KDM-Security via Homomorphic Smooth Projective Hashing

Hoeteck Wee[⋆]

ENS, Paris, France
wee@di.ens.fr

Abstract. We present new frameworks for constructing public-key encryption schemes satisfying key-dependent message (KDM) security and that yield efficient, universally composable oblivious transfer (OT) protocols via the dual-mode cryptosystem framework of Peikert, Waters and Vaikuntanathan (Crypto 2008).

– Our first framework yields a conceptually simple and unified treatment of the KDM-secure schemes of Boneh et al. (Crypto 2008), Brakerski and Goldwasser (Crypto 2010) and Brakerski, Goldwasser and Kalai (TCC 2011) in the single-key setting.
– Using our second framework, we obtain new dual-mode cryptosystems based on the d-linear, quadratic residuocity and decisional composite residuocity assumptions.

Both of these frameworks build on the notion of smooth projective hashing introduced by Cramer and Shoup (Eurocrypt 2002), with the additional requirement that the hash function is homomorphic, as is the case for all known instantiations.

[⋆] CNRS, INRIA and Columbia University. Partially supported by ANR Project EnBiD (ANR-14-CE28-0003), NSF Award CNS-1445424 and the Alexander von Humboldt Foundation.

1 Introduction

The most basic security guarantee we require of a public key encryption scheme is that of semantic security against chosen-plaintext attacks (CPA) [21]: it is infeasible to learn anything about the plaintext from the ciphertext. However, a series of increasingly sophisticated use of encryption —both directly in the case of practical applications, and indirectly as a cryptographic building block in more theoretical work — call for encryption schemes with much stronger security guarantees. In this work, we consider two such security notions.

Key-dependent message (KDM) security. The standard CPA security definition does not provide any guarantee where the plaintext depends on the secret key (as pointed out in [21]), as may be the case in disk encryption. It was later observed that this situation is not so unlikely and may sometimes even be desirable [12, 1]. Black, Rogaway and Shrimpton [7] formally defined key-dependent message (KDM) security: roughly speaking, we want to guarantee semantic security even against an adversary that can obtain encryptions of (efficient) functions of its choosing, taken from some specified class of functions $\mathcal{F}$, applied to the secret key.

Several years ago, Boneh et al. (BHHO) [9] presented a public-key encryption scheme that is KDM-secure w.r.t. the class of affine functions under the decisional Diffie-Hellman (DDH) assumption. Since then, Applebaum et al. [4] presented a scheme under the LWE assumption (which is itself a variant of Regev's cryptosystem [33]) and Brakerski and Goldwasser [10] presented a BHHO-like scheme based on the quadratic residuocity (QR) and decisional composite residuocity (DCR) assumptions. All of these schemes achieve KDM-security w.r.t. the class of affine functions, which can in turn be "boosted" to the class of circuits of a-priori bounded size [5, 3]. In spite of the fact that many of these schemes inherit the BHHO algebraic structure, there does not seem to be a general principle that explains the design or analysis of these schemes: the BHHO analysis uses an intermediate notion of an "expanded system", whereas that of Brakerski and Goldwasser rely on an incomparable "interactive vector" game.

Dual-mode cryptosystems. Dual-mode cryptosystems were put forth by Peikert, Vaikuntanathan and Waters [32] as a tool for constructing efficient and universally composable oblivious transfer (OT) protocols. Oblivious transfer is a fundamental two-party cryptographic primitive for secure two-party and multi-party computation [35, 20, 28]: it allows one party, called the receiver, to obtain exactly one of two values from another party, called the sender. The receiver remains oblivious to the other value, and the sender is oblivious to which value was received.

A natural approach towards realizing OT is to have the receiver generate a pair of public keys, and have the sender encrypt both of its input values under the respective public keys [17, 19]. In order to provide security against a malicious sender, we can simply generate a pair of "normal" public keys along with the corresponding secret keys and we can then decrypt the ciphertexts sent by the sender to extract both its inputs. On the other hand, if the receiver is malicious, we need to ensure that (at least) one of the two public keys be "messy", namely it carries no information about the ciphertext encrypted under the key.

A dual-mode cryptosystem provides exactly both of these guarantees in the common reference string (CRS) model. The cryptosystem admits two types of public keys, "normal" keys that enable correct decryption, and "messy" keys that carries no information statistically about the ciphertext. Moreover, a simulator can generate the CRS in one of two indistinguishable modes: a "messy" mode which ensures that amongst any pair of possibly adversarially chosen public keys, at least one must be “messy”; and a “decryption” mode which allows a simulator to generate a pair of “normal” keys.

Peikert et al. also presented three instantiations of dual-mode cryptosystems based on DDH, QR and LWE. However, there seems to be no overarching theme to the three constructions – the DDH-based scheme relies on a “re-randomization trick” from the earlier OT protocols of Naor and Pinkas [30] whereas the QR-based scheme relies on algebraic properties of Cocks’ IBE scheme [14].

Our results. We present new frameworks for constructing KDM-secure encryption schemes and dual-mode cryptosystems that admit a very simple and modular analysis. Both of these frameworks build on the notion of smooth projective hashing, introduced by Cramer and Shoup in the context of CCA-secure encryption [16, 15], with the additional requirement that the hash function is homomorphic, as is the case for all known instantiations. Using our frameworks, we obtain:

- a unified treatment of the KDM-secure encryption schemes based on DDH, QR, and DCR given in [9, 10] for affine functions of the secret key, as well as those for low-degree functions of the secret key in [11] (we focus here on the single-key setting, which already captures much of the difficulty in realizing KDM-security in prior works; see Section 2.1 for a discussion on multiple keys),
- new constructions of dual-mode cryptosystems: (i) a construction based on the d-linear assumption, generalizing the previous construction based on DDH; (ii) a simple construction based on QR, which does not rely on the Cocks IBE; (iii) a new construction based on DCR.

We regard our first construction for KDM security as our primary technical contribution. The second for dual-mode cryptosystems builds heavily upon existing constructions of OT from smooth project hashing in [23], although highlighting the role of the group structure and homomorphism for dual-mode cryptosystems appears to be novel to this work (c.f. comparison in Section 2.2).

Our high-level approach for KDM security is quite simple. Via the projective property, we will define ciphertexts via decryption with the secret key instead of encryption with the public key. Now, by feeding the decryption algorithm some “malformed” ciphertext, decryption leaks a function f of the secret key $\textsc{sk}$. In fact, we can design the malformed ciphertexts carefully so that they decrypt to $f(\textsc{sk})$; moreover, these malformed ciphertexts are indistinguishable from random encryptions of $f(\textsc{sk})$. It is important here that the distribution of the malformed ciphertext depends only on f and the public key $\textsc{pk}$. For this to work out, we require some algebraic structure for the decryption algorithm and the space of ciphertexts, as is captured by precisely by homomorphic projective hashing.

2 Overview of Our Constructions

Smooth projective hashing. We begin with an informal overview of smooth projective hashing [16, 15], since our constructions build on this framework. We consider a family of hash functions $\Lambda_{\textsc{hk}}(\cdot)$ indexed by a hashing key $\textsc{hk}$, whose input comes from a group $\mathcal{G}$. Let $\mathcal{G}_{\textsc{yes}}$ be a subgroup of $\mathcal{G}$ and let $\mu(\cdot)$ denote a projection map defined on the hashing key $\textsc{hk}$. We are interested in hash functions that satisfy the following properties:

- (projective) for $C \in \mathcal{G}_{\textsc{yes}}$, the value $\Lambda_{\textsc{hk}}(C)$ is uniquely determined by $\mu(\textsc{hk})$ and C. Moreover, there is an algorithm Pub that given $\mu(\textsc{hk})$ along with the randomness r used to sample C, outputs $\Lambda_{\textsc{hk}}(C)$.
- (smoothness) for $C \notin \mathcal{G}_{\textsc{yes}}$, the value $\Lambda_{\textsc{hk}}(C)$ is statistically close to random even given $\mu(\textsc{hk})$ and C.
- (homomorphic) for all $C_0, C_1 \in \mathcal{G}$, we have $\Lambda_{\textsc{sk}}(C_0 \cdot C_1) = \Lambda_{\textsc{sk}}(C_0) \cdot \Lambda_{\textsc{sk}}(C_1)$.

In addition, we require that the uniform distributions over $\mathcal{G}_{\mathrm{YES}}$ and $\mathcal{G}$ be computationally indistinguishable, and that the uniform distributions over $\mathcal{G}_{\mathrm{YES}}$ and $\mathcal{G}_{\mathrm{NO}} := \mathcal{G} \setminus \mathcal{G}_{\mathrm{YES}}$ are also computationally indistinguishable. (If $\mathcal{G}_{\mathrm{YES}}$ has negligible density, then the former implies the latter.)

2.1 KDM-security

Starting with a smooth projective hash function $\Lambda_{\mathrm{HK}}(\cdot)$ defined on $\mathcal{G}$, we can build a CPA-secure encryption scheme —which we will refer to as the "Cramer-Shoup scheme"— as follows:

- $\mathsf{Gen}(1^k)$: Sample a uniform hashing key HK and output the key pair

  $$\mathrm{PK} := \mu(\mathrm{HK}) \quad \text{and} \quad \mathrm{SK} := \mathrm{HK}$$

  Henceforth, we will use SK and HK interchangeably for this scheme.
- $\mathsf{Enc}(\mathrm{PK}, m)$: To encrypt a message m, sample $C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}$ with randomness r, output the ciphertext

  $$(C, \Lambda_{\mathrm{SK}}(C) \cdot m)$$

  where $\Lambda_{\mathrm{SK}}(C)$ is computed via the projective property using $\mathsf{Pub}(\mathrm{PK}, C, r)$.
- $\mathsf{Dec}(\mathrm{SK}, (C, \psi))$: On input a ciphertext (C, ψ), output the plaintext

  $$(\Lambda_{\mathrm{SK}}(C)^{-1} \cdot \psi)$$

A standard argument shows that this scheme is CPA-secure: we switch the distribution of C in the ciphertext to $C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{NO}}$ and then by smoothness, the ciphertext statistically hides m. Moreover:

> **Theorem (informal).** Suppose in addition that $\Lambda_{\mathrm{SK}}(\cdot)$ is homomorphic. Then this encryption scheme is KDM-secure[1] w.r.t. the class of functions $\{\mathrm{SK} \mapsto \Lambda_{\mathrm{SK}}(e)\}$ for any $e \in \mathcal{G}$.

Once we have KDM-security for affine functions, we can "boost" to the class of circuits of a-priori bounded size [5, 3].

Simulating KDM queries. The core difficulty lies in simulating encryptions of $\Lambda_{\mathrm{SK}}(e)$ given only the public key, which turns out to be really simple in our framework.

$$\begin{aligned}
\mathsf{Enc}(\mathrm{PK}, \Lambda_{\mathrm{SK}}(e)) &\equiv \langle C, \mathsf{Pub}(\mathrm{PK}, C, r) \cdot \Lambda_{\mathrm{SK}}(e) \rangle && : C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}, \text{randomness } r \\
&\equiv \langle C, \Lambda_{\mathrm{SK}}(C) \cdot \Lambda_{\mathrm{SK}}(e) \rangle && : C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}, \text{via projective property} \\
&\approx_c \langle C, \Lambda_{\mathrm{SK}}(C) \cdot \Lambda_{\mathrm{SK}}(e) \rangle && : C \leftarrow_{\mathrm{R}} \mathcal{G}, \text{via subgroup membership} \\
&\equiv \langle C, \Lambda_{\mathrm{SK}}(C \cdot e) \rangle && : C \leftarrow_{\mathrm{R}} \mathcal{G}, \text{since } \Lambda_{\mathrm{SK}}(\cdot) \text{ is homomorphic} \\
&\equiv \langle C \cdot e^{-1}, \Lambda_{\mathrm{SK}}(C) \rangle && : C \leftarrow_{\mathrm{R}} \mathcal{G}, \text{since } e \in \mathcal{G} \\
&\approx_c \langle C \cdot e^{-1}, \Lambda_{\mathrm{SK}}(C) \rangle && : C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}} \\
&\equiv \langle C \cdot e^{-1}, \mathsf{Pub}(\mathrm{PK}, C, r) \rangle && : C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}, \text{randomness } r, \text{ via projective}
\end{aligned}$$

Note that:

- we can sample from the final distribution given only PK;
- the above transition does not rely on smoothness, and therefore everything goes through even if we append SK to the view, namely $(\mathrm{SK}, \mathsf{Enc}(\Lambda_{\mathrm{SK}}(e))) \approx_c (\mathrm{SK}, \langle C \cdot e^{-1}, \mathsf{Pub}(\mathrm{PK}, C, r) \rangle)$, which allows us to carry out a hybrid argument over the KDM queries;

[1] as noted earlier in the introduction, we only address KDM-security in this paper with a single public/secret key; we address multiple keys later in this section.

- the treatment of KDM queries relies on the projective and homomorphic properties of $\Lambda_{\textsc{sk}}(\cdot)$ but not smoothness; instead, we will use smoothness for the normal encryption queries.

Again, we stress that the proof crucially exploits the projective property; the role of the projective property is not captured by any of the prior "expanded system", "interactive vector" or the "triple proofs" frameworks for KDM-security in [9, 10, 29].

An instantiation. In the BHHO DDH-based KDM-secure encryption scheme, the underlying projective hash function is defined on a group $\mathcal{G} := \mathbb{G}^\ell$ where $\mathbb{G}$ is the DDH group with some generator g, and ℓ is a parameter. The hashing key (also the secret key) $\textsc{sk} = (s_1,\dots,s_\ell)$ lies in $\{0,1\}^\ell$, and given an instance $C = (c_1,\dots,c_\ell) \in \mathbb{G}^\ell$,

$$\Lambda_{\textsc{sk}}^{\mathsf{BHHO}}(C) = c_1^{s_1} \cdot c_2^{s_2} \cdots c_\ell^{s_\ell}$$

This means that given any $(a_1,\dots,a_\ell) \in \mathbb{Z}_q^\ell$,

$$\Lambda_{\textsc{sk}}^{\mathsf{BHHO}}((g^{a_1},\dots,g^{a_\ell})) = g^{a_1 s_1 + \cdots + a_\ell s_\ell}$$

Average-case smoothness follows readily from the left-over hash lemma. Now, if we modify the underlying Cramer-Shoup scheme to encrypt the message in the exponent, this function corresponds precisely to linear functions of the bits of the secret key. To handle affine functions, we need to handle an additional offset as described in Section 4.

Moreover, we can further extend the hash proof system to handle KDM-security with respect to some fixed functions $f_1,\dots,f_t$ for any polynomial t (for instance, constant-degree polynomials in the bits of the secret keys or uniform Turing machine computation of description at most $c\log k$ bits) as is the setting considered in Brakerski, Goldwasser and Kalai [11]. We now consider instances $C = (c_1,\dots,c_{\ell+t}) \in \mathbb{G}^{\ell+t}$,

$$\Lambda_{\textsc{sk}}^{\mathsf{BHHO}}(C) = c_1^{s_1} \cdot c_2^{s_2} \cdots c_\ell^{s_\ell} \cdot c_{\ell+1}^{f_1(\textsc{sk})} \cdots c_{\ell+t}^{f_t(\textsc{sk})}$$

Average-case smoothness follows as before from the left-over hash lemma. Then, $\Lambda_{\textsc{sk}}^{\mathsf{BHHO}}(g^{\mathbf{e}_{\ell+i}}) = g^{f_i(\textsc{sk})}$ corresponds to an encryption of $f_i(\textsc{sk})$. This provides a more direct construction of KDM-security with respect to $f_1,\dots,f_t$ as opposed to the entropic-KDM framework in [11].

On KDM-security with multiple keys. We clarify that we only address KDM-security in this paper with a single public/secret key, whereas the previous constructions in [9, 10] address KDM-security with multiple public/secret key pairs. We note that simplifying KDM-security for a single public/secret key is still important in and of itself: (1) it suffices for some applications, e.g. disk encryption, (2) it already captures much of the technical difficulty in realizing KDM-security, (3) previous schemes in [9, 4, 10] first establish KDM-security for a single public/secret key, and then bootstrap to multiple keys, (4) more recent schemes for RKA-KDM-security in [8] also reduces security to KDM-security for a single public/secret key. In particular, our framework clarifies the first step of the analysis for multiple key pairs; our framework is also the first to point out the role of the projective property for KDM-security (which is not covered in prior "expanded system", "interactive vector" or the "triple proofs" frameworks for KDM-security in [9, 10, 29]) and that captures the algebraic structure needed for the decryption algorithm and the space of ciphertexts via homomorphic projective hashing.

Connection to leakage resilience. Let us informally refer to a Cramer-Shoup scheme as "linear" if $\Lambda_{\textsc{sk}}(\cdot)$ computes a linear function of $\textsc{sk}$ (possibly in the exponent), where the coefficients of the linear function are specified by the instance. From the preceding discussion, we see that (1) linear Cramer-Shoup schemes are KDM-secure w.r.t. linear functions, and (2) the BHHO scheme [9] along with the BHHO-like schemes given by Brakerski and Goldwasser [10] are examples of such schemes. Naor and Segev [31] also showed that linear Cramer-Shoup schemes are resilient to bounded key leakage; this follows from the fact that random linear functions are good strong extractors. This yields a simple explanation as to why the BHHO scheme and variants there-of are both KDM-secure and resilient to bounded key leakage.

2.2 Dual-mode encryption

Starting with a smooth projective hash function $\Lambda_{\text{HK}}(\cdot)$ defined on $\mathcal{G}$, we can build a different CPA-secure encryption scheme —which we will refer to as the "dual Cramer-Shoup scheme"— as follows:

- $\mathsf{Gen}(1^k)$: Sample $C \leftarrow_{\text{R}} \mathcal{G}_{\text{YES}}$ with randomness r and output the key pair
$$\text{PK} := C \quad \text{and} \quad \text{SK} := r$$
- $\mathsf{Enc}(\text{PK}, m)$: To encrypt a message m, sample a random HK and output the ciphertext
$$(\mu(\text{HK}), \Lambda_{\text{HK}}(C) \cdot m)$$
- $\mathsf{Dec}(\text{SK}, (p, \psi))$: On input a ciphertext (p, ψ), compute $K := \Lambda_{\text{HK}}(C)$ using Pub on input p, C and r (via the projective property) and output
$$(K^{-1} \cdot \psi)$$

As observed in by Halevi and Kalai [23, 24], if we sample the public key $C \leftarrow_{\text{R}} \mathcal{G}_{\text{NO}}$, smoothness tells us that we obtain a "messy" public key where the ciphertext carries no information about the message. This suggests the following natural construction of a dual-mode cryptosystem / OT protocol:

- the receiver generates a pair of public keys $C_0, C_1 \in \mathcal{G}$ subject to the constraint that $C_0 \cdot C_1$ is the CRS.
- in the normal mode, we pick $C_0, C_1 \leftarrow_{\text{R}} \mathcal{G}_{\text{YES}}$, and the CRS is chosen uniformly from $\mathcal{G}_{\text{YES}}$.
- in the messy mode, the CRS is chosen uniformly from $\mathcal{G}_{\text{NO}}$. Now, whenever a possibly malicious receiver sends a pair of public keys (C_0, C_1) such that $C_0 \cdot C_1 \in \mathcal{G}_{\text{NO}}$, then we know that one of C_0, C_1 lies in $\mathcal{G}_{\text{NO}}$ and is therefore messy. (Otherwise, if $C_0, C_1 \in \mathcal{G}_{\text{YES}}$, then $C_0 \cdot C_1 \in \mathcal{G}_{\text{YES}}$ by closure properties of the subgroup.)

We note that exploiting subgroup structure of $\mathcal{G}_{\text{YES}}$ appears to be novel to this work, and we use subgroup structure in two ways: first, to argue that if $C_0 \cdot C_1 \in \mathcal{G}_{\text{NO}}$, then one of C_0, C_1 lies in $\mathcal{G}_{\text{NO}}$; and second, randomizing $\mathcal{G}_{\text{YES}}$ in the CRS (which is necessary for reusability in the context of UC security) by adding another random $\mathcal{G}_{\text{YES}}$ instance. In contrast, the prior work [23] uses the fact that if two pairs of group elements agree on the first component and disagree on the second, then one of them is a non-DDH tuple, and there is no need for randomizing $\mathcal{G}_{\text{YES}}$ as it addresses stand-alone security.

2.3 Discussion

On lattice-based instantiations. A natural question is whether our frameworks extend to LWE-based instantiations of KDM-secure encryption and dual-mode cryptosystems given in [4, 2, 32], while relying on an approximate notions of smooth projective hashing as given in [27]. In the LWE setting, the "yes" instances as given by valid LWE instances do not form a subgroup. We note that for KDM security, our proof does not rely on the fact that $\mathcal{G}_{\text{YES}}$ forms a subgroup. For dual-mode cryptosystems, we only require that the "product" of two instances in $\mathcal{G}_{\text{YES}}$ is "far" from $\mathcal{G}_{\text{NO}}$, which is indeed satisfied by LWE instances. However, in order to obtain an OT protocol where the same CRS can be reused for an a-priori unbounded number executions, it is crucial that we can statistically rerandomize instances in $\mathcal{G}_{\text{YES}}$. We do not know how to achieve the latter for LWE; indeed, the LWE-based OT in [32] only achieves security for an a-priori bounded number of OT executions. In particular, we do not know any LWE instantiations for the "full-fledged" notion of dual-mode cryptosystems.

Additional related work. Smooth projective hashing is an extremely versatile tool that has found many other applications beyond CCA-security – two-message oblivious transfer [23], password-authenticated key exchange [18, 6], bounded leakage resilience [31], and encryption schemes secure against selective opening attacks [24]. The works of Barak et al. and Applebaum [5, 3], Brakerski, Goldwasser and Kalai [11], and Malkin, Teranishi and Yung [29] each presented general and different techniques to extend KDM-security to richer classes of functions with incomparable trade-offs. Haitner and Holenstein [22] presented black-box impossibility results for (single-key) KDM-security based on general assumptions. In subsequent work, Hofheinz [25] presented a KDM-CCA-secure scheme with compact ciphertexts, inspired in part by the connection between smooth projective hashing and KDM-security established in this work.

Organization. We present definition and results on KDM-secure public-key encryption in Section 4, and those for dual-mode encryption in Section 5. We present the instantiations in Sections 6 and 7.

3 Preliminaries

Notation. We denote by $s \leftarrow_{\text{R}} S$ the fact that s is picked uniformly at random from a finite set S and by $x, y, z \leftarrow_{\text{R}} S$ that all x, y, z are picked independently and uniformly at random from S. By PPT, we denote a probabilistic polynomial-time algorithm. Throughout, we use 1^k as the security parameter. We use $\cdot$ to denote multiplication (or group operation) as well as component-wise multiplication. We use lower case boldface to denote (column) vectors and upper case boldface to denote matrices.

3.1 Smooth Projective Hashing

We present the notion of smooth projective hashing as introduced by Cramer and Shoup [16], in the context of group-theoretic languages.

Setup. Fix a family of groups $\mathcal{G}_{\text{PP}}$ indexed by a public parameter PP. We require that PP be efficiently samplable along with a secret parameter SP given a security parameter 1^k, and assume that all algorithms are given PP as part of its input. We omit PP henceforth whenever the context is clear. We consider subgroups $\mathcal{G}_{\text{YES}}$ of $\mathcal{G}$ and we use $\mathcal{G}_{\text{NO}}$ to denote $\mathcal{G} \setminus \mathcal{G}_{\text{YES}}$. We will require that each of these groups $\mathcal{G}, \mathcal{G}_{\text{YES}}, \mathcal{G}_{\text{NO}}$ be efficiently samplable given PP. Observe that if $\mathcal{G}_{\text{YES}}$ has negligible density (as is the case for most instantiations), we may use the same sampling algorithm for both $\mathcal{G}$ and $\mathcal{G}_{\text{NO}}$ since both distributions are statistically indistinguishable.

Subgroup membership assumption. We will consider two related computational assumptions pertaining to the group $\mathcal{G}$, which we refer to collectively as the *subgroup membership assumption*. The first assumption states that the uniform distributions over $\mathcal{G}_{\text{YES}}$ and $\mathcal{G}$ are computationally indistinguishable, even given PP. The second assumption states that the uniform distributions over $\mathcal{G}_{\text{YES}}$ and $\mathcal{G}_{\text{NO}}$ are computationally indistinguishable, even given PP. Again, observe that if $\mathcal{G}_{\text{YES}}$ has negligible density, these two assumptions are equivalent, since the distributions over $\mathcal{G}$ and $\mathcal{G}_{\text{NO}}$ are then statistically indistinguishable. Finally, we require that given the secret parameter SP, we can efficiently verify membership in $\mathcal{G}_{\text{YES}}$.

Homomorphic projective hashing. Fix a public parameter PP. We consider a family of hash functions $\{\Lambda_{\text{HK}} : \mathcal{G} \to \mathcal{K}\}$ indexed by a hashing key HK. We require that $\Lambda_{\text{HK}}(\cdot)$ be efficiently computable (by a 'private evaluation' algorithm), and HK be efficiently samplable. In addition, we require that both $\mathcal{G}$ and $\mathcal{K}$ are groups, and that $\Lambda_{\text{HK}}(\cdot)$ is a group *homomorphism*, that is, for all HK and all $C_0, C_1 \in \mathcal{G}$, we have $\Lambda_{\text{HK}}(C_0) \cdot \Lambda_{\text{HK}}(C_1) = \Lambda(C_0 \cdot C_1)$. We say that $\Lambda_{\text{HK}}(\cdot)$ is *projective* if there exists a projection map $\mu(\cdot)$ defined on HK such that $\mu(\text{HK})$ defines the action of Λ_{HK} on inputs from $\mathcal{G}_{\text{YES}}$. Specifically, we require that there exists an efficient public evaluation algorithm Pub that on input $\mu(\text{HK})$ and $C \in \mathcal{G}_{\text{YES}}$ along with the randomness r used to sample C, outputs the value $\Lambda_{\text{HK}}(C)$.

Smoothness. We say that $\Lambda_{\text{HK}}(\cdot)$ is *smooth* if the action of Λ_{HK} on $\mathcal{G}_{\text{NO}}$ is completely undetermined. That is, for all $C \in \mathcal{G}_{\text{NO}}$, the following distributions are statistically close:

$$\langle \text{PK}, \Lambda_{\text{HK}}(C) \rangle \quad \text{and} \quad \langle \text{PK}, K \rangle$$

where HK is random, $\text{PK} = \mu(\text{HK})$ and $K \leftarrow_{\text{R}} \mathcal{K}$. (Looking ahead, we will also consider a relaxed notion in some of our instantiations where we choose K from the uniform distribution over some subset of $\mathcal{K}$; see Section 7.) We also say that $\Lambda_{\text{HK}}(\cdot)$ is *average-case smooth* where we relax the requirement for smoothness to hold for a random $C \in \mathcal{G}$ [31]. That is, the following distributions are statistically close:

$$\langle C, \text{PK}, \Lambda_{\text{HK}}(C) \rangle \quad \text{and} \quad \langle C, \text{PK}, K \rangle$$

where HK is random, $\text{PK} = \mu(\text{HK})$, $C \leftarrow_{\text{R}} \mathcal{G}$ and $K \leftarrow_{\text{R}} \mathcal{K}$.

4 KDM-Secure Encryption

Key-Dependent Message Security. We adopt a simulation-based variant of key-dependent message (KDM) security from [7, 9], in the setting where there is only one public/secret key pair. Fix a public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. For a stateful adversary $\mathcal{A}$, we define the advantage function

$$\mathsf{AdvKDM}^{\mathcal{A},\mathcal{F}}(k) := \Pr\left[\begin{array}{l}(\text{PK}, \text{SK}) \leftarrow \mathsf{Gen}(1^k);\\ \mathcal{A}^{\mathsf{kdmEnc}(\cdot), \mathsf{Enc}(\text{PK},\cdot)}(\text{PK}) = 1\end{array}\right] - \Pr\left[\begin{array}{l}(\text{PK}, \text{SK}) \leftarrow \mathsf{Gen}(1^k);\\ \mathcal{A}^{\mathsf{kdmEnc}^*(\cdot), \mathsf{Enc}^*(\text{PK},\cdot)}(\text{PK}) = 1\end{array}\right]$$

where

- $\mathsf{kdmEnc}(\cdot)$ is an oracle that on input $f \in \mathcal{F}$ returns a random encryption $\mathsf{Enc}(\text{PK}, f(\text{SK}))$;
- $\mathsf{kdmEnc}^*(\text{PK}, \cdot)$ corresponds to a simulator that gets as input $f \in \mathcal{F}$;
- $\mathsf{Enc}^*(\text{PK}, \cdot)$ is an oracle that on input m, returns $\mathsf{Enc}(\text{PK}, 0^{|m|})$.

An encryption scheme is said to be *$\mathcal{F}$-KDM secure* if there exists an efficient $\mathsf{kdmEnc}^*()$ such that for all PPT $\mathcal{A}$, the advantage $|\mathsf{AdvKDM}^{\mathcal{A},\mathcal{F}}(k)|$ is a negligible function in k.

Construction. Starting with a projective hash function $\Lambda_{\text{HK}} : \mathcal{G} \to \mathcal{K}$, we may derive a semantically secure public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. The message space is $\mathcal{M}$, and we require an injective map $\phi : \mathcal{M} \to \mathcal{K}$ which is efficiently computable and invertible.

- $\mathsf{Gen}(1^k)$: Sample public parameters PP, a uniform hashing key HK and compute $\mathrm{PK} := (\mathrm{PP}, \mu(\mathrm{HK}))$. Output the key pair
$$\mathrm{PK} := (\mathrm{PP}, \mu(\mathrm{HK})) \quad \text{and} \quad \mathrm{SK} := \mathrm{HK}$$
- $\mathsf{Enc}(\mathrm{PK}, m)$: Sample $C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}$ with randomness r, output the ciphertext
$$(C, \mathsf{Pub}(\mathrm{PK}, C, r) \cdot \phi(m))$$
- $\mathsf{Dec}(\mathrm{SK}, (C, \psi))$: Output the plaintext
$$\phi^{-1}(\Lambda_{\mathrm{SK}}(C)^{-1} \cdot \psi)$$

Theorem 1. *Suppose $\Lambda_{\mathrm{HK}}(\cdot)$ is a projective hash function that is average-case smooth and homomorphic, and the subgroup membership problem is hard (w.r.t. $\mathcal{G}$ vs $\mathcal{G}_{\mathrm{YES}}$). Then, the encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *described above is $\mathcal{F}$-KDM secure where $\mathcal{F} = \{f_{e,k} : \mathrm{SK} \mapsto \phi^{-1}(\Lambda_{\mathrm{SK}}(e) \cdot k) \mid e \in \mathcal{G}, k \in \mathcal{K}\}$.*

We do require that given a description of the function $f_{e,k}$, we can efficiently compute the corresponding $e \in \mathcal{G}, k \in \mathcal{K}$. Later on in the instantiations, the term e allows us to specify the coefficients in a linear function, whereas k corresponds to the constant off-set in an affine function. On the first reading, we suggest that the reader assume ϕ is the identity map.

Proof. Observe that correctness of the encryption scheme follows readily from the projective property. We proceed to establish KDM security. First, we describe kdmEnc^*: on input $\mathrm{PK}, f_{e,k}$ and randomness r, use r to sample $C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}$ and output
$$\langle C \cdot e^{-1}, \mathsf{Pub}(\mathrm{PK}, C, r) \cdot k \rangle$$
We proceed via a sequence of games. Fix a PPT adversary $\mathcal{A}$ that makes at most Q_0 queries to kdmEnc and Q_1 queries to Enc. We show that
$$|\mathsf{AdvKDM}^{\mathcal{A},\mathcal{F}}(k)| \leq (2Q_0 + 2Q_1) \cdot \epsilon$$
where ϵ is the advantage for the subgroup membership assumption. We start with Game 0, where the challenger proceeds like in the security game with $\mathsf{kdmEnc}, \mathsf{Enc}$ oracles in the left experiment and $\mathsf{kdmEnc}^*, \mathsf{Enc}^*$ oracles in the right experiment.

Game 1. For $i = 1, \ldots, Q_0$, replace the i'th query $f_{e,k}$ to kdmEnc on the left with kdmEnc^*. We will run a hybrid argument over the Q_0 queries, and thus it suffices to show that for each i,
$$(\mathrm{PK}, \mathrm{SK}, \mathsf{Enc}(\mathrm{PK}, f_{e,k}(\mathrm{SK}))) \overset{2\epsilon}{\approx_c} (\mathrm{PK}, \mathrm{SK}, \langle C \cdot e^{-1}, \mathsf{Pub}(\mathrm{PK}, C, r) \cdot k \rangle),$$
where we would use SK to simulate the remaining kdmEnc queries and PK for the Enc queries. For notational simplicity, we omit $(\mathrm{PK}, \mathrm{SK})$ in the hybrid transitions below:
$$\begin{array}{lll}
 & \mathsf{Enc}(\mathrm{PK}, f_{e,k}(\mathrm{SK}); r) & \\
\equiv & \langle C, \mathsf{Pub}(\mathrm{PK}, C, r) \cdot \Lambda_{\mathrm{SK}}(e) \cdot k \rangle & : C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}, \text{randomness } r \\
\equiv & \langle C, \Lambda_{\mathrm{SK}}(C) \cdot \Lambda_{\mathrm{SK}}(e) \cdot k \rangle & : C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}, \text{via projective property} \\
\approx_c & \langle C, \Lambda_{\mathrm{SK}}(C) \cdot \Lambda_{\mathrm{SK}}(e) \cdot k \rangle & : C \leftarrow_{\mathrm{R}} \mathcal{G}, \text{via subgroup membership} \\
\equiv & \langle C, \Lambda_{\mathrm{SK}}(C \cdot e) \cdot k \rangle & : C \leftarrow_{\mathrm{R}} \mathcal{G}, \text{since } \Lambda_{\mathrm{SK}}(\cdot) \text{ is homomorphic} \\
\equiv & \langle C \cdot e^{-1}, \Lambda_{\mathrm{SK}}(C) \cdot k \rangle & : C \leftarrow_{\mathrm{R}} \mathcal{G}, \text{since } e \in \mathcal{G} \\
\approx_c & \langle C \cdot e^{-1}, \Lambda_{\mathrm{SK}}(C) \cdot k \rangle & : C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}} \\
\equiv & \langle C \cdot e^{-1}, \mathsf{Pub}(\mathrm{PK}, C, r) \cdot k \rangle & : C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}, \text{randomness } r, \text{ via projective}
\end{array}$$

Note that the above transition does not rely on smoothness, and therefore everything goes through even if we append $(\mathrm{PK},\mathrm{SK})$ to the view.

Game 2. For $i = 1,\ldots,Q_1$, replace the i'th query m to Enc on the left with Enc^*. We will run a hybrid argument over the Q_1 queries, and thus it suffices to show that for each i,

$$(\mathrm{PK},\mathsf{Enc}(\mathrm{PK},m)) \overset{2\epsilon}{\approx_c} (\mathrm{PK},\mathsf{Enc}(\mathrm{PK},0^{|m|})).$$

This is standard CPA-security of the Cramer-Shoup encryption. Observe that the view includes PK, which is sufficient to run kdmEnc^*.

$$\begin{aligned}
\mathsf{Enc}(\mathrm{PK},m) &\equiv \langle C,\mathsf{Pub}(\mathrm{PK},C,r)\cdot\phi(m)\rangle &&: C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}, \text{randomness } r\\
&\equiv \langle C,\Lambda_{\mathrm{SK}}(C)\cdot\phi(m) &&: C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}, \text{via projective property}\\
&\approx_c \langle C,\Lambda_{\mathrm{SK}}(C)\cdot\phi(m) &&: C \leftarrow_{\mathrm{R}} \mathcal{G}, \text{via subgroup membership}\\
&\equiv \langle C,K\cdot\phi(m)\rangle &&: C \leftarrow_{\mathrm{R}} \mathcal{G}, K \leftarrow_{\mathrm{R}} \mathcal{K}, \text{via smoothness}\\
&\equiv \langle C,K\cdot\phi(0^{|m|})\rangle &&: C \leftarrow_{\mathrm{R}} \mathcal{G}, K \leftarrow_{\mathrm{R}} \mathcal{K}, \text{via uniformity of } K\\
&\approx_c \mathsf{Enc}(\mathrm{PK},0^{|m|})) && \text{by reversing the hybrids}
\end{aligned}$$

We conclude by observing that in Game 2, the left and right experiments are identical (both use the $\mathsf{kdmEnc}^*,\mathsf{Enc}^*$ oracles), and therefore the advantage is 0. □

5 Dual-Mode Encryption

In this section, we present the definition of a dual-mode cryptosystem from [32], and show a generic construction from smooth projective hashing. By [32, Theorem 4.1], once we have a dual-mode cryptosystem, we immediately obtain UC-secure two-message oblivious transfer in the CRS model.

Preliminaries. Most of this is copied verbatim from [32, Section 3].

- $\mathsf{Setup}(1^k,\mu)$: given security parameter 1^k and mode $\mu \in \{0,1\}$, outputs (CRS,τ). The CRS is a common string for the remaining algorithms, and τ is a trapdoor value that enables either the $\mathsf{FindMessy}$ or $\mathsf{TrapKeyGen}$ algorithm, depending on the selected algorithm. We will also denote the messy setup algorithm using $\mathsf{SetupMessy}(\cdot) := \mathsf{Setup}(\cdot,0)$ and the decryption mode setup algorithm using $\mathsf{SetupDec}(\cdot) := \mathsf{Setup}(\cdot,1)$. All the remaining algorithms take CRS as their first input, but for notational clarity, we usually omit it from the list of arguments.
- $\mathsf{KeyGen}(\sigma)$: given a desired decryptable branch value $\sigma \in \{0,1\}$, outputs $(\mathrm{PK},\mathrm{SK})$ where PK is a public encryption key and SK is a corresponding secret key for messages encrypted on branch σ.
- $\mathsf{Enc}(\mathrm{PK},b,m)$: given a public key PK, a branch value $b \in \{0,1\}$, and a message $m \in \{0,1\}^\ell$, outputs a ciphertext c encrypted on branch b.
- $\mathsf{Dec}(\mathrm{SK},\psi)$: given a secret key SK and a ciphertext ψ, outputs a message $m \in \{0,1\}^\ell$.
- $\mathsf{FindMessy}(\tau,\mathrm{PK})$: given a trapdoor τ for CRS generated in messy mode and some (possibly malformed) public key PK, outputs a branch value $b \in \{0,1\}$ corresponding to a messy branch of PK.
- $\mathsf{TrapKeyGen}(\tau)$: given a trapdoor τ for CRS generated in decryption mode, outputs $(\mathrm{PK},\mathrm{SK}_0,\mathrm{SK}_1)$ where PK is a public encryption key and $\mathrm{SK}_0,\mathrm{SK}_1$ are corresponding secret decryption keys for branches 0 and 1 respectively.

Definition 1 (Dual-Mode Encryption). *A* dual-mode cryptosystem *is a tuple of algorithms described above that satisfy the following properties:*

1. *Completeness for decryptable branch: For every $\mu \in \{0,1\}$, every $(\textsc{crs},\tau) \leftarrow \mathsf{Setup}(1^k,\mu)$, every $\sigma \in \{0,1\}$, every $(\textsc{pk},\textsc{sk}) \leftarrow \mathsf{KeyGen}(\sigma)$ and every $m \in \{0,1\}^\ell$, decryption is correct on branch σ, i.e.* $\mathsf{Dec}(\textsc{sk},\mathsf{Enc}(\textsc{pk},\sigma,m)) = m$.
2. *Indistinguishability of modes: the first outputs of* $\mathsf{SetupMessy}$ *and* $\mathsf{SetupDec}$ *are computationally indistinguishable, i.e.* $\mathsf{SetupMessy}_1(1^k) \approx_c \mathsf{SetupDec}_1(1^k)$.
3. *(Messy Mode) Trapdoor identification of messy branch: For every $(\textsc{crs},\tau) \leftarrow \mathsf{SetupMessy}(1^k)$ and every (possibly malformed) $\textsc{pk}$, $\mathsf{FindMessy}(\tau,\textsc{pk})$ outputs a branch value $b \in \{0,1\}$ such that $\mathsf{Enc}(\textsc{pk},b,\cdot)$ is messy. Namely, for every $m_0,m_1 \in \{0,1\}^\ell$, $\mathsf{Enc}(\textsc{pk},b,m_0) \approx_s \mathsf{Enc}(\textsc{pk},b,m_1)$.*
4. *(Decryption Mode) Trapdoor generation of keys decryptable on both branches: For every $(\textsc{crs},\tau) \leftarrow \mathsf{SetupDec}(1^k)$, $\mathsf{TrapKeyGen}(\tau)$ outputs $(\textsc{pk},\textsc{sk}_0,\textsc{sk}_1)$ such that for every $\sigma \in \{0,1\}$: $(\textsc{pk}) \approx_s \mathsf{KeyGen}(\sigma)_1$ and $(\textsc{pk},\textsc{sk}_\sigma) \in \mathsf{Supp}(\mathsf{KeyGen}(\sigma))$.*

Remark 1. Our requirement for decryption mode is actually weaker than that in [32], which stipulates that for every $\sigma \in \{0,1\}$, $(\textsc{pk},\textsc{sk}_\sigma) \approx_s \mathsf{KeyGen}(\sigma)$. That is, we allow $\mathsf{TrapKeyGen}$ output any valid secret key $\textsc{sk}_\sigma$ for branch σ, whereas the original requirement is that the distribution of $\textsc{sk}_\sigma$ be close to that output by $\mathsf{KeyGen}(\sigma)$. This weaker guarantee is nonetheless sufficient for UC-secure OT, since the decryption mode is used in the case of a corrupted sender. A corrupted sender sees only $\textsc{pk}$ and not $\textsc{sk}_0$ or $\textsc{sk}_1$; moreover, as long as both $\textsc{sk}_0$ and $\textsc{sk}_1$ are valid, we will be able to extract both of its inputs.

Dual-mode encryption from projective hashing. We begin with the set-up algorithms:

- $\mathsf{SetupMessy}(1^k)$: Run $\mathsf{Param}(1^k) \leftarrow (\textsc{pp},\textsc{sp})$ and sample $C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{NO}}$. Output
$$\textsc{crs} := (\textsc{pp},C) \quad \text{and} \quad \tau := \textsc{sp}$$
- $\mathsf{SetupDec}(1^k)$: Run $\mathsf{Param}(1^k) \leftarrow (\textsc{pp},\textsc{sp})$ and sample $C \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}$ with randomness r. Output
$$\textsc{crs} := (\textsc{pp},C) \quad \text{and} \quad \tau := r$$

All the remaining algorithms take $\textsc{crs} = (\textsc{pp},C)$ where $C \in \mathcal{G}$ as their first input.

- $\mathsf{KeyGen}(\sigma)$: On input a branch value $\sigma \in \{0,1\}$, sample $C_\sigma \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}$ with randomness r_σ. Set $C_{1-\sigma} := C \cdot C_\sigma^{-1}$. Output
$$\textsc{pk} := (C_0,C_1) \quad \text{and} \quad \textsc{sk} := (\sigma, r_\sigma)$$
- $\mathsf{Enc}(\textsc{pk},b,m)$: On input $\textsc{pk} = (C_0,C_1)$, sample a uniform hashing key $\textsc{hk}$ and output
$$\psi := (\mu(\textsc{hk}), \Lambda_{\textsc{hk}}(C_b) \cdot m)$$
- $\mathsf{Dec}(\textsc{sk},\psi)$: On input $\textsc{sk} = (\sigma,r)$ and $\psi = (\textsc{pk}^*,\psi^*)$, output
$$m := \mathsf{Pub}(\textsc{pk}^*,C_\sigma,r)^{-1} \cdot \psi^*$$
- $\mathsf{FindMessy}(\tau,\textsc{pk})$: On input $\tau = \textsc{sp}$ and $\textsc{pk} = (C_0,C_1)$, check that $C_0 \cdot C_1 = C$. Output
$$b := \begin{cases} 1 & \text{if } C_0 \in \mathcal{G}_{\mathrm{YES}} \\ 0 & \text{otherwise} \end{cases}$$
- $\mathsf{TrapKeyGen}(\tau)$: On input $\tau = r$, sample $C_0 \leftarrow_{\mathrm{R}} \mathcal{G}_{\mathrm{YES}}$ with randomness r_0 and compute $C_1 \in \mathcal{G}_{\mathrm{YES}}$ with randomness $r_1 := r - r_0$ (so that $C_0 \cdot C_1 = C$). Output
$$\textsc{pk} := (C_0,C_1) \quad \text{and} \quad (\textsc{sk}_0,\textsc{sk}_1) := (r_0,r_1)$$

Theorem 2. *Suppose $\Lambda_{\text{HK}}(\cdot)$ is a smooth projective hash function, and the subgroup membership problem is hard (w.r.t. $\mathcal{G}_{\text{YES}}$ vs $\mathcal{G}_{\text{NO}}$). Then, the above construction yields a dual-mode cryptosystem.*

We note here that our construction requires an additional property from underlying group, namely that given the respective randomness r_0, r_1 for sampling $C_0, C_1 \in \mathcal{G}_{\text{YES}}$, the value $r_0 + r_1$ is the randomness for sampling $C_0 \cdot C_1$ (that is, the sampling algorithm is also homomorphic). This requirement may be eliminated if we are willing to settle for the weaker guarantee where each CRS may only be used for a single (or a-priori bounded) instance of OT, as with the LWE-based instantiation in [32].

Proof. We verify that our construction satisfies all of the four properties in Definition 1:

1. Completeness for decryptable branch: This follows readily from the projective property.
2. Indistinguishability of modes: This follows readily from our subset membership assumption.
3. (Messy Mode) Trapdoor identification of messy branch: In the messy mode, we require that $C_0 \cdot C_1 = C \in \mathcal{G}_{\text{NO}}$. Therefore, (at least) one of $C_0, C_1 \in \mathcal{G}_{\text{NO}}$ (a subgroup is closed under multiplication, so if $C_0, C_1 \in \mathcal{G}_{\text{YES}}$, then $C_0 \cdot C_1 \in \mathcal{G}_{\text{YES}}$). Moreover, using the membership trapdoor, we can identify which of C_0 or C_1 is in $\mathcal{G}_{\text{NO}}$. The corresponding ciphertext must be messy by smoothness.
4. (Decryption Mode) Trapdoor generation of keys decryptable on both branches: It is clear that the distribution of each of C_0 and C_1 is the uniform distribution over $\mathcal{G}_{\text{YES}}$. Moreover, r_0 and r_1 are randomness used for sampling C_0 and C_1 respectively. Therefore, by the projective property, we can decrypt ciphertexts on both branches. □

6 Instantiations from DLIN

Let $\mathbb{G}$ be a group of prime order q specified using a generator g. The DDH assumption asserts that g^{ab} is pseudorandom given g, g^a, g^b where $g \leftarrow_{\text{R}} \mathbb{G}; a, b \leftarrow_{\text{R}} \mathbb{Z}_q$. The d-LIN assumption asserts that $g_{d+1}^{r_1+\cdots+r_d}$ is pseudorandom given $g_1, \ldots, g_{d+1}, g_1^{r_1}, \ldots, g_d^{r_d}$ where $g_1, \ldots, g_{d+1} \leftarrow_{\text{R}} \mathbb{G}; r_1, \ldots, r_d \leftarrow_{\text{R}} \mathbb{Z}_q$. DDH is equivalent to 1-LIN.

6.1 Dual-mode encryption

For dual-mode encryption, we use the original Cramer-Shoup DDH-based hash proof system in [16, 15] and its generalization to d-LIN [26, 34].

Setup. Sample $\mathbf{P} \leftarrow_{\text{R}} \mathbb{Z}_q^{d\times(d+1)}$ along with a check vector $\mathbf{v} \neq \mathbf{0}$ so that $\mathbf{P}\mathbf{v} = \mathbf{0}$. Output

$$\text{PP} := (\mathbb{G}, q, g, g^{\mathbf{P}}) \quad \text{and} \quad \text{SP} := (\mathbf{v})$$

The subgroup indistinguishability problem is given by:

$$\mathcal{G}_{\text{YES}} := \left\{ g^{\mathbf{r}^\top \mathbf{P}} : \mathbf{r} \in \mathbb{Z}_q^d \right\} \qquad \text{and} \qquad \mathcal{G} := \left\{ g^{\mathbf{a}^\top} : \mathbf{a} \in \mathbb{Z}_q^{d+1} \right\}$$

where $\mathsf{SampR}(\mathbf{r}) = g^{\mathbf{r}^\top \mathbf{P}}$ and the group operation is the natural one given by entry-wise product. The uniform distributions over $\mathcal{G}_{\text{YES}}$ and $\mathcal{G}$ are computationally distinguishable under the d-LIN assumption as shown in [31, 9]. Observe that we can efficiently verify membership in $\mathcal{G}_{\text{YES}}$ using $\mathbf{v}$ since:

$$g^{\mathbf{a}^\top} \in \mathcal{G}_{\text{YES}} \iff g^{\mathbf{a}^\top \mathbf{v}} = 1$$

Hashing. The hashing key is given by a column vector $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{d+1}$, with

$$\mu(g^{\mathbf{P}}, \mathbf{s}) := g^{\mathbf{Ps}} \in \mathbb{G}^{d \times 1}$$

Private and public evaluation are given by:

$$\Lambda_{\mathbf{s}}(g^{\mathbf{a}^\top}) := g^{\mathbf{a}^\top \mathbf{s}} \in \mathbb{G} \qquad \text{and} \qquad \mathsf{Pub}(g^{\mathbf{Ps}}, \mathbf{C}, \mathbf{r}) := g^{\mathbf{r}^\top (\mathbf{Ps})}$$

Clearly, $\Lambda_{\mathbf{s}}(\cdot)$ is a group homomorphism. For the projective property, observe that for $\mathbf{C} = g^{\mathbf{r}^\top \mathbf{P}} \in \mathcal{G}_{\mathrm{YES}}$, we have

$$\Lambda_{\mathbf{s}}(\mathbf{C}) = g^{\mathbf{r}^\top \mathbf{Ps}} = \mathsf{Pub}(g^{\mathbf{Ps}}, \mathbf{C}, \mathbf{r})$$

Smoothness. Observe that for any $g^{\mathbf{a}^\top} \in \mathcal{G}_{\mathrm{NO}}$ (and $\mathbf{a} \neq \mathbf{0}$), we have that $\mathbf{a}^\top$ is not in the row span of $\mathbf{P}$. This means that for a random $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{d+1}$, $\mathbf{a}^\top \mathbf{s}$ is uniformly distributed over $\mathbb{Z}_q$ given $\mathbf{Ps}$. Smoothness follows readily.

6.2 KDM-security

We extend the d-LIN based hash proof system in [9, 31], which are the vectorial analogues of the preceding constructions, augmented with t functions following [11]. This in turn captures the DDH-based KDM-secure encryption in [9] and the DLIN-based scheme in [13]. Fix $\ell \geq (d+2)\log q$ and suppose we have t additional (efficiently computable) functions $f_1, \ldots, f_t : \{0,1\}^\ell \to \{0,1\}$, where $t \geq 0$. For instance, these functions may be low-degree polynomials of the bits of the input, as considered in [11].

Setup. Sample $\mathbf{P} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{d \times (\ell + t)}$. Output

$$\mathrm{PP} := (\mathbb{G}, q, g, g^{\mathbf{P}})$$

The subgroup indistinguishability problem is given by:

$$\mathcal{G}_{\mathrm{YES}} := \left\{ g^{\mathbf{r}^\top \mathbf{P}} : \mathbf{r} \in \mathbb{Z}_q^d \right\} \qquad \text{and} \qquad \mathcal{G} := \left\{ g^{\mathbf{a}^\top} : \mathbf{a} \in \mathbb{Z}_q^{\ell+t} \right\}$$

where the group operation is the natural one given by entry-wise product. The uniform distributions over $\mathcal{G}_{\mathrm{YES}}$ and $\mathcal{G}$ are computationally distinguishable under the d-LIN assumption as shown in [31, 9].

Hashing. The hashing key is given by a column vector $\mathbf{s} \leftarrow_{\mathrm{R}} \{0,1\}^\ell$. We then set $\hat{\mathbf{s}} \in \{0,1\}^{\ell+t}$ to be the concatenation of $\mathbf{s}$ and $f_1(\mathbf{s}), \ldots, f_t(\mathbf{s})$.

$$\mu(g^{\mathbf{P}}, \mathbf{s}) := g^{\mathbf{P}\hat{\mathbf{s}}} \in \mathbb{G}^{d \times 1}$$

Private and public evaluation are given by:

$$\Lambda_{\mathbf{s}}(g^{\mathbf{a}}) := g^{\mathbf{a}^\top \hat{\mathbf{s}}} \in \mathbb{G} \qquad \text{and} \qquad \mathsf{Pub}(g^{\mathbf{P}\hat{\mathbf{s}}}, \mathbf{C}, \mathbf{r}) := g^{\mathbf{r}^\top (\mathbf{P}\hat{\mathbf{s}})}$$

Clearly, $\Lambda_{\mathbf{s}}(\cdot)$ is a group homomorphism and the projective property simply follows from the fact that $g^{(\mathbf{r}^\top \mathbf{P})\hat{\mathbf{s}}} = g^{\mathbf{r}^\top (\mathbf{P}\hat{\mathbf{s}})}$.

Smoothness. For average-case smoothness, the left-over hash lemma implies that for $\ell > (d+2)\log q$, the following distributions:

$$\langle \mathbf{P}, \mathbf{P}\hat{\mathbf{s}}, \mathbf{a}, \mathbf{a}^\top \hat{\mathbf{s}} \rangle \quad \text{and} \quad \langle \mathbf{P}, \mathbf{P}\hat{\mathbf{s}}, \mathbf{a}, a' \rangle$$

are $1/q$-statistically close, where $\mathbf{s} \leftarrow_{\mathrm{R}} \{0,1\}^\ell, \mathbf{a} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^\ell, a' \leftarrow_{\mathrm{R}} \mathbb{Z}_q$. Note that $\hat{\mathbf{s}}$ has ℓ bits of min-entropy, so $\hat{\mathbf{s}}$ conditioned on $\mathbf{P}\hat{\mathbf{s}} \in \mathbb{Z}_q^{d \times 1}$ has roughly $\ell - d\log q \geq 2\log q$ bits of min-entropy.

Class $\mathcal{F}$. The message space $\mathcal{M} = \{0,1\}$ and $\phi(m) = g^m$.

- Observe that for all $\mathbf{a} \in \mathbb{Z}_q^\ell, c \in \mathbb{Z}_q$ (such that $\mathbf{a}^\top\mathbf{s} + c \in \{0,1\}$ for all $\mathbf{s} \in \{0,1\}^\ell$):
$$\Lambda_{\mathbf{s}}(g^{(\mathbf{a}\|\mathbf{0})^\top}) \cdot g^c = g^{(\mathbf{a}\|\mathbf{0})^\top\hat{\mathbf{s}}} \cdot g^c = \phi(\mathbf{a}^\top\mathbf{s} + c)$$
- Moreover, for all $i \in [t]$,
$$\Lambda_{\mathbf{s}}(g^{\mathbf{e}_{\ell+i}}) = g^{f_i(\mathbf{s})} = \phi(f_i(\mathbf{s}))$$
where $\mathbf{e}_{\ell+i} \in \{0,1\}^{\ell+t}$ is the unit vector with a 1 in the $(\ell+t)$'th index.

That is, the resulting scheme is $\mathcal{F}$-KDM secure for $\mathcal{F} = \{\mathbf{s} \mapsto \mathbf{a}^\top\mathbf{s} + c \mid \mathbf{a} \in \mathbb{Z}_q^\ell, c \in \mathbb{Z}_q\} \cup \{f_1,\ldots,f_t\}$, i.e. affine functions of the bits of the secret key (which includes flipping the i'th bit of the key $\mathbf{s} \mapsto 1 - s_i$) plus the functions $f_1,\ldots,f_t$.

7 Instantiations from QR and DCR

We will rely on the subgroup indistinguishability framework of Brakerski and Goldwasser [10] (also [16, Section 7.4.2]). We consider a family of finite commutative groups $\mathbb{G}$ that is generated by two elements g, h of co-prime order (thus $|\mathbb{G}| = \mathrm{ord}(g) \cdot \mathrm{ord}(h)$); we use $\mathbb{G}_0$ to denote $\langle g \rangle$. We will require the following additional properties:

- given the public description of $\mathbb{G}$, we may compute $\mathrm{ord}(h)$ and a good approximation a for $\mathrm{ord}(g)$ (so that the uniform distributions over $[a]$ and over $[\mathrm{ord}(g)]$ are statistcally close).
- computing discrete log with respect to h is easy.
- the uniform distributions over $\mathbb{G}_0$ and over $\mathbb{G}$ are computationally indistinguishable, given g, h.
- given some trapdoor, deciding membership in $\langle g \rangle$ is easy.

For our instantiations here, the output of $\Lambda_{\mathrm{HK}}(\cdot)$ lies in $\mathbb{G}$. We will work with a relaxed notion of smoothness here in this section, where instead of requiring that $\Lambda_{\mathrm{HK}}(\cdot)$ be random over $\mathbb{G}$, we only require that $\Lambda_{\mathrm{HK}}(\cdot) \bmod \mathbb{G}_0$ be random over $\langle h \rangle$. More formally, smoothness states that for all $C \in \mathcal{G}_{\mathrm{NO}}$: $\Lambda_{\mathrm{HK}}(C) \bmod \mathbb{G}_0$ is statistically close to uniform over the subgroup $\langle h \rangle$ even given $\mu(\mathrm{HK})$. Similarly, average-case smoothness states that the following distributions are statistically close:
$$(\mu(\mathrm{HK}), C, \Lambda_{\mathrm{HK}}(C) \bmod \mathbb{G}_0) \quad \text{and} \quad (\mu(\mathrm{HK}), C, h')$$
where $C \leftarrow_{\mathrm{R}} \mathcal{G}$ and $h' \leftarrow_{\mathrm{R}} \langle h \rangle$. The relaxed notion of smoothness is sufficient for all of our applications as long as we will embed the message into the subgroup $\langle h \rangle$.

Instantiation from QR. Fix a Blum integer $N = PQ$ for k-bit safe primes $P, Q \equiv 3 \pmod 4$ (such that $P = 2p+1$ and $Q = 2q+1$ for primes p, q). Let $\mathbb{J}_N$ denote the subgroup of $\mathbb{Z}_N^*$ with Jacobi symbol $+1$, and let $\mathbb{QR}_N$ denote the subgroup of quadratic residues. The QR assumption states that the uniform distributions over $\mathbb{QR}_N$ and $\mathbb{J}_N \setminus \mathbb{QR}_N$ are computationally indistinguishable. That is, we may take $\mathbb{G}$ and $\mathbb{G}_0$ to be $\mathbb{J}_N$ and $\mathbb{QR}_N$ respectively. Observe that $\mathbb{J}_N$ is isomorphic to $\mathbb{QR}_N \times (\pm 1)$ and that $|\mathbb{J}_N| = 2pq = 2|\mathbb{QR}_N|$. We can then sample g by squaring a random element in $\mathbb{Z}_N^*$ and fix h to be -1. Note that $|\mathbb{QR}_N| = pq = N/4 - O(\sqrt{N})$, which we may approximate by $N/4$.

Instantiation from DCR. (See [16, Section 8.2]). Again, fix a Blum integer $N = PQ$ for k-bit safe primes $P, Q \equiv 3 \pmod 4$ (such that $P = 2p+1$ and $Q = 2q+1$ for primes p, q). Let $\mathbb{J}_{N^2}$ denote the subgroup of $\mathbb{Z}_{N^2}^*$

with Jacobi symbol +1, so $|\mathbb{J}_{N^2}| = N\phi(N)/2 = 2Npq$. Consider the cyclic subgroup $\mathbb{G}_0$ of $\mathbb{J}_{N^2}$ consisting of all N'th powers of elements of $\mathbb{J}_{N^2}$. Then, $\mathbb{J}_{N^2} = \mathbb{G}_0 \times \langle 1+N \rangle$. Roughly speaking, the DCR assumption states that the uniform distributions over $\mathbb{G}_0$ and $\mathbb{J}_{N^2}$ are computationally indistinguishable. We can sample a random generator g of $\mathbb{G}_0$ as follows: pick $x \leftarrow_{\mathrm{R}} \mathbb{Z}^*_{N^2}$ and set $g := -x^N$. In addition, we can fix $h := 1+N$. Note that $|\mathbb{G}_0| = Npq = N^2/4 - O(\sqrt{N})$, which we may approximate by $N^2/4$.

7.1 Dual-mode encryption

For dual-mode encryption, we use the Cramer-Shoup QR/DCR-based hash proof system in [16].

Setup. Sample a random group $\mathbb{G}$ along with generators g and h.

$$\mathrm{PP} := (\mathbb{G}, g, h)$$

The subgroup indistinguishability problem is given by:

$$\mathcal{G}_{\text{YES}} := \left\{ g^r : r \in \mathbb{Z}_{\mathrm{ord}(g)} \right\} = \mathbb{G}_0 \qquad \text{and} \qquad \mathcal{G} := \left\{ h^d \cdot g^r : d \in \mathbb{Z}_{\mathrm{ord}(h)}, r \in \mathbb{Z}_{\mathrm{ord}(g)} \right\} = \mathbb{G}$$

where $\mathsf{SampR}(r) = g^r$. We also denote by SP the trapdoor that allows us to verify membership in $\mathcal{G}_{\text{YES}}$; for the instantiations from QR and DCR, this would be the factorization of N.

Hashing. The hashing key is given by $s \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(\mathbb{G})}$.

$$\mu(\mathrm{PP}, s) := g^s \in \mathbb{G}$$

Private and public evaluation are given by:

$$\Lambda_s(C) := C^s \in \mathbb{G} \qquad \text{and} \qquad \mathsf{Pub}(g^s, g^r, r) := (g^s)^r = g^{rs}$$

Clearly, $\Lambda_s(\cdot)$ is a group homomorphism. The projective property follows from the fact that $(g^r)^s = (g^s)^r$. For smoothness, first observe that by the Chinese Remainder Theorem, $s \bmod \mathrm{ord}(h)$ is random even given g^s. Hence, $\Lambda_s(h^d g^r) \bmod \mathbb{G}_0 = h^{ds}$ is random over $\langle h \rangle$ if $d \neq 0$.

7.2 KDM-security

The next construction is implicit in [10], and is the vectorial analogue of the preceding construction, augmented with t functions following [11]. Let $\ell > 3\log|\mathbb{G}|$. Suppose we have t additional (efficiently computable) functions $f_1, \ldots, f_t : \{0,1\}^\ell \to \mathbb{Z}_{\mathrm{ord}(h)}$, where $t \geq 0$.

Setup. Sample a random group $\mathbb{G}$ along with generators g and h. In addition, sample $\mathbf{p} \leftarrow_{\mathrm{R}} \mathbb{Z}^{\ell+t}_{\mathrm{ord}(g)}$. Output

$$\mathrm{PP} := (\mathbb{G}, g^{\mathbf{p}}, h)$$

The subgroup indistinguishability problem is given by:

$$\mathcal{G}_{\text{YES}} := \left\{ g^{r\mathbf{p}} : r \in \mathbb{Z}_{\mathrm{ord}(g)} \right\} \subseteq \mathbb{G}_0^{\ell+t} \qquad \text{and} \qquad \mathcal{G} := \left\{ h^{\mathbf{d}} \cdot g^{r\mathbf{p}} : \mathbf{d} \in \mathbb{Z}^{\ell}_{\mathrm{ord}(h)}, r \in \mathbb{Z}_{\mathrm{ord}(g)} \right\} \subseteq \mathbb{G}^{\ell+t}$$

where the group operation over $\mathbb{G}^{\ell+t}$ is the natural one given by coordinate-wise product. The uniform distributions over $\mathcal{G}_{\text{YES}}$ and $\mathcal{G}$ are computationally distinguishable under subgroup indistinguishability as shown in [10]. (The reduction is fairly straight-forward: it essentially takes the challenge (x, g, h) where either $x \leftarrow_{\mathrm{R}} \mathbb{G}_0$ or $x \leftarrow_{\mathrm{R}} \mathbb{G}$ and computes $(g^{\mathbf{p}'}, x^{\mathbf{p}'})$ where $\mathbf{p}' \leftarrow_{\mathrm{R}} \mathbb{Z}^{\ell+t}_{|\mathbb{G}|}$.)

Hashing. The hashing key is given by a column vector $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}^{\ell}_{\mathrm{ord}(h)}$. We then set $\hat{\mathbf{s}} \in \mathbb{Z}^{\ell+t}_{\mathrm{ord}(h)}$ to be the concatenation of $\mathbf{s}$ and $f_1(\mathbf{s}), \ldots, f_t(\mathbf{s})$.

$$\mu(g^{\mathbf{p}}, \mathbf{s}) := g^{\mathbf{p}^{\top}\hat{\mathbf{s}}} \in \mathbb{G}$$

Private and public evaluation are given by:

$$\Lambda_{\mathbf{s}}(\mathbf{c}) := \mathbf{c}^{\hat{\mathbf{s}}} \in \mathbb{G} \qquad \text{and} \qquad \mathsf{Pub}(g^{\mathbf{p}^{\top}\hat{\mathbf{s}}}, \mathbf{c}, r) := (g^{\mathbf{p}^{\top}\hat{\mathbf{s}}})^r$$

where $\mathbf{c}^{\hat{\mathbf{s}}} := \sum_{i=1}^{\ell+t} \mathbf{c}_i^{\hat{\mathbf{s}}_i}$. Clearly, $\Lambda_{\mathbf{s}}(\cdot)$ is a group homomorphism. The projective property simply follows from the fact that $g^{(r\mathbf{p})^{\top}\hat{\mathbf{s}}} = g^{r\mathbf{p}^{\top}\hat{\mathbf{s}}} = (g^{\mathbf{p}^{\top}\hat{\mathbf{s}}})^r$.

Smoothness. To establish average-case smoothness, first observe that:

$$\Lambda_{\hat{\mathbf{s}}}(h^{\mathbf{d}} \cdot g^{r\mathbf{p}}) \bmod \mathbb{G}_0 = h^{\mathbf{d}^{\top}\hat{\mathbf{s}}}$$

The left-over hash lemma tells us that $\mathbf{d}^{\top}\hat{\mathbf{s}}$ is statistically close to uniform over $\mathbb{Z}_{\mathrm{ord}(h)}$. More precisely, for $\ell > 3\log|\mathbb{G}|$, the following distributions:

$$\langle \mathbf{p}, \mathbf{p}^{\top}\hat{\mathbf{s}} \bmod |\mathbb{G}_0|, \mathbf{d}, \mathbf{d}^{\top}\hat{\mathbf{s}} \bmod \mathrm{ord}(h) \rangle \quad \text{and} \quad \langle \mathbf{p}, \mathbf{p}^{\top}\hat{\mathbf{s}} \bmod |\mathbb{G}_0|, \mathbf{d}, d' \rangle$$

are statistically close, where $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}^{\ell}_{\mathrm{ord}(h)}, \mathbf{d} \leftarrow_{\mathrm{R}} \mathbb{Z}^{\ell}_{\mathrm{ord}(h)}, d' \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(h)}$. Average-case smoothness follows readily, since $g^{\mathbf{p}^{\top}\hat{\mathbf{s}}}$ is completely determined by $\mathbf{p}^{\top}\hat{\mathbf{s}} \bmod |\mathbb{G}_0|$.

Class $\mathcal{F}$. The message space $\mathcal{M} = \mathbb{Z}_{\mathrm{ord}(h)}$ and $\phi(m) = h^m$.

- Observe that for all $\mathbf{a} \in \mathbb{Z}^{\ell}, c \in \mathbb{Z}$ (such that $\mathbf{a}^{\top}\mathbf{s} + c \in \mathbb{Z}_{\mathrm{ord}(h)}$ for all $\mathbf{s} \in \mathbb{Z}^{\ell}_{\mathrm{ord}(h)}$):

$$\Lambda_{\mathbf{s}}(h^{\mathbf{a}\|\mathbf{0}}) \cdot h^c = h^{\mathbf{a}^{\top}\mathbf{s}+c} = \phi(\mathbf{a}^{\top}\mathbf{s} + c)$$

- Moreover, for all $i \in [t]$,

$$\Lambda_{\mathbf{s}}(h^{\mathbf{e}_{\ell+i}}) = h^{f_i(\mathbf{s})} = \phi(f_i(\mathbf{s}))$$

  where $\mathbf{e}_{\ell+i} \in \{0,1\}^{\ell+t}$ is the unit vector with a 1 in the $(\ell+t)$'th index.

That is, the resulting scheme is $\mathcal{F}$-KDM secure for $\mathcal{F} = \{\mathbf{s} \mapsto \mathbf{a}^{\top}\mathbf{s} + c \mid \mathbf{a} \in \mathbb{Z}^{\ell}, c \in \mathbb{Z}\} \cup \{f_1, \ldots, f_t\}$, i.e. affine functions of the bits of the secret key, plus the functions $f_1, \ldots, f_t$.

Acknowledgments. I would like to thank David Cash, Kai-Min Chung and Dennis Hofheinz for helpful discussions.

References

[1] P. Adão, G. Bana, J. Herzog, and A. Scedrov. Soundness of formal encryption in the presence of key-cycles. In *ESORICS*, pages 374–396, 2005.

[2] J. Alperin-Sheriff and C. Peikert. Circular and KDM security for identity-based encryption. In *Public Key Cryptography*, pages 334–352, 2012.

[3] B. Applebaum. Key-dependent message security: Generic amplification and completeness. In *EUROCRYPT*, pages 527–546, 2011. Cryptology ePrint Archive, Report 2010/513.

[4] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618, 2009.

[5] B. Barak, I. Haitner, D. Hofheinz, and Y. Ishai. Bounded key-dependent message security. In *EUROCRYPT*, pages 423–444, 2010.

[6] F. Benhamouda, O. Blazy, C. Chevalier, D. Pointcheval, and D. Vergnaud. New techniques for SPHFs and efficient one-round PAKE protocols. In *CRYPTO (1)*, pages 449–475, 2013.

[7] J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Selected Areas in Cryptography*, pages 62–75, 2002.

[8] F. Böhl, G. T. Davies, and D. Hofheinz. Encryption schemes secure under related-key and key-dependent message attacks. In *Public Key Cryptography*, pages 483–500, 2014.

[9] D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from Decision Diffie-Hellman. In *CRYPTO*, pages 108–125, 2008.

[10] Z. Brakerski and S. Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In *CRYPTO*, pages 1–20, 2010. Also, Cryptology ePrint Archive, Report 2010/522.

[11] Z. Brakerski, S. Goldwasser, and Y. T. Kalai. Black-box circular-secure encryption beyond affine functions. In *TCC*, pages 201–218, 2011.

[12] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT*, pages 93–118, 2001.

[13] J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *EUROCRYPT*, pages 351–368, 2009.

[14] C. Cocks. An identity based encryption scheme based on quadratic residues. In *IMA Int. Conf.*, pages 360–363, 2001.

[15] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.

[16] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64, 2002. Also, Cryptology ePrint Archive, Report 2001/085.

[17] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. In *CRYPTO*, pages 205–210, 1982.

[18] R. Gennaro and Y. Lindell. A framework for password-based authenticated key exchange. *ACM Trans. Inf. Syst. Secur.*, 9(2): 181–234, 2006.

[19] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. In *FOCS*, pages 325–335, 2000.

[20] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*, pages 218–229, 1987.

[21] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[22] I. Haitner and T. Holenstein. On the (im)possibility of key dependent encryption. In *TCC*, pages 202–219, 2009.

[23] S. Halevi and Y. T. Kalai. Smooth projective hashing and two-message oblivious transfer. *J. Cryptology*, 25(1):158–193, 2012.

[24] B. Hemenway, B. Libert, R. Ostrovsky, and D. Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In *ASIACRYPT*, pages 70–88, 2011. also Cryptology ePrint Archive, Report 2009/088.

[25] D. Hofheinz. Circular chosen-ciphertext security with compact ciphertexts. In *EUROCRYPT*, pages 520–536, 2013.

[26] D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. In *CRYPTO*, pages 553–571, 2007.

[27] J. Katz and V. Vaikuntanathan. Smooth projective hashing and password-based authenticated key exchange from lattices. In *ASIACRYPT*, pages 636–652, 2009.

[28] J. Kilian. Founding cryptography on oblivious transfer. In *STOC*, pages 20–31, 1988.

[29] T. Malkin, I. Teranishi, and M. Yung. Efficient circuit-size independent public key encryption with KDM security. In *EUROCRYPT*, pages 507–526, 2011.

[30] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *SODA*, pages 448–457, 2001.

[31] M. Naor and G. Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages 18–35, 2009.

[32] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO*, pages 554–571, 2008. Also, Cryptology ePrint Archive, Report 2007/118.

[33] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[34] H. Shacham. A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074, 2007.

[35] A. C.-C. Yao. How to generate and exchange secrets. In *FOCS*, pages 162–167, 1986.