A}_{x^{(k)}_{j-1}} + \mathbf{\Delta}\right)\prod_{i=1}^{j-2}\mathbf{A}_{x^{(k)}_i} + \mathbf{B}\prod_{i=1}^{\ell}\mathbf{A}_{x_i}\right\rfloor_p$, and therefore Sim has simulated H_j. Under the LWE assumption, the claim follows. ∎

Lemma 4.4. $\Pr[\mathsf{G}_1 = 1] = \Pr\left[\mathsf{Expt}_1^{\mathsf{prf\text{-}rka}} = 1\right]$.

Proof. Recall that in G_1, on query $(\phi_{\mathbf{C},\mathbf{B}}, x) \in \Phi_{\mathsf{ln\text{-}aff}} \times \{0,1\}^\ell$, the challenger responds with $\mathbf{N} = \left\lfloor \mathbf{C}\mathbf{Z} + \mathbf{B}\prod_{i=1}^{\ell}\mathbf{A}_{x_i}\right\rfloor_p$, where each $\mathbf{Z}$ is uniformly and independently distributed for distinct input queries x. Since $\mathbf{C}$ is full rank and $\mathbf{B}\prod_{i=1}^{\ell}\mathbf{A}_{x_i}$ is independent of $\mathbf{Z}$, it follows that $\mathbf{N}$ is distributed as a uniform element in $\mathbb{Z}_p^{m\times m}$, independently for each input query $x \in \{0,1\}^\ell$. ∎

Applying Lemmas 4.2–4.4 yields Theorem 4.1.

5 Unique-Input RKA-secure PRFs for a Class of Polynomials

Recall the definition of the PRF F_{DLIN} from Section 3.2 and the definition of the d-MDHE assumption from Section 2.4. In this section, under the d-MDHE assumption, we show that F_{DLIN} is RKA-secure against unique-input adversaries with respect to the following class of bounded-degree polynomials. For positive integers ℓ, d and prime p we define

$$\Phi_{\mathsf{poly}(d)} = \left\{\phi_{P(\cdot)} : \mathbb{Z}_p^{\ell\times\ell} \to \mathbb{Z}_p^{\ell\times\ell} \mid \phi_{P(\cdot)}(\mathbf{K}) = P(\mathbf{K})\right\},$$

for polynomials P over $\mathbb{Z}_p^{\ell\times\ell}$ of degree at most d which have at least one coefficient matrix (excluding the constant coefficient matrix) which is full rank. In other words, if $P(\mathbf{K}) = \sum_{i=0}^{d}\mathbf{C}_i \cdot \mathbf{K}^i$ for matrices $\mathbf{C}_i \in \mathbb{Z}_p^{\ell\times\ell}$, then there exists a $j > 0$ such that $\mathbf{C}_j \in \mathbf{Rk}_\ell\left(\mathbb{Z}_p^{\ell\times\ell}\right)$. .

Theorem 5.1. *Under the d-MDHE assumption, the PRF F_{DLIN} is RKA-secure against unique-input adversaries for the class $\Phi_{\mathsf{poly}(d)}$.*

Proof of Theorem 5.1. For a bit string x on ℓ bits, we use $x|_j$ to denote the bit string comprising bits j through ℓ of x, and let $x|_{\ell+1}$ denote the empty string ε^*. Let $\mathcal{A}$ be a probabilistic polynomial time unique-input RKA adversary. We consider the following experiments interacting with $\mathcal{A}$.

Experiment Expt_j for $j \in [1, \ell]$.

1. The challenger samples public parameters $\mathbf{A}_0, \mathbf{A}_1 \leftarrow \mathbf{Rk}_\ell(\mathbb{Z}_p^{\ell\times\ell})$. Then the challenger sends $pp = \left(g, g^{\mathbf{A}_0}, g^{\mathbf{A}_1}\right)$ to the adversary.
2. The challenger creates a lookup table L of pairs $\left(w, \langle\mathbf{Z}_i\rangle_{i\in[d]}\right) \in \{0,1\}^{\ell-j+1} \times \mathbb{Z}_p^{\ell\times\ell}$, and initializes L to contain only the pair $\left(\varepsilon^*, \langle\mathbf{R}_i\rangle_{i\in[d]}\right)$ for some randomly chosen vector of matrices $\langle\mathbf{R}_i\rangle_{i\in[d]} \in \left(\mathbb{Z}_p^{\ell\times\ell}\right)^d$.
3. For $k \in [1, Q]$, the adversary (adaptively) sends input queries $\left(\phi^{(k)}_{P(\cdot)}, x^{(k)}\right) \in \Phi_{\mathsf{poly}(d)} \times \{0,1\}^\ell$ to the challenger. For $i \in [0, d]$, let $\mathbf{C}_i \in \mathbb{Z}_p^{\ell\times\ell}$ be the coefficients of P, so that for all $\mathbf{M} \in \mathbb{Z}_p^{\ell\times\ell}$, $P(\mathbf{M}) = \sum_{i=0}^{d}\mathbf{C}_i \cdot \mathbf{M}^i$. For each input query, the challenger checks to see if there is a pair $\left(x^{(k)}|_j, \langle\mathbf{Z}_i\rangle_{i\in[d]}\right)$ in L for a list of matrices $\langle\mathbf{Z}_i\rangle_{i\in[d]} \in \left(\mathbb{Z}_p^{\ell\times\ell}\right)^d$. If there is no

such pair, then the challenger chooses a random list of matrices $\langle \mathbf{Y}_i \rangle_{i\in[d]} \in \left(\mathbb{Z}_p^{\ell\times\ell}\right)^d$, adds the pair $\left(x^{(k)}|_j, \langle \mathbf{Y}_i \rangle_{i\in[d]}\right)$ to L, and sets $\mathbf{Z}_i = \mathbf{Y}_i$ for all $i \in [d]$. The challenger computes $\mathbf{N} = \mathbf{C}_0 \cdot \prod_{i=1}^{\ell} \mathbf{A}_{x_i^{(k)}} + \left(\prod_{i=1}^{d} \mathbf{C}_i \cdot \mathbf{Z}_i\right) \cdot \left(\prod_{i=1}^{j-1} \mathbf{A}_{x_i^{(k)}}\right)$ and returns $g_\ell^{\mathbf{N}} \in (\mathbb{G}_\ell)^{\ell\times\ell}$ to the adversary.

4. Eventually the adversary outputs a bit $b' \in \{0,1\}$, which the experiment also outputs.

Lemma 5.2. *Under the d-MDHE assumption,* $\left|\Pr[\mathsf{Expt}_\ell = 1] - \Pr\left[\mathsf{Expt}_0^{\mathsf{prf\text{-}rka}} = 1\right]\right|$ *is negligible.*

Proof. We construct a simulator Sim against a MDHE challenger such that when the MDHE challenger gives "real" challenges to the simulator, Sim behaves as a challenger for Experiment $\mathsf{Expt}_0^{\mathsf{prf\text{-}rka}}$, and when the MDHE challenger gives "random" challenges, Sim behaves as a challenger for Experiment Expt_ℓ.

The simulator Sim queries the MDHE challenger for a sample $\left(g^{\mathbf{U}^*}, \langle g^{\mathbf{U}_i}\rangle_{i\in[d]}, g^{\mathbf{V}^*}, \langle g^{\mathbf{V}_i}\rangle_{i\in[d]}\right)$, where $\mathbf{U}^*, \mathbf{V}^* \in \mathbf{Rk}_\ell(\mathbb{Z}_p^{\ell\times\ell})$ and $\mathbf{U}_i, \mathbf{V}_i \in \mathbb{Z}_p^{\ell\times\ell}$ for each $i \in [d]$. The simulator must decide whether there exists some $\mathbf{S} \in \mathbb{Z}_p^{\ell\times\ell}$ such that $\mathbf{U}_i = \mathbf{S}^i \cdot \mathbf{U}^*$ and $\mathbf{V}_i = \mathbf{S}^i \cdot \mathbf{V}^*$ for all $i \in [d]$, or whether all $2d+2$ matrices in the set $\left\{\mathbf{U}^*, \mathbf{V}^*, \langle \mathbf{U}_i\rangle_{i\in[d]}, \langle \mathbf{V}_i\rangle_{i\in[d]}\right\}$ are distributed uniformly and independently.

The simulator embeds the MDHE challenge by setting $\mathbf{A}_0 = \mathbf{U}^*$ and $\mathbf{A}_1 = \mathbf{V}^*$. The simulator then creates a lookup table of pairs $\mathsf{L} : \{0,1\} \times (\mathbb{G}_1)^{\ell\times\ell}$, initializing the table to contain the pair $\left(\varepsilon^*, \langle g^{\mathbf{R}_i}\rangle_{i\in[d]}\right)$ for a randomly chosen list of matrices $\langle \mathbf{R}_i\rangle_{i\in[d]} \leftarrow \left(\mathbb{Z}_p^{\ell\times\ell}\right)^d$. Sim sends $pp = \left(g, g^{\mathbf{A}_0}, g^{\mathbf{A}_1}\right)$ to the adversary.

Now, when the adversary $\mathcal{A}$ makes a query $(\phi_{P(\cdot)}, \hat{x}) \in \Phi_{\mathsf{poly}(d)} \times \{0,1\}^\ell$, for each $i \in [0,d]$, let $\mathbf{C}_i \in \mathbb{Z}_p^{\ell\times\ell}$ be the coefficients of P, so that for all $\mathbf{M} \in \mathbb{Z}_p^{\ell\times\ell}$, $P(\mathbf{M}) = \sum_{i=0}^{d} \mathbf{C}_i \cdot \mathbf{M}^i$. Sim first checks if the pair $\left(\hat{x}_\ell, \langle g^{\mathbf{Z}_i}\rangle_{i\in[d]}\right)$ exists in L, for some list of matrices $\langle \mathbf{Z}_i\rangle_{i\in[d]} \in \left(\mathbb{Z}_q^{\ell\times\ell}\right)^d$. If not, he adds the pairs $\left(0, \langle g^{\mathbf{U}_i}\rangle_{i\in[d]}\right)$ and $\left(1, \langle g^{\mathbf{V}_i}\rangle_{i\in[d]}\right)$ to the table L, and for each $i \in [d]$ sets $g^{\mathbf{Z}_i} = g^{\mathbf{U}_i}$ if $\hat{x}_\ell = 0$ and sets $g^{\mathbf{Z}_i} = g^{\mathbf{V}_i}$ if $\hat{x}_\ell = 1$. Then, Sim responds to the adversary's query by returning $g_\ell^{\mathbf{N}} \in (\mathbb{G}_\ell)^{\ell\times\ell}$ where $\mathbf{N} = \mathbf{C}_0 \cdot \prod_{i=1}^{\ell} \mathbf{A}_{\hat{x}_i} + \left(\prod_{i=1}^{d} \mathbf{C}_i \cdot \mathbf{Z}_i\right) \cdot \left(\prod_{i=1}^{\ell-1} \mathbf{A}_{\hat{x}_i}\right)$.[5] Finally, when $\mathcal{A}$ outputs a bit b', Sim also outputs b'.

If the MDHE challenge is of the form $\left(g^{\mathbf{U}^*}, \left\langle g^{\mathbf{S}^i\cdot\mathbf{U}^*}\right\rangle_{i\in[d]}, g^{\mathbf{V}^*}, \left\langle g^{\mathbf{S}^i\cdot\mathbf{V}^*}\right\rangle_{i\in[d]}\right)$ for some $\mathbf{S} \in \mathbb{Z}_p^{\ell\times\ell}$, then on query $(\phi_{P(\cdot)}, x)$, for each $i \in [d]$, $\mathbf{Z}_i$ is of the form $\mathbf{S}^i \cdot \mathbf{A}_{x_\ell}$, which means that Sim responds to the query with $g_\ell^{\mathbf{N}} \in (\mathbb{G}_\ell)^{\ell\times\ell}$ where $\mathbf{N} = P(\mathbf{S}) \cdot \prod_{i=1}^{\ell} \mathbf{A}_{x_i}$, and therefore Sim has simulated $\mathsf{Expt}_0^{\mathsf{prf\text{-}rka}}$. If instead the MDHE challenge is of the form $\left(g^{\mathbf{U}^*}, \langle g^{\mathbf{U}_i}\rangle_{i\in[d]}, g^{\mathbf{V}^*}, \langle g^{\mathbf{V}_i}\rangle_{i\in[d]}\right)$ for uniformly and independently distributed $\mathbf{U}_i, \mathbf{V}_i \leftarrow \mathbb{Z}_p^{\ell\times\ell}$ across all $i \in [d]$, then on query $(\phi_{P(\cdot)}, x)$, Sim responds with $g_\ell^{\mathbf{N}} \in (\mathbb{G}_\ell)^{\ell\times\ell}$ where $\mathbf{N} = \mathbf{C}_0 \cdot \prod_{i=1}^{\ell} \mathbf{A}_{x_i} + \left(\sum_{i=1}^{d} \mathbf{C}_i \cdot \mathbf{U}_i\right) \cdot \prod_{i=1}^{\ell-1} \mathbf{A}_{x_i}$ if $x_\ell = 0$ and $\mathbf{N} = \mathbf{C}_0 \cdot \prod_{i=1}^{\ell} \mathbf{A}_{x_i} + \left(\sum_{i=1}^{d} \mathbf{C}_i \cdot \mathbf{V}_i\right) \cdot \prod_{i=1}^{\ell-1} \mathbf{A}_{x_i}$ if $x_\ell = 1$, and therefore Sim has simulated Expt_ℓ. Under the d-MDHE assumption, the claim follows. ∎

Lemma 5.3. *Under the* 1*-MDHE assumption, when* $Q = \mathrm{poly}(\lambda)$ *and* $d = \mathrm{poly}(\lambda)$*, for all* $j \in [2,\ell]$*,* $\left|\Pr\left[\mathsf{Expt}_{j-1} = 1\right] - \Pr\left[\mathsf{Expt}_j = 1\right]\right|$ *is negligible.*

[5] This can be computed using the multilinear map $\hat{e}$ given $g^{\mathbf{A}_0}$, $g^{\mathbf{A}_1}$, $\langle g^{\mathbf{Z}_i}\rangle_{i\in[d]}$, and the coefficients α_i for $i \in [0,d]$.

Proof. We construct a simulator Sim against a 1-MDHE challenger which gives Qd MDHE challenges rather than just one. A standard hybrid argument can be used to show that an adversary against a 1-MDHE challenger which gives Qd challenges can be used to construct an adversary against a 1-MDHE challenger which gives one challenge, with a $1/(Qd)$ loss in advantage. When the 1-MDHE challenger gives "real" challenges to the simulator, Sim behaves as a challenger for Experiment Expt_j, and when the 1-MDHE challenger gives "random" challenges, Sim behaves as a challenger for Experiment Expt_{j-1}.

The simulator Sim queries the 1-MDHE challenger to receive Qd challenges which we will write in the form $\left(g^{\mathbf{U}^*}, \langle g^{\mathbf{U}_{i,k}} \rangle_{i\in[d],k\in[Q]}, g^{\mathbf{V}^*}, \langle g^{\mathbf{V}_{i,k}} \rangle_{i\in[d],k\in[Q]}\right)$, where $\mathbf{U}^*, \mathbf{V}^* \in \mathbf{Rk}_\ell(\mathbb{Z}_p^{\ell\times\ell})$, and $\mathbf{U}_{i,k}, \mathbf{V}_{i,k} \in \mathbb{Z}_p^{\ell\times\ell}$ for each $i \in [d]$ and $k \in [Q]$. The simulator must decide whether there exists for each $i \in [d]$ and $k \in [Q]$ a matrix $\mathbf{S}_{i,k} \in \mathbb{Z}_p^{\ell\times\ell}$ such that $\mathbf{U}_{i,k} = \mathbf{S}_{i,k} \cdot \mathbf{U}^*$ and $\mathbf{V}_{i,k} = \mathbf{S}_{i,k} \cdot \mathbf{V}^*$, or whether all $2Qd + 2$ matrices in the set $\left\{\mathbf{U}^*, \mathbf{V}^*, \langle \mathbf{U}_{i,k} \rangle_{i\in[d],k\in[Q]}, \langle \mathbf{V}_{i,k} \rangle_{i\in[d],k\in[Q]}\right\}$ are distributed uniformly and independently.

The simulator embeds the 1-MDHE challenges by setting $\mathbf{A}_0 = \mathbf{U}^*$ and $\mathbf{A}_1 = \mathbf{V}^*$. The simulator then creates a lookup table of pairs $\mathsf{L} : \{0,1\}^{\ell-j+1} \times (\mathbb{G}_1)^{\ell\times\ell}$, initializing the table to contain the pair $\left(\varepsilon^*, \langle g^{\mathbf{R}_i} \rangle_{i\in[d]}\right)$ for a randomly chosen list of matrices $\mathbf{R}_i \leftarrow \left(\mathbb{Z}_p^{\ell\times\ell}\right)^d$. Sim also keeps a counter $\mathsf{k} \in \mathbb{Z}$, initialized to 1. Sim sends $pp = \left(g, g^{\mathbf{A}_0}, g^{\mathbf{A}_1}\right)$ to the adversary.

Now, when the adversary $\mathcal{A}$ makes a query $(\phi_{P(\cdot)}, \hat{x}) \in \Phi_{\mathsf{poly}(d)} \times \{0,1\}^\ell$, for each $i \in [0, d]$, let $\mathbf{C}_i \in \mathbb{Z}_p^{\ell\times\ell}$ be the coefficients of P, so that for all $\mathbf{M} \in \mathbb{Z}_p^{\ell\times\ell}$, $P(\mathbf{M}) = \sum_{i=0}^d \mathbf{C}_i \cdot \mathbf{M}^i$. Sim first checks if the pair $(\hat{x}_{j-1}, \langle g^{\mathbf{Z}_i} \rangle)$ exists in L, for some list of matrices $\langle \mathbf{Z} \rangle_{i\in[d]} \in \left(\mathbb{Z}_q^{\ell\times\ell}\right)^d$. If not, he adds the pairs $\left(0 \parallel (\hat{x}|_j), \langle g^{\mathbf{U}_{i,\mathsf{k}}} \rangle_{i\in[d]}\right)$ and $\left(1 \parallel (\hat{x}|_j), \langle g^{\mathbf{V}_{i,\mathsf{k}}} \rangle_{i\in[d]}\right)$ to the table L, and for all $i \in [d]$ sets $g^{\mathbf{Z}_i} = g^{\mathbf{U}_{i,\mathsf{k}}}$ if $\hat{x}_{j-1} = 0$ and sets $g^{\mathbf{Z}_i} = g^{\mathbf{V}_{i,\mathsf{k}}}$ if $\hat{x}_{j-1} = 1$, and then increments k by 1. Sim responds to the adversary's query by returning $g_\ell^{\mathbf{N}} \in (\mathbb{G}_\ell)^{\ell\times\ell}$ where $\mathbf{N} = \mathbf{C}_0 \cdot \prod_{i=1}^\ell \mathbf{A}_{\hat{x}_i} + \left(\sum_{i=1}^d \mathbf{C}_i \cdot \mathbf{Z}_i\right) \cdot \left(\prod_{i=1}^{j-2} \mathbf{A}_{\hat{x}_i}\right)$.[6] Finally, when $\mathcal{A}$ outputs a bit b', Sim also outputs b'.

If the 1-MDHE challenges are of the form $\left(g^{\mathbf{U}^*}, \langle g^{\mathbf{S}_{i,k}\cdot\mathbf{U}^*} \rangle_{i\in[d],k\in[Q]}, g^{\mathbf{V}^*}, \langle g^{\mathbf{S}_{i,k}\cdot\mathbf{V}^*} \rangle_{i\in[d],k\in[Q]}\right)$ for some uniformly and indepently chosen $\mathbf{S}_{1,1}, \ldots, \mathbf{S}_{Q,d} \in \mathbb{Z}_p^{\ell\times\ell}$, then on the k^{th} (unique) query $(\phi_{P(\cdot)}, x)$, for each $i \in [d]$, $\mathbf{Z}_i$ is of the form $\mathbf{S}_{i,k} \cdot \mathbf{A}_{x_{j-1}}$, which means that Sim responds to the query with $g_\ell^{\mathbf{N}} \in (\mathbb{G}_\ell)^{\ell\times\ell}$ where $\mathbf{N} = \mathbf{C}_0 \cdot \prod_{i=1}^\ell \mathbf{A}_{x_i} + \left(\sum_{i=1}^d \mathbf{C}_i \cdot \mathbf{Z}_i\right) \cdot \mathbf{A}_{x_{j-1}} \cdot \prod_{i=1}^{j-2} \mathbf{A}_{x_i}$, and therefore Sim has simulated Expt_j, with $\mathbf{S}_{i,k}$ playing the role of freshly chosen $\mathbf{Z}_i$ (consistent with the suffix of x). If instead the 1-MDHE challenges are of the form $\left(g^{\mathbf{U}^*}, \langle g^{\mathbf{U}_{i,k}} \rangle_{i\in[d],k\in[Q]}, g^{\mathbf{V}^*}, \langle g^{\mathbf{V}_{i,k}} \rangle_{i\in[d],k\in[Q]}\right)$ for uniformly and independently distributed $\mathbf{U}^{i,k}, \mathbf{V}^{i,k} \leftarrow \mathbb{Z}_p^{\ell\times\ell}$ across all $i \in [d]$ and $k \in [Q]$, then on the k^{th} query $(\phi_{P(\cdot)}, x)$, Sim responds with $g_\ell^{\mathbf{N}} \in (\mathbb{G}_\ell)^{\ell\times\ell}$ where $\mathbf{N} = \mathbf{C}_0 \cdot \prod_{i=1}^\ell \mathbf{A}_{x_i} + \left(\sum_{i=1}^d \mathbf{C}_i \cdot \mathbf{Z}_i\right) \cdot \prod_{i=1}^{j-2} \mathbf{A}_{x_i}$, and therefore Sim has simulated Expt_{j-1}, with $\mathbf{U}_{i,k}$ playing the role of the freshly chosen $\mathbf{Z}_i$ if $x_{j-1} = 0$, and $\mathbf{V}_{i,k}$ playing the role of the freshly chosen $\mathbf{Z}_i$ if $x_{j-1} = 1$ (consistent with the suffix of x). ∎

Lemma 5.4. $\Pr[\mathsf{Expt}_1 = 1] = \Pr\left[\mathsf{Expt}_1^{\mathsf{prf\text{-}rka}} = 1\right]$.

Proof. Recall that in Expt_1, on query $(\phi_{P(\cdot)}, x) \in \Phi_{\mathsf{poly}(d)} \times \{0,1\}^\ell$, the challenger responds with

[6] This can be computed using the multilinear map $\hat{e}$ given $g^{\mathbf{A}_0}$, $g^{\mathbf{A}_1}$, $\langle g_i^{\mathbf{Z}} \rangle_{i\in[d]}$, and the coefficients α_i for $i \in [0, d]$.

$g_\ell^{\mathbf{N}} \in (\mathbb{G}_\ell)^{\ell \times \ell}$ where $\mathbf{N} = \mathbf{C}_0 \cdot \prod_{i=1}^{\ell} \mathbf{A}_{x_i} + \sum_{i=1}^{d} \mathbf{C}_i \cdot \mathbf{U}_i$, where each of $\mathbf{U}_1, \ldots, \mathbf{U}_d \in \mathbb{Z}_p^{\ell \times \ell}$ is uniformly and independently distributed for distinct input queries x. Since there exists some $j \in [d]$ for which $\mathbf{C}_j$ is full rank, it follows that $\mathbf{C}_j \cdot \mathbf{U}_j$ and hence $\mathbf{N}$ is distributed as a uniform element in $\mathbb{Z}_p^{\ell \times \ell}$, independently for each input query $x \in \{0,1\}^\ell$. ∎

Applying Lemmas 5.2–5.4 yields Theorem 5.1.

6 Conclusions

We construct the first lattice-based PRFs secure against a class of related-key attacks from an (almost) linear class of functions. We achieve RKA security under the standard (super-polynomial) LWE assumption for a restricted linear class of related-key functions and this result is comparable to the DDH-based RKA-secure PRF construction by Bellare and Cash [6]. Under the powerful multilinear map abstraction [19], we construct RKA-secure PRFs against a large and natural class of affine related-key deriving functions with minimal restrictions. We believe this to be the most expressive affine class of transformations attainable under the Bellare-Cash framework. We also achieve the weaker notion of unique-input RKA security for an affine class of related-key deriving functions by considering the LWE-based key homomorphic PRF by Boneh *et al.* [13]. We show that by working with the proof of pseudorandomness and utilizing the algebraic structure of the PRF, we can overcome restrictions on the related-key class that are necessary to apply the Bellare-Cash framework. Finally, we show how, under the d-MDHE assumption in the presence of multilinear maps, we can achieve RKA security against unique-input adversaries for the class of degree-d polynomials. Our work on constructing new RKA-secure PRFs leads to several interesting open problems:

- ◇ Can we construct LWE-based PRFs under the Bellare-Cash framework for a class less restrictive than Φ_{lin^*}? The only known LWE-based PRFs [4, 13] both require rounding and have "error terms" in proofs that have to be carefully dealt with. This will require a more careful application of the Bellare-Cash framework.
- ◇ Can we construct unique-input RKA-secure PRFs from other LWE-based PRFs by Banerjee *et al.* [4] and (more recently) Banerjee and Peikert [3]?
- ◇ Can we construct RKA-secure PRFs against unique-input adversaries for classes of polynomials from more standard assumptions such as LWE or DLIN?

References

[1] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, 2009.

[2] Benny Applebaum, Danny Harnik, and Yuval Ishai. Semantic security under related-key attacks and applications. In *ICS*, 2011.

[3] Abhishek Banerjee and Chris Peikert. New and improved key-homomorphic pseudorandom functions. Cryptology ePrint Archive, Report 2014/074, 2014. `http://eprint.iacr.org/`.

[4] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *EUROCRYPT*, 2012.

[5] Alessandro Barenghi, Luca Breveglieri, Israel Koren, and David Naccache. Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures. *Proceedings of the IEEE*, 100(11), 2012.

[6] Mihir Bellare and David Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In *CRYPTO*, 2010.

[7] Mihir Bellare, David Cash, and Rachel Miller. Cryptography secure against related-key attacks and tampering. In *ASIACRYPT*, 2011.

[8] Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In *EUROCRYPT*, 2003.

[9] Mihir Bellare, Sarah Meiklejohn, and Susan Thomson. Key-versatile signatures and applications: RKA, KDM and joint enc/sig. *To appear in EUROCRYPT*, 2014.

[10] Mihir Bellare, Kenneth G. Paterson, and Susan Thomson. RKA security beyond the linear barrier: IBE, encryption and signatures. In *ASIACRYPT*, 2012.

[11] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In *CRYPTO*, 1997.

[12] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of eliminating errors in cryptographic computations. *J. Cryptology*, 14(2), 2001.

[13] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic PRFs and their applications. In *CRYPTO (1)*, 2013.

[14] Dan Boneh, Hart William Montgomery, and Ananth Raghunathan. Algebraic pseudorandom functions with improved efficiency from the augmented cascade. In *ACM CCS*, 2010.

[15] Joseph Bonneau and Ilya Mironov. Cache-collision timing attacks against aes. In *CHES*, 2006.

[16] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *STOC*, 2013.

[17] Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In *PKC*, 2005.

[18] Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Michael Stay, David Wagner, and Doug Whiting. Improved cryptanalysis of rijndael. In *FSE*, 2000.

[19] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In *CRYPTO (2)*, 2013.

[20] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 34(4), 1986.

[21] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. Lest we remember: cold-boot attacks on encryption keys. *Commun. ACM*, 52(5), 2009.

[22] Goce Jakimoski and Yvo Desmedt. Related-key differential cryptanalysis of 192-bit key aes variants. In *Selected Areas in Cryptography*, 2003.

[23] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *CRYPTO*, 1996.

[24] Allison B. Lewko and Brent Waters. Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In *CCS*, 2009.

[25] Stefan Lucks. Ciphers secure against related-key attacks. In *FSE*, 2004.

[26] Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Computers*, 51(5), 2002.

[27] Daniele Micciancio and Chris Peikert. Hardness of sis and lwe with small parameters. In *CRYPTO (1)*, 2013.

[28] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *FOCS*, 1997.

[29] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. *SIAM J. Comput.*, 41(4), 2012.

[30] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*. ACM, 2009.

[31] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.

[32] Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *CCS*, 2009.

[33] Hovav Shacham. A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. *IACR Cryptology ePrint Archive*, 2007.

[34] Hoeteck Wee. Public key encryption against related key attacks. In *PKC*, 2012.

[35] Wentao Zhang, Lei Zhang, Wenling Wu, and Dengguo Feng. Related-key differential-linear attacks on reduced aes-192. In *INDOCRYPT*, 2007.

A Security of the d-MDHE Assumption in the Generic Group Model

Recall the definition of the d-MDHE problem, defined in Definition 2.6, and the d-MDHE assumption for which we use to prove the security of Theorem 5.1. In this section, we work with a generalized form of the the d-MDHE problem, defined as follows. Consider a sequence of groups $\vec{\mathbb{G}} = (\mathbb{G}_1, \ldots, \mathbb{G}_\ell)$, each of prime order p, with a set of bilinear maps $\hat{e}_i$ for $i \in [1, \ell - 1]$, and a generator g of $\mathbb{G}_1$.

Definition A.1 (Multilinear Diffie-Hellman Exponent, generalized)**.** The d-Multilinear Diffie-Hellman Exponent (d-MDHE) problem in the presence of a graded ℓ-linear map, the sequence of groups $\vec{\mathbb{G}}$, and $\{\hat{e}_i\}_{i\in[1,\ell-1]}$, for any positive integers m, n where $n \geq \ell$, is to distinguish between the two distributions

$$\left\{g^{\mathbf{A}}, g^{\mathbf{AS}}, \ldots, g^{\mathbf{AS}^d}\right\}_{\mathbf{A}\leftarrow\mathbb{Z}_p^{m\times n}, \mathbf{S}\leftarrow\mathbb{Z}_p^{n\times n}} \qquad \text{and} \qquad \left\{g^{\mathbf{U}^{(0)}}, \ldots, g^{\mathbf{U}^{(d)}}\right\}_{\mathbf{U}^{(0)},\ldots,\mathbf{U}^{(d)}\leftarrow\mathbb{Z}_p^{m\times n}}.$$

Note that the Definition 2.6 is the assumption that the above (generalized) problem is computationally hard when $m = 2n$. In what follows, we show that when $d\ell \leq n$ and p is sufficiently large, any adversary restricted to the generic group model has a negligible advantage in solving the d-MDHE problem. Although Theorem 5.1 relies on the d-MDHE assumption *without* the extra restriction that $d\ell \leq n$, we note that the proof of Theorem 5.1 can be modified slightly to hold for when all matrices are drawn from $\mathbb{Z}_p^{n\times n}$ instead of $\mathbb{Z}_p^{\ell\times\ell}$. In other words, we can show that for $d\ell \leq n$, under the d-MDHE assumption, the PRF F_{DLIN} over $n \times n$ matrices is RKA-secure against unique-input adversaries for the class $\Phi_{\mathsf{poly}(d)}$ over $n \times n$ matrices.

A.1 Definitions

Matrices and sets. For an arbitrary matrix $\mathbf{M}$, let $\mathbf{M}_{i,j}$ represent entry (i, j) of $\mathbf{M}$. Let $\mathbf{A}, \mathbf{U}^{(0)}, \ldots, \mathbf{U}^{(d)} \in \mathbb{Z}_p^{m\times n}$ and $\mathbf{S} \in \mathbb{Z}_p^{n\times n}$. Consider the set of variables comprising entries from $\mathbf{U}^{(k)}$ denoted as $\mathbb{U} = \left\{\mathbf{U}_{i,j}^{(k)} : i \in [m], j \in [n], k \in [0, d]\right\}$. Similarly, define the set $\mathbb{S} = \{\mathbf{A}_{i,j}\}_{i\in[m],j\in[n]} \cup \{\mathbf{S}_{i,j}\}_{i,j\in[n]}$ for entries of the matrices $\mathbf{A}$ and $\mathbf{S}$.

Polynomials. For an arbitrary set S, we write $P \in \mathbb{F}_p[S]$ if there exist integers $\alpha, \beta_1, \ldots, \beta_\alpha \geq 0$, scalars $c_1, \ldots, c_\alpha \in \mathbb{F}_p$, and variables $X_{i,j} \in S$ for each $i \in [\alpha]$ and $j \in [\beta_i]$ such that

$$P(S) = \sum_{i=1}^{\alpha} c_i \prod_{j=1}^{\beta_i} X_{i,j}. \tag{A.1}$$

We refer to Equation A.1 as a *standard form* of P if $c_1, \ldots, c_\alpha$ are non-zero, and the sets $\{X_{1,j}\}_{j\in[\beta_1]}, \ldots, \{X_{\alpha,j}\}_{j\in[\beta_\alpha]}$ are all distinct.

Definition A.2 (Induced polynomial)**.** Let $P \in \mathbb{F}_p[\mathbb{U}]$ and $Q \in \mathbb{F}_p[\mathbb{S}]$. We say that P induces Q, or that Q is the induced polynomial of P, if it is the case that if $\mathbf{U}^{(i)} = \mathbf{AS}^i$ for all $i \in [0, d]$, then for all $\mathbf{A} \in \mathbb{Z}_p^{m\times n}$ and $\mathbf{S} \in \mathbb{Z}_p^{n\times n}$, $P(\mathbb{U}) \equiv Q(\mathbb{S})$.

Definition A.3 ($\mathbb{U}$-terms and $\mathbb{S}$-terms)**.** If $P \in \mathbb{F}_p[\mathbb{U}]$ and integers $\alpha, \beta_1, \ldots, \beta_\alpha \geq 0$, scalars $c_1, \ldots, c_\alpha \in \mathbb{F}_p$, and variables $X_{1,1}, \ldots, X_{\alpha,\beta}$ are the variables of the standard form of P, then for each $i \in [\alpha]$, we will refer to the expression $\prod_{j=1}^{\beta_i} X_{i,j}$ as a $\mathbb{U}$-*term* of P. If instead $P \in \mathbb{F}_p[\mathbb{S}]$, then for each $i \in [\alpha]$, we will refer to the expression $\prod_{j=1}^{\beta_i} X_{i,j}$ as a $\mathbb{S}$-*term* of P.

$\mathbb{S}$-terms. Let $P \in \mathbb{F}_p[\mathbb{U}]$ and $Q \in \mathbb{F}_p[\mathbb{S}]$ be the induced polynomial of P. Note that each $\mathbb{S}$-term of Q can be written in the form

$$\prod_{k=1}^{\ell'} \left(\mathbf{A}_{i_1^{(k)}, i_2^{(k)}} \left(\prod_{j=2}^{d_k} \mathbf{S}_{i_j^{(k)}, i_{j+1}^{(k)}} \right) \right) \tag{A.2}$$

for some integer $\ell' \geq 0$, $d_1, \ldots, d_{\ell'} \leq d$, and indices $\left\{i_j^{(k)}\right\}_{k\in[\ell'],j\in[d_k+1]}$. Note also that every product of terms written in the form of Equation A.2 can be interpreted as a $\mathbb{S}$-term of Q.

Definition A.4 (Index set)**.** Let $P \in \mathbb{F}_p[\mathbb{U}]$, let $Q \in \mathbb{F}_p[\mathbb{S}]$ be the induced polynomial of P, and let ρ be a $\mathbb{S}$-term of Q. Let $\ell' \leq \ell$, $d_1, \ldots, d_{\ell'} \leq d$, and indices $i_j^{(k)}$ for each $k \in [\ell']$ and $j \in [d_{\ell'} + 1]$ be a setting of the variables of ρ as in Equation (A.2). An *index set* of the $\mathbb{S}$-term ρ is a set of ℓ' tuples, where for each $k \in [\ell']$, the k^{th} tuple contains the indices $\left(i_1^{(k)}, \ldots, i_{d_k+1}^{(k)}\right)$. By definition, every $\mathbb{S}$-term has at least one index set.

Definition A.5 (Well-formed $\mathbb{S}$-term)**.** Let $P \in \mathbb{F}_p[\mathbb{U}]$, let $Q \in \mathbb{F}_p[\mathbb{S}]$ be the induced polynomial of P, let ρ be a $\mathbb{S}$-term of Q, and let ℓ', $d_1, \ldots, d_{\ell'}$, and indices $\left\{i_j^{(k)}\right\}_{k\in[\ell'],j\in[d_k+1]}$ be a setting of the variables of ρ as in Equation (A.2). Let $S = \bigcup_{k=1}^{\ell'} \left\{i_2^{(k)}, \ldots, i_{d_k}^{(k)}\right\}$. We say that a the $\mathbb{S}$-term ρ is *well-formed* if the following is true:

Property 1: The elements of the set S are all distinct, and

Property 2: For each $k \in [\ell']$, $i_{d_k+1}^{(k)} \notin S$.

A.2 Useful Lemmas

The following lemma follows from the definition of a well-formed $\mathbb{S}$-term.

Lemma A.6. *Let $P \in \mathbb{F}_p[\mathbb{U}]$ be a degree-ℓ polynomial and let $Q \in \mathbb{F}_p[\mathbb{S}]$ be the induced polynomial of P. Every well-formed $\mathbb{S}$-term of Q has a unique index set.*

Proof. Let ρ be a well-formed $\mathbb{S}$-term of the form $\prod_{k=1}^{\ell'}\left(\mathbf{A}_{i_1^{(k)},i_2^{(k)}}\left(\prod_{j=2}^{d_k}\mathbf{S}_{i_j^{(k)},i_{j+1}^{(k)}}\right)\right)$ for some $\ell' \in [0, \ell]$, $d_1, \ldots, d_{\ell'} \in [d]$, and for each $k \in [\ell']$ and $j \in [d_k + 1]$, indices $i_j^{(k)} \in [n]$. By definition, the set $\pi = \left\{\left(i_1^{(1)}, \ldots, i_{d_1+1}^{(1)}\right), \ldots, \left(i_1^{(\ell')}, \ldots, i_{d_{\ell'}+1}^{(\ell')}\right)\right\}$ is an index set of ρ.

Let $\pi' \neq \pi$ be another index set of ρ. The proof focuses on a maximal subsequence of indices (denoted γ^*) as part of some tuple in both π and π'. Let $\gamma, \gamma', \gamma^*, \gamma'_{\mathsf{pre}}, \gamma_{\mathsf{post}}, \gamma'_{\mathsf{post}}$ be (possibly empty) tuples such that $\gamma \in \pi$ and $\gamma' \in \pi'$ can be written in the form

$$\begin{aligned} \gamma &= \qquad \gamma^* \parallel \gamma_{\mathsf{post}} \\ \gamma' &= \gamma'_{\mathsf{pre}} \parallel \gamma^* \parallel \gamma'_{\mathsf{post}}, \end{aligned}$$

and the length of γ^* is maximized. Without loss of generality, we will assume that the tuple γ is such that it is not the case that *both* γ_{post} and γ'_{post} are empty. To see why such a tuple γ exists, note that if no such tuple existed, then for every tuple in π, its elements form a subsequence of some tuple in π', which contradicts the assumption that $\pi \neq \pi'$, or that both π and π' are valid index sets of ρ.

There are two cases to consider, and we will show that each leads to a contradiction based on the well-formedness of ρ. Let r be the last entry of γ^*.

Case 1: If either $\gamma_{\mathsf{post}} \neq \emptyset$ and $\gamma'_{\mathsf{post}} = \emptyset$, or $\gamma_{\mathsf{post}} = \emptyset$ and $\gamma'_{\mathsf{post}} \neq \emptyset$, then there must exist a tuple in π, integers $k \in [\ell']$, $d_k \in [d]$, and $j \in [d_k]$, where the tuple is of the form $\left(i_j^{(k)}, \ldots, i_{d_k+1}^{(k)}\right)$ with $r = i_j^{(k)}$. This contradicts Property 2 of the well-formedness of ρ.

Case 2: If both $\gamma_{\mathsf{post}} \neq \emptyset$ and $\gamma'_{\mathsf{post}} \neq \emptyset$, then there must exist a pair of tuples in π, both of which contain r as an entry. This contradicts Property 1 of the well-formedness of ρ.

We have contradicted the assumption that $\pi' \neq \pi$. The claim follows. ∎

Lemma A.7. *Let $P \in \mathbb{F}_p[\mathbb{U}]$ be a degree-ℓ polynomial, and let $Q \in \mathbb{F}_p[\mathbb{S}]$ be its induced polynomial. If $d\ell \leq n$, $p > \ell$, p is prime, and $P \not\equiv 0$, then $Q \not\equiv 0$.*

Proof. Assuming that $P \not\equiv 0$, let τ be an arbitrary $\mathbb{U}$-term of P, and let $Q' \in \mathbb{F}_p[\mathbb{S}]$ be the polynomial induced by τ.

Claim A.8. *If $d\ell \leq n$, then there exists a well-formed $\mathbb{S}$-term ρ in Q'.*

Proof. Let τ be of the form $\prod_{k=1}^{\ell'} \mathbf{U}^{(d_k)}_{i_k,j_k}$ for some $\ell' \in [0,\ell]$, $d_1,\ldots,d_{\ell'} \in [d]$, $i_1,\ldots,i_{\ell'} \in [m]$, and $j_1,\ldots,j_{\ell'} \in [n]$. Then, the set of all $\mathbb{S}$-terms in Q' is the set of all $\mathbb{S}$-terms which can be written in the form of Equation (A.2), where for each $k \in [\ell']$, $i_1^{(k)} = i_k$ and $i_{d_k+1}^{(k)} = j_k$, and $i_2^{(k)},\ldots,i_{d_k}^{(k)} \in [n]$.

To pick a well-formed $\mathbb{S}$-term, we must ensure that, for $k \in [\ell']$ and $j \in [2,d_k]$, each $i_j^{(k)}$ is picked distinctly from the set $[n] \setminus \{j_k\}_{k\in[\ell']}$. Note that the set $[n] \setminus \{j_k\}_{k\in[\ell']}$ has at least $n-\ell' \geq n-\ell$ elements, and we must pick elements from this set for at most $\ell'(d-1) \leq \ell(d-1)$ indices. If $d\ell \leq n$, $\ell(d-1) \leq n-\ell'$, which implies that the indices can be picked to form a well-formed $\mathbb{S}$-term, which implies the claim. ∎

Claim A.9. *Let ρ be a well-formed $\mathbb{S}$-term in the induced polynomial of τ. Let Γ be the set of all $\mathbb{U}$-terms in P whose induced polynomial contain ρ. Then, $\Gamma = \{\tau\}$.*

Proof. Let $\left\{\left(i_1^{(1)},\ldots,i_{d_1+1}^{(1)}\right),\ldots,\left(i_1^{(\ell')},\ldots,i_{d_{\ell'}+1}^{(\ell')}\right)\right\}$ be an index set of ρ, for some $\ell' \in [0,\ell]$, $d_1,\ldots,d_{\ell'} \in [d]$, and indices $i_1^{(k)},\ldots,i_{d_k+1}^{(k)} \in [n]$ for each $k \in [\ell']$. Then, the $\mathbb{U}$-term τ which contains the $\mathbb{S}$-term ρ must be of the form

$$\tau = \prod_{k=1}^{\ell'} \mathbf{U}^{(d_k)}_{i_1^{(k)},i_{d_k+1}^{(k)}}.$$

Since this index set for ρ is unique by Lemma A.6, it follows that this $\mathbb{U}$-term is unique, as well, which proves the claim. ∎

Claim A.10. *If $p > \ell$ and p is prime, and ρ is a well-formed $\mathbb{S}$-term of τ, then the scalar associated with ρ in τ is non-zero.*

Proof. Let τ be of the form $\prod_{k=1}^{\ell'} \left(\mathbf{U}^{(d_k)}_{i_k,j_k}\right)^{e_k}$ for $\ell' \in [\ell]$, and for each $k \in [\ell']$, *distinct* triples of indices $(i_k,j_k,d_k) \in [n] \times [n] \times [d]$, and $e_k \in [\ell]$. Then, the scalar associated with each well-formed $\mathbb{S}$-term of the polynomial induced by τ is equal to $\prod_{k=1}^{\ell'}(e_k!)$. To see this, note that ρ has a unique index set of the form $\left\{\left(i_1^{(k)},\ldots,i_{d_k+1}^{(k)}\right)\right\}_{k\in[\ell']}$, for indices $i_j^{(k)}$ for each $k \in [\ell']$ and $j \in [d_k+1]$, where $i_1^{(k)} = i_k$ and $i_{d_k+1}^{(k)} = j_k$. The number of times ρ appears in the polynomial induced by τ is exactly the number of ways to bijectively map each tuple of the index set with the set $[\ell']$ such that the k^{th} tuple, of the form $\left(i_1^{(k)},\ldots,i_{d_k+1}^{(k)}\right)$, is such that $i_1^{(k)} = i_k$ and $i_{d_k+1}^{(k)} = j_k$. The only such maps which can exist are permutations of duplicate elements in the index set, which is given by the expression $\prod_{k=1}^{\ell'}(e_k!)$. Since $p > \ell$, p is prime, and $e_k \leq \ell$ for each $k \in [\ell']$, this expression is not a multiple of p, and hence it is non-zero in $\mathbb{F}_p$. ∎

Note that the scalar associated with the $\mathbb{S}$-term ρ (whose existence is ensured by Claim A.8) in the polynomial induced by τ is non-zero, by Claim A.10. Therefore, since the scalar associated with τ is also non-zero in $\mathbb{F}_p$, and since p is prime, the scalar associated with ρ in Q must also be non-zero in $\mathbb{F}_p$. Therefore, we conclude that $Q \not\equiv 0$, which proves the lemma. ∎

A.3 The Main Theorem

Theorem A.11. *Let p, s, ℓ, m, n, d be positive integers such that $p > \ell$, p is prime, $s = \lceil \log_2 p \rceil$, and $d\ell \leq n$. Let $\mathcal{A}$ be a probabilistic polynomial time adversary. Let $\mathbb{G}_1, \ldots, \mathbb{G}_\ell$ be groups of order p, along with pairing functions $\hat{e}_{i,j} : \mathbb{G}_i \times \mathbb{G}_j \to \mathbb{G}_{i+j}$ for all i, j such that $i + j \leq \ell$. Let $\zeta_0, \ldots, \zeta_d$ be a set of random encodings, where each $\zeta_i : \mathbb{G}_i \to \{0,1\}^s$. The adversary is given access to an oracle to compute the group action for each group $\mathbb{G}_i$ as well as an oracle to compute the pairing function $\hat{e}_{i,j}$. Let q be the number of queries the adversary makes to these oracles. Then,*

$$\left| \Pr\left[\mathcal{A}\left(p, \zeta_0\left(g^{\mathbf{U}^{(0)}}\right), \ldots, \zeta_0\left(g^{\mathbf{U}^{(d)}}\right)\right) = b \right] - \frac{1}{2} \right| \leq \frac{\binom{q+mn(d+1)}{2}\ell}{p}$$

where a bit $b \in \{0,1\}$ is chosen randomly and if $b = 0$, then $\mathbf{U}^{(i)} = \mathbf{A}\mathbf{S}^i$ for $\mathbf{A}$ and $\mathbf{S}$ sampled uniformly and independently from $\mathbb{Z}_p^{m \times n}$ and $\mathbb{Z}_p^{n \times n}$, respectively, and if $b = 1$, then $\mathbf{U}^{(0)}, \ldots, \mathbf{U}^{(d)}$ are sampled uniformly and independently from $\mathbb{Z}_p^{m \times n}$.

Proof. The simulator will keep ℓ lists $L_1, \ldots, L_\ell$ of distinct polynomials mapping to random values (for each of the ℓ levels of encodings). The list $L_i : \mathbb{G}_i \to \{0,1\}^s$ is initialized to contain a map of each variable in the set $\left\{ g^{\mathbf{A}}, \ldots, g^{\mathbf{A}\mathbf{S}^d} \right\}$ to a random string in $\{0,1\}^s$. We will use $\mathcal{R}(L_i) \subset \{0,1\}^s$ to denote the set of random strings that have already been assigned to in group $\mathbb{G}_i$.

When $\mathcal{A}$ makes a query to the oracle for a group operation in $\mathbb{G}_i$ on two random encodings $\zeta_i(x)$ and $\zeta_i(y)$, the simulator performs polynomial addition on x and y to obtain the polynomial z. If a random encoding for z is already defined in the list L_i, then the simulator returns this string. Otherwise, the simulator picks a fresh random string in $\{0,1\}^s \setminus \mathcal{R}(L_i)$ to represent $\zeta_i(z)$, returning this string. Similarly, when $\mathcal{A}$ makes a query to the oracle for a pairing of two random encodings $\zeta_i(x)$ and $\zeta_j(y)$, the simulator performs polynomial multiplication on x and y to obtain the polynomial z. If a random encoding for z is already defined in the list L_{i+j}, then the simulator returns this string. Otherwise, the simulator picks a fresh random string in $\{0,1\}^s \setminus \mathcal{R}(L_{i+j})$ to represent $\zeta_{i+j}(z)$, returning this string. It follows that the simulation is perfect unless the chosen random variables for $\mathbf{A} \in \mathbb{Z}_p^{m \times n}$ and $\mathbf{S} \in \mathbb{Z}_p^{n \times n}$ result in an equality relation between intermediate values that is not an equality of polynomials. Furthermore, note that in a perfect simulation, the distribution of the random strings $\zeta_0\left(g^{\mathbf{U}^{(0)}}\right), \ldots, \zeta_0\left(g^{\mathbf{U}^{(d)}}\right)$ when $b = 0$ is identical to their distribution when $b = 1$, and hence the adversary has advantage $1/2$ in this case.

The adversary is given $mn(d+1)$ random encodings from the challenger, and can receive at most $q + mn(d+1)$ random encodings after q oracle queries. Hence, there are at most $\binom{q+mn(d+1)}{2}$ distinct pairs which represent polynomials that the adversary may check for equality. Note that each equality check can be formulated as a non-zero degree-ℓ polynomial $P \in \mathbb{F}_p[\mathbb{U}]$. Hence, by Lemma A.7, the induced polynomial Q is also non-zero. By the Schwartz-Zippel lemma, since Q is a degree-ℓ non-zero polynomial, the probability that a random assignment of variables results in the polynomial evaluating to 0 is at most ℓ/p. The claim follows via a union bound over all distinct pairs of polynomials that the adversary may check for equality. ∎