On Generic Constructions of Circularly-Secure, Leakage-Resilient Public-Key Encryption Schemes

Mohammad Hajiabadi, Bruce M. Kapron, Venkatesh Srinivasan

Department of Computer Science, University of Victoria
Victoria, BC, CANADA
{mhaji,bmkapron,venkat}@cs.uvic.ca

Abstract. We propose generic constructions of public-key encryption schemes, satisfying *key-dependent message (KDM) security for projections* and different forms of *key-leakage resilience*, from CPA-secure private key encryption schemes with two main abstract properties: (1) additive homomorphism with respect to both messages and randomness, and (2) *reproducibility*, providing a means for reusing encryption randomness across independent secret keys. More precisely, our construction transforms a private-key scheme with the stated properties (and one more mild condition) into a public-key one, providing:

- n-KDM-projection security, an extension of circular security, where the adversary may also ask for encryptions of negated secret key bits;
- a $(1-o(1))$ resilience rate in the bounded-memory leakage model of Akavia et al. (TCC 2009); and
- *Auxiliary-input security* against subexponentially-hard functions.

We introduce *homomorphic weak pseudorandom functions*, a homomorphic version of the weak PRFs proposed by Naor and Reingold (FOCS '95) and use them to realize our base encryption scheme. We obtain homomorphic weak PRFs under assumptions including subgroup indistinguishability (implied, in particular, by QR and DCR) and *homomorphic hash-proof systems* (*HHPS*). As corollaries of our results, we obtain (1) a projection-secure encryption scheme (as well as a scheme with a $(1-o(1))$ resilience rate) based solely on the HHPS assumption, and (2) a unifying approach explaining the results of Boneh et al (CRYPTO '08) and Brakerski and Goldwasser (CRYPTO '10). Finally, by observing that Applebaum's KDM amplification method (EUROCRYPT '11) preserves both types of leakage resilience, we obtain schemes providing at the same time high leakage resilience and KDM security against any fixed polynomial-sized circuit family.

1 Introduction

A central goal in cryptography is to build a variety of cryptographic primitives with a high degree of versatility from assumptions that are as general as possible. Encryption in particular has been defined, starting with the seminal paper of Goldwasser and Micali [23], with respect to successively stronger models of security. However, standard notions of encryption security (i.e., CPA and different forms of CCA security [23, 35, 38, 17]) fall short in certain applications, in particular, where the adversary may obtain some side information about the internal secret parameters (e.g., the secret key) of the scheme. This leakage of side information may occur due to some unforeseen attacks on the scheme (*side-channel attacks*), or more fundamentally, when encryption is used as a primitive in a complex protocol which may inherently expose inside information. These observations have led to the definitions and realizations of stronger notions of encryption security, such as security against different forms of leakage [32, 19, 1, 34, 15, 14, 2, 24, 9], and key-dependent message (KDM) security [7, 27, 8, 5, 4, 9, 31, 3, 10]. Our goal is to construct schemes realizing these security properties from general assumptions. Our results concern a basic model of leakage, known as the *bounded-leakage model* [1], a basic model of KDM security, known as *projection* security (which is slightly stronger than *circular* security) and a model of *auxiliary-input security* [15, 14]. We first provide some background and then describe our results.

For all definitions below (unless otherwise stated) we assume we are encrypting the secret key (or functions thereof) bit-by-bit, i.e., the scheme is either bit encryption, or there is a mapping from bits to two fixed plaintext messages.

KDM security. KDM security is defined wrt a function family F: informally, an encryption scheme (G, E, Dec) is F-KDM$^{(1)}$ secure if no adversary can distinguish between two oracles, where the first one,

on input $f \in F$, returns $E_{pk}(f(sk))$ (for a random (pk, sk) chosen at the beginning), and the second one, regardless of the input, returns an encryption of a fixed message. A basic form of KDM$^{(1)}$ security is 1-circular security, allowing the adversary to obtain encryptions of any bit of the secret key. Another basic notion is projection security, which also allows the adversary to obtain encryptions of negations of secret key bits. KDM$^{(1)}$ security generalizes naturally to the case of multiple pairs of keys, giving rise to the notion of F-KDM$^{(n)}$-security, where a chosen function f is applied to all of the secret keys to yield $f(sk_1, \ldots, sk_n)$. For example, multiple-key circular security allows the adversary to see encryptions of any bit of any secret key under (possibly) any other public key.

KDM security was originally defined by Black et al. [7], who built a *fully-KDM*-secure scheme (i.e., wrt all function families) in the random oracle model. In [8] Boneh et al. gave the first construction in the standard model, based on the DDH assumption, of a public-key scheme proved KDM$^{(n)}$ secure wrt *affine functions*. This positive result led to a series of subsequent works, focusing on building affine-KDM$^{(n)}$ security under alternate specific assumptions (i.e., LPN/LWE [4], and QR/DCR and more generally *subgroup indistinguishability (SG)* assumptions [9]), and on developing *KDM-amplification* methods for transforming schemes with basic forms of KDM security into schemes with more sophisticated forms of KDM security [5, 10, 3]. These amplification methods in turn employ techniques such as garbled circuits [5], *randomized encoding of functions* [3] and *entropic-KDM* security [10] to enable KDM transformations. Most relevant to our work are the results of Applebaum [3], showing that projection security is sufficient to obtain KDM security wrt any fixed circuit family whose size is poly-bounded.Thus, a fundamental question regarding KDM security is to study general assumptions sufficient for realizing projection security, which is one of the main goals in our paper.

It turns out that realizing even 1-circular security for bit encryption is considerably more difficult than the case where the secret-key space is a subset of the plaintext space (so one can encrypt the whole key at once). In the latter case, through simple modifications to the encryption algorithm, one can make any CPA-secure scheme 1-circularly secure. Currently, the only constructions that provide bitwise 1-circular security are those of [8, 4, 10], which are based on specific assumptions. Also, it was shown in [41] that the implication "whether any CPA-secure bit encryption scheme is also 1-circularly secure" is not provable using reductions that use both the adversary and the scheme in a blackbox way.[1] Moreover, under well-believed assumptions, there exist CPA-secure bit-encryption schemes that are not 1-circularly secure [41, 30].

Leakage resilience. Akavia et al. [1] introduce the notion of encryption security against bounded memory leakage, wherein an adversary (after seeing the public key) may obtain arbitrary information about the secret key, of the form $f(sk)$ for adaptively chosen f, as long as the total number of bits leaked does not exceed an a priori fixed quantity, ℓ. (We refer to the fraction $\ell/|sk|$ as the leakage rate.) They showed that Regev's scheme [39] and the identity based encryption scheme of [20], both under the LWE assumption, are resilient to leakage of rate $O(1/polylog(|sk|))$. Naor and Segev [34] showed how to obtain encryption schemes resilient to high leakage lengths (but with low resilience rates) from any hash-proof system [13] and how to obtain schemes with $(1 - o(1))$-resilience rates from d-linear assumptions; moreover, they showed that the circularly-secure scheme of [8] provides a $(1 - o(1))$ resilience rate. Brakerski and Goldwasser [9], under the subgroup indistinguishability assumption, implied in turn by the QR and DCR assumptions, showed how to obtain encryption schemes that are affine-KDM secure, with a $(1 - o(1))$ resilience rate.

Auxiliary-input security. In the auxiliary-input model [15, 14] the adversary is given some side information of the form $h(pk, sk)$, and the goal is to guarantee security as long as recovering sk from $h(pk, sk)$ is sufficiently, computationally hard. For public-key encryption Dodis et al. [14] build schemes based on LWE and DDH (where their DDH-based scheme is a variant of [8]) secure against *subexponentially-hard-to-invert* functions. (That is for any $\epsilon > 0$ they build a scheme, whose parameters may depend on ϵ, that is secure against 2^{-k^ϵ}-hard functions.) Brakerski and Goldwasser [9] present schemes with the same level of auxiliary-input security under the subgroup indistinguishability assumption.

1.1 Our results (constructions)

As pointed our earlier, the only constructions of circularly-secure bit encryption (even 1-circular security) are based on specific assumptions [8, 4, 9]. Moreover, the schemes of [8, 9], referred to as BHHO and BG

[1] Note that this is different from asking whether CPA-secure bit encryption implies the existence of circularly-secure bit encryption.

henceforth, besides KDM security, also provide security against different forms of leakage (as shown in [14, 34, 9]). Therefore, a natural question is whether there exist more general constructions that encompass all these specific constructions.

We will try to answer these questions by building leakage-resilient, projection-secure encryption schemes from CPA-secure private-key schemes with some special properties (sketched below). Then we will use this encryption abstraction as a stepping stone toward obtaining our results under a higher-level primitive.

The first property is a generalized version of additive homomorphism, where homomorphism is required to hold also wrt randomness, and the second property is what Bellare et al. [6] call *reproducibility*, requiring that given a message m_2, secret key sk_2 and ciphertext $c = E_{sk_1}(m_1; r)$, where sk_1, m_1 and r are unknown, one can efficiently obtain $E_{sk_2}(m_2; r)$, i.e., there is a way to efficiently transfer the randomness from one encryption to another, provided the secret key for the second encryption is known[2]. Note that if an encryption algorithm reveals its randomness in the clear, then reproducibility is trivially satisfied, e.g., the standard way of building CPA-secure private-key encryption from a pseudorandom function family F, by encrypting as $E_{sk}(m) = (r, F_{sk}(r) \oplus m)$, provides reproducibility. In fact, we will later use this idea to obtain our encryption primitive, based on the existence of *homomorphic weak pseudorandom functions*. Finally, the last property is what we call *flexibility*, stating (for bit-encryption) that from any encryption $E_{sk}(b; r)$, for unknown sk, b, r, one can obtain $E_{sk}(1; 0)$, i.e., the encryption of bit 1 under key sk based on the identity randomness element.[3]

Let Hom and Rep be the functions associated with the first two properties above. We introduce a construction C (formalized in Construction 1) that transforms a private-key scheme with the stated properties into a public-key one and show the following result.

Theorem 1 (informal). Assume that $\mathcal{E} = (G, E, Dec, Hom, Rep)$ is a CPA-secure private-key, bit-encryption scheme that is additively homomorphic, reproducible, and flexible.Then the constructed scheme $\mathcal{E}' = C(\mathcal{E})$ is a public-key bit-encryption scheme that satisfies the following properties.

- For any integer n, by appropriately choosing the system parameters, $\mathcal{E}'$ is n-projection secure. (Formalized in Theorems 1, 2)
- By appropriately choosing the system parameters, $\mathcal{E}'$ provides a $(1 - o(1))$-leakage resilience rate. (Formalized in Theorem 3)
- $\mathcal{E}'$ provides auxiliary-input leakage resilience against subexponentially-hard functions. (Formalized in Theorem 4 and Remark 2)

We mention that a similar construction can be given assuming the base scheme is public key; under this construction the last property (i.e., flexibility) is no longer needed—of course assuming that the identity randomness element is publicly known. However, the private-key-based construction has the advantage that it allows us to realize the base assumptions under homomorphic PRFs.

In Appendix A.4 we show a generalization of the construction above to the case of non-bit-encryption, by maintaining the plaintext space of the base scheme (Construction 2, Theorem 11 and Corollary 2). Also, as explained next, under homomorphic weak PRFs, all the conditions spelled out above are satisfied.

1.2 Realization from homomorphic weak PRFs

Pseudorandom function families (PRFs) provide a convenient way of realizing reproducible CPA-secure private-key encryption via the standard PRF-based encryption construction. Towards providing homomorphism for a PRF-based scheme, we call a function family *homomorphic* if both the domain and range of the underlying functions form groups, and each function acts as a homomorphism. A standard PRF cannot, however, be homomorphic since with high probability a truly random function will not be homomorphic and an adversary with the power to (even nonadaptively) query a function oracle may easily exploit this fact. To prevent this type of attack, we work with *weak PRFs*, defined by Naor and Reingold [33], which allow an adversary to see values of the function only on a sequence of random inputs. Formally, f_k is *weakly pseudorandom* if no adversary can distinguish between $(d_1, f_k(d_1)), \ldots, (d_p, f_k(d_p))$ and $(d_1, r_1), \ldots, (d_p, r_p)$, where all d_i's and r_i's are chosen independently at random. As we see next,

[2] Both these conditions were used implicitly by Peikert and Waters as the main building blocks for their construction of lossy-trapdoor functions [37].

[3] The actual assumption we need is substantially weaker. However, we leave it this way for the sake of readability. In fact, under all concrete schemes we present, $E_{sk}(m; 0)$ depends only on m and is independent of sk.

not only is the notion of homomorphic weak PRFs meaningful, it is naturally realizable under specific assumptions. We also note that the standard construction of private-key encryption from a PRF, when applied to homomorphic weak PRFs, results in a scheme that satisfies the properties we need from our base encryption primitive (Lemma 3).

For a DDH-hard group $\mathbb{G}$ with $o = |\mathbb{G}|$, define $F = \{f_k \colon \mathbb{G} \to \mathbb{G}\}_{k \in \mathbb{Z}_o}$ as $f_k(g) = g^k$. The fact that f_k is homomorphic is clear, and the fact that F is weakly pseudorandom follows from standard facts about DDH. Interestingly, by plugging this PRF into our general construction, we obtain a scheme which is a close variant of BHHO. We also give an instantiation of weak homomorphic PRFs under *homomorphic hash-proof systems (HHPS)* [13]: Here the PRF is simply the family of hash functions on *valid* points. Finally, for instantiations base on the subgroup indistinguishability (SG) assumption [9] (which is in turn implied by QR and DCR), we weaken weak pseudorandomness even further (we call this weak$_1$ pseudorandomness.) A corollary of our results is the following.

Corollary. Under the HHPS assumption and for any integer n, there exists a public-key encryption scheme that provides, at the same time, n-projection security and a $(1 - o(1))$-leakage resilience rate (Formalized in Corollary 1).

Naor and Segev [34] show how to construct schemes with high tolerated leakage lengths (but with low resilience rates) from any hash-proof system, and also how to obtain schemes with high resilience rates from *k-linear assumptions*. Our results can be thought of as complementing those of [34], by saying that if we add homomorphism to a HPS, we obtain schemes with high resilience rates. Hazay et al. [26] show how to obtain schemes withstanding high leakage lengths from any CPA-secure public-key encryption. Their construction, however, produces a scheme with low resilience rates, and does not imply our leakage resilience result based on HHPS.

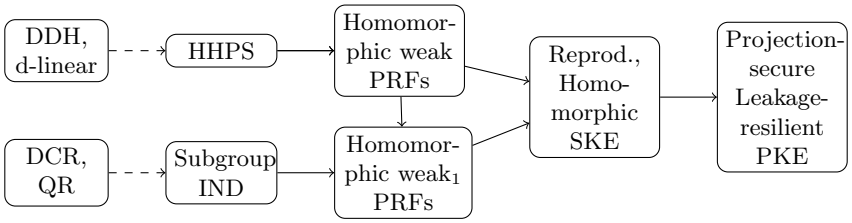

Fig. 1. Summary of results (dashed arrows indicate known implications)

1.3 KDM amplification and leakage resilience

We prove that Applebaum's KDM amplification method [3], which, informally speaking, allows one to obtain KDM-security for any fixed family of bounded circuits from projection security, also preserves both types of leakage resilience (Theorem 9). We were not, however, able to show this for the KDM amplification methods of [5, 10]. Applebaum's transformation has the key property that it only modifies the encryption and decryption algorithms of the base scheme, by applying *randomized encoding and decoding*, which are fixed mappings constructed based on the target function family, inside the encryption and decryption algorithms. This property facilitates reducing leakage resilience and auxiliary input security of the constructed scheme to the same requirements (i.e., with the same parameters) on the base scheme. As a corollary, for any fixed bounded function family F and any integer n, based on weak homomorphic PRFs, we obtain schemes that at the same time provide (1) F-KDM$^{(n)}$ security, (2) a $(1 - o(1))$-leakage resilience rate, and (3) auxiliary-input security against subexponentially-hard functions (Corollary 1).

1.4 Discussion, construction technique and open questions

Our results (see Figure 1) may be viewed from three perspectives. First of all, we show that by adding a few properties to CPA-secure private-key encryption, we are able to obtain public-key schemes with very strong KDM and leakage-resilience properties. Secondly, we introduce homomorphic weak PRFs

and demonstrate their utility for constructing such encryption schemes. Finally, by instantiating these PRFs under a number of standard specific assumptions, we unify and elucidate existing constructions.

While our results enable us to explain those of [8, 9, 34], regarding KDM security and leakage resilience of the BHHO and BG schemes, they suffer from the same limitations as those of [9], in that, in order to achieve KDM$^{(n)}$ security, we should choose the parameters of our constructed scheme based on n. Boneh et al. [8] get around this dependency by using the *random self-reducibility* of DDH and strong key-homomorphism properties of DDH-based schemes. Similar dependencies for (even specific) non-DDH-based assumptions occur in other settings as well, e.g., [11]. We leave it as an open problem to resolve this dependency. We should also mention that the BHHO and BG schemes were proved affine-KDM secure; under the current assumptions, we were not able to extend our results to the affine-KDM setting. However, by adding a *key homomorphism* property, we can prove affine-KDM security and show that our instantiations based on DDH and SG satisfy this property. We defer a detailed discussion on this point to a later version of the paper. Finally, we note that just the fact that we can build a CPA-secure (as opposed to KDM secure) public-key scheme from our private-key assumptions is not unheard of since even weaker forms of homomorphism are known to be sufficient to bridge this gap [40].

Construction and proof techniques. We now give a sketch of the construction, C, and proof techniques. Fix $\mathcal{E} = (G, E, D, Rep, Hom)$ to be a private-key bit-encryption scheme that provides reproducibility and the generalized homomorphism condition; the latter, using additive notation, states the following condition that $Hom(E_{sk}(b_1; r_1), E_{sk}(b_2; r_2)) = E_{sk}(b_1 + b_2; r_1 + r_2)$.
Under $\mathcal{E}' = C(\mathcal{E}) = (G', E', D')$, the secret key is a random string $\mathbf{s} \leftarrow \{0,1\}^l$ (for some poly l) and the public key is a tuple of ciphertexts

$$\mathbf{pk} = (E_{sk}(0; r_1), \ldots, E_{sk}(0; r_l), E_{sk}(0; \mathbf{s} \cdot \mathbf{r})),$$

where all $sk, r_1, \ldots, r_l$ are generated randomly under $\mathcal{E}$ and and $(\cdot)$ denotes the inner product of $\mathbf{s}$ and $\mathbf{r} = (r_1, \ldots, r_l)$. In words, $\mathbf{pk}$ consists of $l + 1$ $\mathcal{E}$-encryptions of zero, where the first l encryptions are produced independently, while the randomness value used for the last encryption is a "subset-sum" of the previous ones based on $\mathbf{s}$. To encrypt a bit b we sample $sk' \leftarrow G(1^\lambda)$ and output $(E_{sk'}(0; r_1), \ldots, E_{sk'}(0; r_l), E_{sk'}(b; \mathbf{s} \cdot \mathbf{r}))$, which can be computed from $\mathbf{pk}$ by applying Rep componentwise. To decrypt $(c_1, \ldots, c_l, c_{l+1})$ under $\mathbf{s}$, we return 0 iff $c_{l+1} = Hom_{\mathbf{s}}(c_1, \ldots, c_l)$, where $Hom_{\mathbf{s}}(c_1, \ldots, c_l)$ "sums" those ciphertexts c_i where $\mathbf{sk}_i = 1$. The correctness of decryption follows.

Some notes are in order. Firstly, under G', the secret key of the old scheme, sk, is only used to compute the encryptions (put in $\mathbf{pk}$) and is discarded after this; roughly, the fact that $\mathbf{s}$ is totally independent of sk is one main reason behind the circular security of $\mathcal{E}'$. Secondly, E' has (perhaps) the unusual property that it calls G, with this being the only underlying randomness part.

As a warm-up we first discuss CPA security. Consider a malformed public key $\mathbf{pk}_{mal}$ formed by having r_{l+1} also be independent of all others (instead of being $\mathbf{s} \cdot \mathbf{r}$). CPA security under $\mathbf{pk}_{mal}$ reduces to showing

$$(E_{sk}(0; r_1), \ldots, E_{sk}(0; r_l), E_{sk}(0; r_{l+1})), (E_{sk'}(0; r_1), \ldots, E_{sk'}(0; r_l), E_{sk'}(b; r_{l+1}))$$

results in computationally indistinguishable distributions for $b \in \{0,1\}$, which in turn follows by resorting to CPA security and reproducibility of $\mathcal{E}$. To complete the CPA-security proof, it would suffice to argue a malformed public-key is indistinguishable from a valid one, which follows information-theoretically (from the *leftover hash lemma*) if l is large enough. Below we extend the arguments given here to argue about KDM and leakage-resilience security of the scheme.

KDM^n-projection security. A main idea used in the proof of 1-circular security (for simplicity) is that if one possesses $\mathbf{s}$, then encrypting a bit b may be equivalently computed as $\mathbf{c} = (c_1, \ldots, c_l, Hom(c_{i_1}, \ldots, c_{i_w}, c'))$, where $sk' \leftarrow G(1^\lambda)$, $c_j \leftarrow E_{sk'}(0)$ for $1 \le j \le l$, $(i_1, \ldots, i_w)$ are the indices of nonzero bits of $\mathbf{s}$ and $c' = E_{sk'}(b; 0)$ (i.e., c' is the encryption of b based on the identity randomness). Now we consider an intermediate hybrid, W_1, in which to encrypt the hth bit of $\mathbf{s}$, we return $(c_1, \ldots, c_l, Hom(c_{i_1}, \ldots, c_{i_w}, c'))$, where now c_h is an encryption of 1, but every other c_j is an encryption of 0 (and c' is an encryption of $\mathbf{s}_h$ under the identity randomness). We will show that W_1 provides a view computationally indistinguishable from the real view, W_0; the main idea is that any distinguisher between W_0 and W_1 can be reduced to an adversary $\mathcal{A}$ that wins in a special vector-encryption game (performed under $\mathcal{E}$), in which $\mathcal{A}$ may adaptively issue fixed-length vectors of bits (of a certain form), and in response to each vector query $\mathbf{v}$, either $\mathbf{v}$ or the all-zero vector (depending on the challenge bit) is

component-wise encrypted under a fresh secret key, but by reusing randomness across each fixed component of vectors (that is the ith component of each vector is always encrypted under a fixed random r_i). In Lemma 1 we show any $\mathcal{A}$ has a negligible advantage under this game, and use this to prove the indistinguishability of W_0 and W_1. (It turns out this last step also requires us to use flexibility of $\mathcal{E}$, to be able to compute $E_{sk'}(1;0)$ obliviously to sk'.) Having proved the indistinguishability of W_0 and W_1 we notice that under W_1 the reply to "encrypt the hth bit of $\mathbf{s}$" is indeed formed as $(E_{sk'}(0;r_1),\ldots,E_{sk'}(0;r_{h-1}),E_{sk'}(1;r_h),E_{sk'}(0;r_{h+1}),\ldots,E_{sk'}(0;r_l),E_{sk}(0;\mathbf{s}\cdot\mathbf{r}))$, and in particular independent of $\mathbf{s}$ beyond $\mathbf{s}\cdot\mathbf{r}$, which makes the rest of the proof follow smoothly using ideas described for the CPA case.

The techniques described above are what might be called *simulated KDM encryptions*, originally introduced in [8], used also in subsequent works [4, 9], showing how to simulate KDM responses under public information. The main challenge in our setting is how to enable such properties under our general assumptions.

Leakage resilience. For simplicity, we first outline the idea of the proof for the case of nonadaptive leakage resilience (that is, the function f is queried before the public key being published). To argue about nonadaptive leakage resilience, one has to show $D_0 \equiv^c D_1$, where

$$D_b = (\underbrace{E_{sk}(0;r_1),\ldots,E_{sk}(0;r_l),E_{sk}(0;\mathbf{s}\cdot\mathbf{r})}_{\mathbf{pk}},\underbrace{E_{sk'}(0;r_1),\ldots,E_{sk'}(0;r_l),E_{sk'}(b;\mathbf{s}\cdot\mathbf{r})}_{\mathbf{c}_b},f(\mathbf{s})).$$

Now since f is chosen independently of $\mathbf{pk}$, it is also independent of $\mathbf{r}$, which allows us to apply the average-case version of the leftover hash lemma [16] (considering the inner product acts as universal hashing) to replace $\mathbf{s}\cdot\mathbf{r}$ with a totally random r_{l+1}; the rest of the proof follows considering $\mathcal{E}$ allows secure reuse of randomness. For the adaptive case, to handle the issue f depends on $\mathbf{pk}$ (and so we cannot apply random extraction directly), we use similar techniques to those used by [34]: we consider a hybrid D'_b, which is similar to D_b, but in which the first l bits encrypted under sk' are independently random bits $b_1,\ldots,b_l$ (as opposed to zeros) and that the last bit is $\mathbf{s}\cdot(b_1,\ldots,b_l)+b$. By proving $D'_b \equiv^c D_b$, for both $b\in\{0,1\}$, (essentially using reproducibility and semantic security of $\mathcal{E}$) we can now apply the generalized leftover hash lemma by taking $(b_1,\ldots,b_l)$ as the seed, $\mathbf{s}$ as the source and considering that b_i's are chosen independently of f and $\mathbf{r}$; this would allow us to replace $\mathbf{s}\cdot(b_1,\ldots,b_l)$ with a uniformly random bit, proving D'_0 is statistically close to D'_1. The leakage resilience proof follows. The proof for the auxiliary-input case essentially follows the same line of arguments as that given above, except for replacing randomness extraction with pseudorandomness extraction [22]. We refer the reader directly to the full proof.

Final remarks. Instantiating the above construction using homomorphic weak PRFs provides an improvement in efficiency, matching the same level of efficiency as those of [8, 9] if the base PRF (in turn) is instantiated under the corresponding assumption. Technically, in this case, it would suffice to define the public key to be $(d_1,\ldots,d_l,\mathbf{s}\cdot(d_1,\ldots,d_l))$, i.e., instead of putting the whole ciphertext in each component, we only give the underlying randomness, which would have been given out by the ciphertext itself in the clear. Also, to encrypt m under $\mathbf{pk}=(d_1,\ldots,d_l,d_{l+1})$, we simply output $(F_{sk}(d_1),\ldots,F_{sk}(d_l),F_{sk}(d_{l+1})+m)$, where sk is a fresh PRF key.

Other related work. Choi and Wee [12] show how to construct *lossy trapdoor functions* from homomorphic reproducible encryption by abstracting the matrix-based construction of Peikert and Waters [37]. This shows one more application of homomorphic weak PRFs as a general primitive.[4] We mention, however, that the main difference between our constructions and those of [37, 12] is that in [37, 12] the trapdoor key of the constructed schemes consists of secret keys produced under the base scheme, while in our setting, the main challenge (and novelty) is to come up with a construction whose encryption function still somehow calls that of the base scheme (in order to inherit its security), but in such a way that the secret keys of the base scheme are not included in the constructed secret key.

[4] The construction of [12] requires a public-key reproducible homomorphic scheme to begin with. However, it is easy to see the same construction works wrt a private-key scheme.

2 Definitions

2.1 Standard Notation and Definitions

For a finite set S we use $x \leftarrow S$ to denote sampling x uniformly at random from S and denote by U_S or $U(S)$ the uniform distribution on S. If D is a distribution then $x \leftarrow D$ denotes choosing x according to D. We denote the length of $x \in \{0,1\}^*$ by $|x|$ and the ith bit of x, for $1 \leq i \leq |x|$, by x_i. We use $A(a_1, a_2, \ldots; r)$ to denote the deterministic output of randomized function A on inputs $a_1, a_2, \ldots$ and randomness r, and use $x \leftarrow A(a_1, a_2, \ldots)$ to denote the distribution formed by first choosing r uniformly at random and then outputting $A(a_1, a_2, \ldots; r)$. We use the term PPT in this paper in the standard sense. We will often omit the adjective PPT/efficient when discussing functions – by default we assume all such functions are efficient. We denote the support set of a distribution D by $Sup(D)$, and write $x \in D$ to indicate $x \in Sup(D)$. We denote the statistical distance between two random variables X and Y by $\Delta[X, Y]$. We call $f : \mathbb{N} \to \mathbb{R}$ negligible if $f(\lambda) < 1/P(\lambda)$, for any poly P and sufficiently large λ. For two ensembles $X = \{X_i\}_{i\in\mathbb{N}}$ and $\{Y_i\}_{i\in\mathbb{N}}$ of random variables we say X is computationally indistinguishable from Y, denoted $X \equiv^c Y$, if for any bit-valued, PPT distinguisher D, we have $|\Pr[D(X_\lambda) = 1] - \Pr[D(Y_\lambda) = 1]| = negl(\lambda)$. We say X and Y are statistically indistinguishable, denoted by $X \equiv^s Y$, if $f(\lambda) = \Delta[X_\lambda, Y_\lambda]$ is negligible. We write $X \equiv Y$ to mean X and Y are identically distributed.

We denote the n-th Cartesian power of a set S by S^n. All groups are assumed to admit efficient group operations, and to be commutative, but not necessarily cyclic, unless otherwise indicated. We denote bit values by lowercase letters $b, b_1, b_2, \ldots$, group elements by lowercase letters $g, q, \cdots$, bit vectors by boldface lowercase letters $\mathbf{b}, \ldots$ and group vectors by boldface lowercase letters $\mathbf{q}, \ldots$. For a vector $\mathbf{v}$ we denote the number of components in $\mathbf{v}$ by $|\mathbf{v}|$ and the i'th coordinate of $\mathbf{v}$ by $\mathbf{v}_i$, for $1 \leq i \leq |\mathbf{v}|$.

2.2 Syntax of encryption schemes

We assume that all cryptographic primitives (encryption, PRFs, etc) discussed in this paper, besides their usual algorithms, have a *parameter-generation* algorithm that produces *public parameters* (e.g., a group) used by all other algorithms. In situations where we talk about generating many keys it should be understood that all keys are sampled under the same public parameters, which were generated randomly at the beginning.

A *public-key encryption scheme* $\mathcal{E}$ is given by algorithms $(Param, G, E, Dec)$, all taking as input a *security parameter* 1^λ (that we make it explicit for $Param$ and G and implicit for other algorithms.) $Param$ takes input 1^λ, and outputs a public parameter, par. The *key-generation* algorithm, takes 1^λ and par and outputs public/secret keys, $(pk, sk) \leftarrow G(1^\lambda, par)$. The *encryption* algorithm E, takes a public key pk, a plaintext $m \in \mathcal{M}_\lambda$ (where $\mathcal{M}_\lambda$ is the *plaintext space*) and randomness $r \in \mathcal{R}_\lambda$ (where $\mathcal{R}_\lambda$ is the *randomness space*), and deterministically produces ciphertext $c = E_{pk}(b; r)$. Finally, the *decryption* algorithm takes a secret key sk and ciphertext c, and deterministically outputs $m = Dec_{sk}(c)$. For correctness, we require, for every $par \in Param(1^\lambda)$, $(pk, sk) \in G(1^\lambda, param)$, every m and $c \in E_{pk}(m)$, that $Dec_{sk}(E_{pk}(m)) = m$. We typically use $\mathcal{PK}_\lambda$ and $\mathcal{SK}_\lambda$ to refer to the public-key and secret-key spaces. Formally, $(\mathcal{PK}_\lambda, \mathcal{SK}_\lambda) = Sup(G(1^\lambda))$. We make the inclusion of $Param$ implicit henceforth.

2.3 Key-dependent-message security

In this paper we consider encryption schemes, whose generated secret keys are always bitstrings, but whose plaintext space may or may not be the single-bit space, e.g., it may be a group space. For the latter case, in order to make the notion of bitwise encryption of the secret key meaningful, we assume that a fixed mapping $(\{0,1\} \to \mathcal{M}_\lambda)$ is already in place. In the following, when we say $E_{pk}(b)$, where b is a bit, if E is a bit encryption algorithm, then we are encrypting the actual bit b, and otherwise, we are encrypting the element that b is mapped to. We now proceed to describe the notion of KDM$^{(n)}$ security for an arbitrary encryption scheme $\mathcal{E} = (G, E, Dec)$ (bit encryption or otherwise).

Assume that $F = \{F_\lambda\}$ is an ensemble (indexed by security parameter λ) of sets of functions, where for each $f \in F_\lambda$, it holds that $f : \mathcal{SK}_\lambda^n \to \{0,1\}$.

We define F-KDM$^{(n)}$ security through the following F-KDM$^{(n)}$ game, played between a challenger and an adversary. The challenger first chooses $b \leftarrow \{0,1\}$, generates $(pk_1, sk_1), \ldots, (pk_n, sk_n) \leftarrow G(1^\lambda)$,

and gives $pk_1, \ldots, pk_n$ to the adversary. The adversary, $\mathcal{A}$, given pk_i's, can repeatedly and adaptively, for $1 \leq i \leq n$, make queries of the form (i, f), $f \in F_\lambda$, or of the form (i, m), where $m \in \mathcal{M}_\lambda$, and in return,

- If $b = 0$, the challenger returns $E_{pk_i}(f(sk_1, \ldots, sk_n))$ in response to (i, f) and $E_{pk_i}(m)$ in response to (i, m); and
- If $b = 1$, the challenger returns $E_{pk_i}(0)$

$\mathcal{A}$ finally outputs a bit b'. We define the F-KDM$^{(n)}$ *advantage of* $\mathcal{A}$ as

$$Adv^{F\text{-KDM}^{(n)}}(\mathcal{A}) = \left|\Pr[b' = 1 | b = 0] - \Pr[b' = 1 | b = 1]\right|,$$

where the probabilities are computed over the coins of $\mathcal{A}$ and of the challenger.

We say that $\mathcal{E}$ is F-KDM$^{(n)}$-secure if for any $\mathcal{A}$ in the above game, it holds that $Adv^{F\text{-KDM}^{(n)}}(\mathcal{A}) = negl$. We now give the following definitions.

- Let $S_\lambda = \{Sel_{i,j}(x_1, ..., x_n) : 1 \leq i \leq n, 1 \leq j \leq |x_1|\}$, be the set of *selector* functions, where $Sel_{i,j}(x_1, ..., x_n) = (x_i)_j$. and let $\hat{S}_\lambda = \{\overline{Sel_{i,j}}(x_1, ..., x_n) : 1 \leq i \leq n, 1 \leq j \leq |x_1|\}$, where we write $\overline{f}(x)$ for $\overline{f(x)}$.
- We call $\mathcal{E}$ *n-circularly secure* if $\mathcal{E}$ is F-KDM$^{(n)}$ secure, where $F_\lambda = S_\lambda$.
- We call $\mathcal{E}$ *n-projection secure* if $\mathcal{E}$ is F-KDM$^{(n)}$ secure for $F_\lambda = S_\lambda \cup \hat{S}_\lambda$.

For private-key encryption it is convenient to work with the following equivalent definition of CPA security. (1) The challenger chooses $b \leftarrow \{0, 1\}$ and private key $sk \leftarrow G(1^\lambda)$ and gives the adversary oracle access to $E_{sk}(\cdot)$, which on input m returns $E_{sk}(m)$. (2) The adversary calls the encryption oracle adaptively any number of times and finally submits a single pair (m_0, m_1) and receives $E_{sk}(m_b)$. (3) The adversary continues to have oracle access and finally outputs its guess b'. We define the CPA-security advantage of the adversary as

$$\left|\Pr[b' = 1 | b = 0] - \Pr[b' = 1 | b = 1]\right|,$$

and call the scheme CPA secure if all adversaries have negligible advantage.

2.4 Leakage resilience

For $\ell : \mathbb{N} \to \mathbb{N}$, we say that encryption scheme $\mathcal{E} = (G, E, Dec)$ is *ℓ-length leakage resilient* if, for any adversary $\mathcal{A}$, the ℓ-leakage-advantage of $\mathcal{A}$, $Adv^{\ell\text{-}leak}(\mathcal{A})$, defined via the following game, is negligible.

- Setup: The challenger generates $(pk, sk) \leftarrow G(1^\lambda)$ and gives pk to $\mathcal{A}$.
- Leakage queries: $\mathcal{A}$ sends function $f : \mathcal{SK}_\lambda \to \{0, 1\}^*$ to the challenger, where $|f(sk)| \leq \ell(\lambda)$, and receives, in response, $f(sk)$.
- Challenge: $\mathcal{A}$ submits $(m_0, m_1) \in \mathcal{M}_\lambda^2$, and the challenger, samples $b \leftarrow \{0, 1\}$, and returns $E_{pk}(m_b)$ to $\mathcal{A}$. Finally, $\mathcal{A}$ returns an output bit b'.

We define $Adv^{\ell\text{-}leak}(\mathcal{A}) = \left|\Pr[b' = 1 | b = 0] - \Pr[b' = 1 | b = 1]\right|$. We say that $\mathcal{E}$ is *r-rate leakage resilient* (or has *resilience rate* r) if $\mathcal{E}$ is $r \cdot \log|\mathcal{SK}|$-length leakage resilient. Note that restricting $\mathcal{A}$ to a single leakage query is without loss of generality. In particular, the security definition does not become stronger if $\mathcal{A}$ is allowed to adaptively make multiple leakage queries provided that the total length of the bits leaked is bounded by $\ell(\lambda)$. (See [1] for further discussion.)

2.5 Auxiliary-input resilience (security)

Let $\mathcal{E} = (G, E, Dec)$ be an encryption scheme with public-key, secret-key and message spaces, respectively, $\mathcal{PK}_\lambda$, $\mathcal{SK}_\lambda$ and $\mathcal{M}_\lambda$. Throughout this Section we use f to refer to a function with domain $(\mathcal{PK}_\lambda, \mathcal{SK}_\lambda)$ and range $\mathcal{SK}_\lambda$. We follow the notation of [9]. For $\mathcal{E} = (G, E, Dec)$ we define *f-weak inversion* and *f-strong inversion* as follows. We say that f is *ϵ-strongly-uninvertible* under $\mathcal{E}$ if for any adversary $\mathcal{A}$, the probability that $\mathcal{A}$ outputs sk when given $(f(pk, sk), pk)$ is at most $\epsilon(\lambda)$, where the probability is taken over $\mathcal{A}$'s random coins and $(pk, sk) \leftarrow G(1^\lambda)$. Also, we say that f is *ϵ-weakly-uninvertible* under $\mathcal{E}$ if for any adversary $\mathcal{A}$, the probability that $\mathcal{A}$ outputs sk when given $f(pk, sk)$ is at most $\epsilon(\lambda)$, where

the probability is taken over $\mathcal{A}$'s random coins and $(pk, sk) \leftarrow G(1^\lambda)$. Let Aux_ϵ^{st} be the class of all ϵ-strongly-uninvertible functions and Aux_ϵ^{wk} be the class of all ϵ-weakly-uninvertible functions. Note that $Aux_\epsilon^{st} \subseteq Aux_\epsilon^{wk}$.

We say that $\mathcal{E}$ is *f-auxiliary-input secure* if any adversary $\mathcal{A}$ has a negligible advantage in the following game: $\mathcal{A}$ is given $(pk, f(pk, sk))$, where $(pk, sk) \leftarrow G(1^\lambda)$; $\mathcal{A}$ submits $(m_0, m_1) \in \mathcal{M}_\lambda^2$; $\mathcal{A}$ receives $E_{pk}(m_b)$, for $b \leftarrow \{0,1\}$; finally, $\mathcal{A}$ outputs bit b', and achieves the following advantage

$$|\Pr[b' = 1|b = 0] - \Pr[b' = 1|b = 1]|\,.$$

We say that $\mathcal{E}$ is *ϵ-weakly-auxiliary-input secure* (resp., ϵ-strongly-auxiliary-input secure) if $\mathcal{E}$ is *f-auxiliary-input secure* for any $f \in Aux_\epsilon^{st}$ (resp., Aux_ϵ^{wk}). We say $\mathcal{E}$ is auxiliary-input secure against *subexponentially-hard functions* if for some $c > 0$, $\mathcal{E}$ is $1/(2^{\lambda^c})$-strongly-auxiliary-input secure.

2.6 Properties of the base scheme

We give the definitions of the main properties that we need from the initial private-key encryption scheme.

Definition 1. *A private-key encryption scheme $\mathcal{E} = (G, E, Dec)$ provides* reproducibility *(or is* reproducible*) if there is an efficient function Rep such that for any $sk, sk' \in G(1^\lambda)$, $r \in \mathcal{R}_\lambda$ and $m_1, m_2 \in \mathcal{M}_\lambda$,*

$$Rep(E_{sk}(m_1; r), m_2, sk') = E_{sk'}(m_2; r).$$

Definition 2. *Let $\mathcal{E} = (G, E, Dec)$ be a private key encryption scheme where both $(\mathcal{R}_\lambda, +)$ and $(\mathcal{M}_\lambda, +)$ form groups. Then $\mathcal{E}$ is* additively homomorphic wrt plaintexts and randomness (PR-additively homomorphic) *if there is an efficient function Hom such that for every $sk \in G(1^\lambda)$, $m_1, m_2 \in \mathcal{M}_\lambda$, and $r_1, r_2 \in \mathcal{R}_\lambda$,*

$$Hom\left(E_{sk}(m_1; r_1), E_{sk}(m_2; r_2)\right) = E_{sk}(m_1 + m_2; r_1 + r_2).$$

For convenience we extend the notation of $Hom(\cdot)$ to define $Hom(c_1, \ldots, c_m)$ in the straightforward way. Henceforth, when discussing encryption schemes, we will use "homomorphism" as shorthand for "PR-additive homomorphism."

3 Constructions

We first fix some notation. Throughout this section we will be working with additive notation for groups with 0 denoting the identity element. For $\mathbf{g} = (g_1, \ldots, g_p) \in \mathbb{G}^p$ and $\mathbf{b} = (b_1, \ldots, b_p) \in \{0,1\}^p$ we define $\mathbf{b} \cdot \mathbf{g} = b_1 \cdot g_1 + \cdots + b_p \cdot g_p \in \mathbb{G}$, where, $0 \cdot g = 0$, and for $n \in \mathbb{N}$, we define $n \cdot g = g + (n-1) \cdot g$. We may sometimes drop $(\cdot)$ for better readability.

We present a generic construction that transforms a reproducible, homomorphic private-key encryption scheme into a public-key bit-encryption scheme, for which we prove our results. This always produces a bit-encryption scheme even if the base scheme is not. We then show how to adjust the construction, to maintain the plaintext space, at the cost of additional syntactic assumptions (which are satisfied by our schemes built based on homomorphic weak PRFs). For simplicity, we present (and prove the security of) the bit-encryption construction for the case where the base scheme is also bit encryption. The adaptation to the general case follows in a straightforward way.

Let $\mathcal{E} = (G, E, Dec, Hom, Rep)$ be a CPA-secure private-key bit-encryption scheme providing reproducibility (with the associated function *Rep*) and homomorphism (with the associated function *Hom*). Recall for homomorphism, both the message space, $\{0,1\}$, and the randomness space, $\mathcal{R}_\lambda$, form groups, which implies the plaintext group is just $\mathbb{Z}_2$. We now present the construction.

Construction 1 *(Single bit encryption): Let $\mathcal{E} = (G, E, Dec, Hom, Rep)$ be as above and let $l = l(k)$ be a value that we instantiate later.*

- *Key generation G': Choose the secret key as $\mathbf{s} \leftarrow \{0,1\}^l$ and the public key as $(E_{sk}(0; r_1), \ldots, E_{sk}(0; r_l), E_{sk}(0; \mathbf{s} \cdot \mathbf{r}))$, where $sk \leftarrow G(1^\lambda)$, $r_1, \ldots, r_l \leftarrow \mathcal{R}_\lambda$ and $\mathbf{r} = (r_1, \ldots, r_l)$.*

– *Encryption E': To encrypt bit b under public key $(c_1, \ldots, c_l, c_{l+1})$, do the following: choose $sk' \leftarrow G(1^\lambda)$ and return $(c'_1, \ldots, c'_l, c'_{l+1})$, where $c'_i = Rep(c_i, 0, sk')$, for $1 \leq i \leq l$, and $c'_{l+1} = Rep(c_{l+1}, b, sk')$.*

– *Decryption Dec': To decrypt $(c'_1, \ldots, c'_l, c'_{l+1})$ under secret key $\mathbf{s}$, letting $(i_1, \ldots, i_w)$ be the indices of non-zero bits of s, output 0 if $c'_{l+1} = Hom\left(c'_{i_1}, \ldots, c'_{i_w}\right)$, and 1 otherwise.*

The completeness of the scheme follows immediately. A few comments are in order. First, the encryption algorithm of the constructed scheme uses that of the base scheme, but by reusing randomness values hidden in the public key. Second, by using homomorphism and the fact that we are (functionally) able to reuse randomness across encryptions, we do not need any secret keys of the base scheme, e.g., sk, for decryption. Roughly, this is why proving circular security for the constructed scheme should not be much harder than proving CPA security. Third, the public key may alternatively be produced as $(c_1, \ldots, c_l, c_{l+1})$, where $c_i \leftarrow E_{sk}(0)$, for $1 \leq i \leq l$, and $c_{l+1} = Hom\left(c_{i_1}, \ldots, c_{i_w}\right)$, where $(i_1, \ldots, i_w)$ are the indices of non-zero bits of $\mathbf{s}$. This fact becomes essential in security proofs.

The above construction may easily be adapted to yield a bit-encryption scheme even if the original scheme is not bit encryption. Details are given Remark 1. Moreover, the above construction may be adapted to preserve the plaintext space, as explained in Section A.4.

3.1 Proofs of circular security

For clarity of exposition, in this preliminary version of the paper, we present the results wrt asymptotic security, without specifying the exact advantage functions.

We begin by introducing a game that will be used in proving our main results. Intuitively, the following experiment corresponds to a vector-encryption game, in which an adversary may interactively issue vectors of bits (of certain forms) to be encrypted, and each vector is component-wise encrypted while by reusing randomness across each fixed component of all vectors.

The randomness-sharing (RS) Game. Let (G, E, Dec) be a private-key bit-encryption scheme. As some notation, for $l \in \mathbb{N}$, we let $\mathbf{e}^l_i$, for $1 \leq i \leq l$, be the the vector of size l which has 1 in the ith position and 0 everywhere else, and $\mathbf{e'}^l_i$, for $1 \leq i \leq l$, be the vector of size l which has 1 in both its ith position and last position, and 0 everywhere else. We let $\mathbf{0}^l$ be the all-0 vector of size l. Finally, for $\mathbf{b} = (b_1, \ldots, b_l)$ and $\mathbf{r} = (r_1, \ldots, r_l)$, we define $E_{sk}(\mathbf{b}; \mathbf{r}) = (E_{sk}(b_1; r_1), \ldots, E_{sk}(b_l; r_l))$.

The game is parameterized over $p = p(\lambda)$ and is played as follows. The challenger chooses $b \leftarrow \{0, 1\}$ and $\mathbf{r} = (r_1, \ldots, r_p) \leftarrow \mathcal{R}^p_\lambda$, and the game proceeds as follows: the adversary repeatedly and adaptively makes queries of the form $\mathbf{e}$, where $\mathbf{e} \in \{\mathbf{e}^p_1, \ldots, \mathbf{e}^p_p, \mathbf{e'}^p_1, \ldots, \mathbf{e'}^p_p\}$, and in response to each such query, the challenger samples $sk \leftarrow G(1^\lambda)$ (using fresh coins for each query) and returns $E_{sk}(\mathbf{e}; \mathbf{r})$ if $b = 0$, and $E_{sk}(\mathbf{0}^p; \mathbf{r})$, otherwise. Finally, the adversary outputs a bit b' and its advantage is defined as:

$$Adv^{p\text{-rs}}(\mathcal{A}) = \Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1].$$

We give the following lemma used extensively in our subsequent proofs.

Lemma 1. *Assume $\mathcal{E} = (G, E, Dec, Rep)$ is a CPA-secure, private-key bit-encryption scheme that provides reproducibility. For any polynomial $p(\cdot)$, any adversary $\mathcal{A}$ in the p-RS game has a negligible advantage.*

We prove Lemma 1 using reproducibility of $\mathcal{E}$ and the fact that the set of possible queries is poly-bounded.

Proof. For convenience we drop the security parameter from all sets throughout the proof. We also introduce the following notation. For $\mathbf{b} = (b_1, \ldots, b_l)$ and $\mathbf{c} = (c_1, \ldots, c_l)$, define

$$Rep(\mathbf{c}, \mathbf{b}, sk) = (Rep(c_1, b_1, sk), \ldots, Rep(c_l, b_l, sk))$$

.

Assuming that $\mathcal{A}$ makes $t = t(\lambda)$ queries, $\mathbf{q}_1, \ldots, \mathbf{q}_t$ (where each $\mathbf{q}_i$ is of the form $\mathbf{e}^p_h$ or $\mathbf{e'}^l_u$ for some h and u) we define the hybrid W_i, for $1 \leq i \leq t$, as follows: first generate randomness vector $\mathbf{r} = (r_1, \ldots, r_p) \leftarrow \mathcal{R}^p$ and respond to queries as follows: in response to the j'th query, for $1 \leq j < i$, generate $sk_j \leftarrow G(1^\lambda)$ and return $E_{sk_j}(\mathbf{q}_j; \mathbf{r})$ (i.e., encryption of the actual vector); and in response to

the w'th query, for $w \geq i$, generate $sk_w \leftarrow G(1^\lambda)$ and return $E_{sk_w}(\mathbf{0}^p; \mathbf{r})$ (i.e., encryption of the all-zero vector). Note that W_1 and W_{t+1} match exactly the view of the adversary produced under the the RS game when $b = 1$ and $b = 0$, respectively. Thus, for the rest of the proof, we show how to reduce an adversary that can distinguish between W_i and W_{i+1}, for some i that $1 \leq i \leq t$, to an adversary against CPA security; the whole proof then follows using a standard hybrid argument.

Assume that $\mathcal{A}'$ can distinguish between W_i and W_{i+1} with a non-negligible advantage. Noting that W_i and W_{i+1} only differ in the way that the answer to the ith query is made, and that each query vector can take at most $2p$ different values, we guess the ith query vector (that is going to be issued by $\mathcal{A}'$), call the LOR-CPA oracle, which is parameterized over an unknown secret key, on the guessed vector to receive $\mathbf{c} = (c_1, \ldots, c_p)$, and start simulating $\mathcal{A}'$ as follows: in response to the j'th query, $\mathbf{q}_j$, for $1 \leq j < i$, we generate $sk_j \leftarrow G(1^\lambda)$ and return $Rep(\mathbf{c}, \mathbf{q}_j, sk_j)$; in response to the ith query we return $\mathbf{c}$ (if our guess for $\mathbf{q}_i$ was incorrect, we stop and return a random bit); and in response to the w'th query, $\mathbf{q}_w$, for $w > i$, we generate $sk_w \leftarrow G(1^\lambda)$ and return $Rep(\mathbf{c}, \mathbf{0}^p, sk_w)$. Now it is easy to see that, if our guessing for the ith query was correct, depending on whether the CPA-challenge bit was zero or one, the resulting experiment matches exactly either W_i or W_{i+1}. This completes the proof. □

We define the following property.

Property 1. Define the following property for a homomorphic bit-encryption scheme (G, E, Dec): for any $sk \in G(1^\lambda)$ and $b \in \{0,1\}$ and $r \in \mathcal{R}_\lambda$, from $E_{sk}(b; r)$ one may efficiently compute $E_{sk}(1; 0)$, i.e., the encryption of 1 under sk based on the identity randomness value.

Property 1 is needed in our proofs and since not seemingly implied by our other assumptions, we need to explicitly assume it. Under our general PRF-related assumptions, however, this property is readily satisfied.

We recall the following simple consequence of the leftover hash lemma [25].

Lemma 2. *Assume $\mathbb{G}$ is a group with $o = |\mathbb{G}|$, and D is a distribution over $\{0,1\}^l$. Then, it holds that*

$$\Delta\left[(\mathbf{g}, \mathbf{b} \cdot \mathbf{g}), \mathbf{g}'\right] \leq \frac{1}{2}\sqrt{\frac{o}{2^{H_\infty(D)}}},$$

where $\mathbf{g} \leftarrow \mathbb{G}^l$, $\mathbf{g}' \leftarrow \mathbb{G}^{l+1}$ and $\mathbf{b} \leftarrow D$.

We are now ready to prove the projection security of our constructed scheme. For simplicity, and since it already contains the main ideas, we first give the proof for 1-projection security, and then extend it to the n-projection case. We assume, without loss of generality, that an adversary against projection security (and against all other KDM-related games, as discussed throughout) never asks for an encryption of a constant message, i.e., it always asks for an encryption of a secret-key bit or its negation.

Theorem 1. *Let $\mathcal{E} = (G, E, Dec, Hom, Rep)$ be a CPA-secure private-key bit-encryption scheme providing homomorphism, reproducibility and Property 1. For any constant $c > 1$, by taking $l = l(\lambda) = c \cdot \log(|\mathcal{R}_\lambda|)$, the scheme built in Construction 1 is 1-projection secure.*

Proof. To represent the 1-projection game more concisely, we denote:
- *enc-secret*(i) encrypt the ith bit of the secret key; and
- *enc-secret*$(\bar{i})$ encrypt the negation of the ith bit of the secret key.

We introduce a series of hybrid games and show no adversary can distinguish between any two adjacent games. The first game corresponds to the real-encryption circular-security game, while the last game is the one where we always encrypt 0. Letting x_i be the adversary's output in *Game-i*, we write *Game-i* $\equiv^c$ *Game-j* to indicate $|\Pr[x_i = 1] - \Pr[x_j = 1]| = negl$. In all these games, whenever we write $sk \leftarrow G(1^\lambda)$ we mean that sk is chosen freshly, so we keep using the same variable sk inside each game whenever we are producing a new key.

Game-0: real encryption. This game provides a view to the adversary that is identical to that under the projection security game in which the challenge bit is zero. Formally,
- the adversary is given $(c_1, \ldots, c_l, c_{l+1}) =^{\text{def}} (E_{sk}(0; r_1), \ldots, E_{sk}(0; r_l), E_{sk}(0; r_{l+1}))$ as the public key, where $sk \leftarrow G(1^\lambda)$, $\mathbf{r} = (r_1, \ldots, r_l) \leftarrow \mathcal{R}_\lambda^l$, $\mathbf{s} \leftarrow \{0,1\}^l$ is the secret key and $r_{l+1} = \mathbf{s} \cdot \mathbf{r}$. Letting $(h_1, \ldots, h_w)$ be the indices of non-zero bits of s, note that $c_{l+1} = Hom(c_{h_1}, \ldots, c_{h_w})$.

– In response to *enc-secret*(i) (resp., *enc-secret*$(\bar{i})$), we return $(c'_1, \ldots, c'_l, c'_{l+1})$, where $sk \leftarrow G(1^\lambda)$, $c'_i = E_{sk}(0; r_i)$, for $1 \leq i \leq l$, and $c'_{l+1} = E_{sk}(\mathbf{s}_i, r_{l+1})$ (Resp., $c'_{l+1} = E_{sk}(\bar{\mathbf{s}}_i, r_{l+1})$). Again we emphasize that sk is chosen freshly for each query. Also, note that $c'_{l+1} = Hom(c'_{h_1}, \ldots, c'_{h_w}, E_{sk}(\mathbf{s}_i; 0))$ (resp., $c'_{l+1} = Hom(c'_{h_1}, \ldots, c'_{h_w}, E_{sk}(\bar{\mathbf{s}}_i; 0))$).

Game-1: In this game we generate the public key exactly as in *Game*-0, but we reply to *enc-secret* queries in a special manner. Formally, we generate $\mathbf{s} \leftarrow \{0,1\}^l$, let $(h_1, \ldots, h_w)$ be the indices of non-zero bits of $\mathbf{s}$ and act as follows.

– The adversary is given $(c_1, \ldots, c_l, c_{l+1})$ as the public key, where $sk \leftarrow G(1^\lambda)$, $(r_1, \ldots, r_l) \leftarrow \mathcal{R}_\lambda^l$, $c_i = E_{sk}(0; r_i)$ for $1 \leq i \leq l$ and $c_{l+1} = Hom(c_{h_1}, \ldots, c_{h_w})$;
– In response to an *enc-secret*(u) (resp., *enc-secret*$(\bar{u})$) query, we return $(c'_1, \ldots, c'_l, c'_{l+1})$, where $sk \leftarrow G(1^\lambda)$ and $c'_{l+1} = Hom(c'_{h_1}, \ldots, c'_{h_w}, E_{sk}(\mathbf{s}_u; 0))$ (resp., $c'_{l+1} = Hom(c'_{h_1}, \ldots, c'_{h_w}, E_{sk}(\bar{\mathbf{s}_u}; 0))$), but now $c'_i = E_{sk}(0; r_i)$, for any $1 \leq i \neq u \leq l$ and $c'_u = E_{sk}(1; r_u)$.

Now using the fact that c_{l+1}, for the public key, and c'_{l+1}'s, for encryption queries, are formed similarly (by applying Hom) under both *Game*-0 and *Game*-1 (in particular without needing to have $r_1, \ldots, r_l$) and also the fact that all r_i's, for $1 \leq i \leq l$, are sampled independently of $\mathbf{s}$, it can easily be verified that any adversary that can distinguish between *Game*-0 and *Game*-1 can be reduced to break the l-RS security of $\mathcal{E}$ (which is a contradiction by Lemma (1)). Also, note that for this last statement we need to make use of Property 1, that is we can compute $E_{sk}(1; 0)$ from another arbitrary encryption under sk.

Moreover, note that under this game, the distribution of the public key and the distributions of responses to *enc-secret*(i)'s and to *enc-secret*$(\bar{i})$'s are, respectively, as follows:

$$\begin{aligned}
&(E_{sk}(0; r_1), \ldots, E_{sk}(0; r_l), E_{sk}(0; r_{l+1})) \\
&(E_{sk'}(0; r_1), \ldots, E_{sk'}(0; r_{i-1}), E_{sk'}(1; r_i), E_{sk'}(0; r_{i+1}), \ldots, E_{sk'}(0; r_l), E_{sk'}(0; r_{l+1})) \\
&(E_{sk'}(0; r_1), \ldots, E_{sk'}(0; r_{i-1}), E_{sk'}(1; r_i), E_{sk'}(0; r_{i+1}), \ldots, E_{sk'}(0; r_l), E_{sk'}(1; r_{l+1}))\,, \qquad (1)
\end{aligned}$$

where $sk, sk' \leftarrow G(1^\lambda)$ (note that sk' is sampled using fresh coins for each query), $\mathbf{s} \leftarrow \{0,1\}^l$ and $\mathbf{r} = (r_1, \ldots, r_l) \leftarrow \mathcal{R}^l$ and $r_{l+1} = \mathbf{s} \cdot \mathbf{r}$.

Game-2: This game proceeds exactly as in *Game*-1, except we now sample r_{l+1} independently of all other $r'_i s$. Namely, we sample $(r_1, \ldots, r_l, r_{l+1}) \leftarrow \mathcal{R}_\lambda^{l+1}$ and form the public key and responses to the adversary's queries exactly as in Equation (1). Now given the fact the entire game (i.e., producing the public key and answering to the adversary's queries) can be simulated by only having $(r_1, \ldots, r_l, r_{l+1})$, by Lemma (2), and our choice of l, which is $l = c \cdot \log(|\mathcal{R}_\lambda|)$, we conclude that *Game*-1 $\equiv^s$ *Game*-2.

Game-3: In this game we again sample r_{l+1} independently of other $r'_i s$, but reply to all queries as "encryptions" of zero. That is, we generate $(r_1, \ldots, r_l, r_{l+1}) \leftarrow \mathcal{R}_\lambda^{l+1}$ and form the public key and responses to the adversary's queries as follows:

$$\begin{aligned}
&(E_{sk}(0; r_1), \ldots, E_{sk}(0; r_l), E_{sk}(0; r_{l+1})) && \text{public key} \\
&(E_{sk'}(0; r_1), \ldots, E_{sk'}(0; r_l), E_{sk'}(0; r_{l+1})) && \text{response to all queries} \qquad (2)
\end{aligned}$$

where, again, sk' is sampled freshly for each query. Now using the fact that all r_i's are sampled independently, and also that for each query sk' is generated using fresh coins, we obtain that any adversary that can distinguish between *Game*-2 and *Game*-3 can be reduced to break the $(l+1)$-RS security of $\mathcal{E}$ (which is a contradiction by Lemma (1)).

Game-4: In this game we change back the distributions of r_i's to the original, but answer to all the adversary's queries as encryptions of zero. That is, we generate $\mathbf{s} \leftarrow \{0,1\}^l$, $\mathbf{r} = (r_1, \ldots, r_l) \leftarrow \mathcal{R}_\lambda^l$, let $r_{l+1} = \mathbf{s} \cdot \mathbf{r}$, and form the public key and responses to the adversary's queries as follows:

$$\begin{aligned}
&(E_{sk}(0; r_1), \ldots, E_{sk}(0; r_l), E_{sk}(0; r_{l+1})) && \text{public key} \\
&(E_{sk'}(0; r_1), \ldots, E_{sk'}(0; r_l), E_{sk'}(0; r_{l+1})) && \text{responses to all queries} \qquad (3)
\end{aligned}$$

Now, since *Game*-3 and *Game*-4 differ only in the way that $(r_1, \ldots, r_l, r_{l+1})$ is generated, and again using the fact that $l = c \cdot \log(|\mathcal{R}_\lambda|)$ and applying Lemma (2), we conclude that *Game*-3 $\equiv^s$ *Game*-4. This completes the proof. □

Remark 1. For simplicity, we presented Construction 1, which always produces a bit-encryption scheme, only for the case that the base scheme is also a bit-encryption one. It is easy to modify Construction 1, so it works (i.e., produces a bit-encryption scheme) even if the original scheme is not bit-encryption. To do this, we just need to fix a nonzero $m \in \mathcal{M}_\lambda$ and change the encryption algorithm of Construction 1 by replacing b with $b \cdot m$. As for security, we need to change the RS game, so that the queried vectors can now be of the form $(0, \ldots, 0, -m, 0, \ldots, 0)$, $(0, \ldots, 0, m, 0, \ldots, 0)$ or $(0, \ldots, m, 0, \ldots, 0, m)$. Now the proof of Lemma 1 follow with straightforward changes. As for the the proof of Theorem 1, we change Game-1 (i.e., the simulated encryption game), so that (i) in response to *enc-secret*(u) we return $(c'_1, \ldots, c'_l, c'_{l+1})$, where $c'_{l+1} = Hom(c'_{h_1}, \ldots, c'_{h_w}, E_{sk}(\mathbf{s}_u \cdot m; 0))$, in which $c'_i = E_{sk}(0; r_i)$, for any $1 \leq i \neq u \leq l$ and $c'_u = E_{sk}(-m; r_u)$; also, in response to *enc-secret*$(\bar{u})$, we return $(c'_1, \ldots, c'_l, c'_{l+1})$, where $c'_{l+1} = Hom(c'_{h_1}, \ldots, c'_{h_w}, E_{sk}(\bar{\mathbf{s}_u} \cdot m; 0))$, where $c'_i = E_{sk}(0; r_i)$, for any $1 \leq i \neq u \leq l$ and $c'_u = E_{sk}(m; r_u)$. Now Game-0 $\equiv^c$ Game-1 follows by the new version of Lemma 1. Moreover, in Game-1, the view of an adversary against *enc-secret*(u) and *enc-secret*$(\bar{u})$ would be, respectively,

$$(E_{sk'}(0; r_1), \ldots, E_{sk'}(0; r_{i-1}), E_{sk'}(-m; r_i), E_{sk'}(0; r_{i+1}), \ldots, E_{sk'}(0; r_l), E_{sk'}(0; r_{l+1}))\,,$$

$$(E_{sk'}(0; r_1), \ldots, E_{sk'}(0; r_{i-1}), E_{sk'}(m; r_i), E_{sk'}(0; r_{i+1}), \ldots, E_{sk'}(0; r_l), E_{sk'}(m; r_{l+1}))\,,$$

and since all the plaintexts encrypted under sk''s are independent of $\mathbf{s}$, the rest of the proof follows as that of Theorem 1.

We extend to the multiple key case, by building on the ideas of [9], which generalize the DDH-based technique of [8]. Most of the technical details of the proof are already covered in Theorem 1; the full proof is given in Section A.1.

Theorem 2. *Let $\mathcal{E} = (G, E, Dec, Hom, Rep)$ be a CPA-secure private-key bit-encryption scheme providing homomorphism, reproducibility and Property 1. For any constant $c > 1$, by taking $l = cn \log(|\mathcal{R}_\lambda|) + \omega(\log \lambda)$, the scheme built in Construction 1 is n-projection secure.*

3.2 Proof of leakage resilience

An outline of the proof of leakage resilience was given in the introduction, and so we refer the reader to Section A.2 for the full proof. .

Theorem 3. *Let $\mathcal{E} = (G, E, Dec, Hom, Rep)$ be a CPA-secure private-key bit-encryption scheme providing homomorphism, reproducibility and Property 1. Then, the scheme built in Construction 1 is $(l - \log|\mathcal{R}_\lambda| - u)$-length leakage resilient, for any $u \in \omega(\log \lambda)$. Moreover, by taking $l = \omega(\log|\mathcal{R}_\lambda|) + \omega(\log \lambda)$, the constructed scheme achieves a $(1 - o(1))$ resilience rate.*

3.3 Proof of auxiliary-input security

We first consider weak-auxiliary-input security and then discuss the extension to the strong-auxiliary case. The proof of the following theorem essentially follows the same line of arguments as that of Theorem 3 (sketched in the introduction), except for replacing randomness extraction with pseudorandomness extraction [22]. We refer the reader to Section A.3 for the full proof.

Theorem 4. *Let $\mathcal{E} = (G, E, Dec, Hom, Rep)$ be a CPA-secure private-key bit-encryption scheme providing homomorphism, reproducibility and Property 1. Let $\mathcal{E}'$ be the scheme constructed from $\mathcal{E}$ according to Construction 1. For any poly-bounded $l = l(\lambda)$ and negligible function $\epsilon = \epsilon(\lambda)$, it holds that $\mathcal{E}'$ is ϵ-weakly-auxiliary-input secure.*[5]

Remark 2. As in previous work [14, 9] we can prove strong auxiliary-input security for $\mathcal{E}'$ with respect to subexponentially-hard functions by working with a modification of Construction 1, letting $(c_1, \ldots, c_l) = (E_{sk}(0; r_1), \ldots, E_{sk}(0; r_l))$ be the public parameters of the scheme, and letting the public key be computed, under secret key $\mathbf{s}$, as $Hom(c_{i_1}, \ldots, c_{i_w})$, where $\langle i_1, \ldots, i_w \rangle$ are the indices of non-zero bits of $\mathbf{s}$. Now since a public key under the new scheme has at most $l' = |\mathcal{R}_\lambda|$ different values we can obtain $\frac{\epsilon}{l'}$-strong auxiliary-input security from ϵ-weak-auxiliary-input security. This last step follows since,

[5] In order to have this statement useful, it should hold that $\frac{1}{2^l} \leq \epsilon$, because otherwise the statement will be vacuously true.

for any scheme with l' different public keys, if recovering sk from $f(pk, sk)$ is ϵ/l'-hard (i.e., succeeds with a probability at most ϵ/l'), recovering sk from $(f(pk, sk), pk)$ is ϵ-hard. Finally, we mention that the proof of multiple-key circular security (Theorem 2) extends to the setting above which contains public parameters.

4 Homomorphic hash-proof systems

We review the notion of a *homomorphic hash-proof system* (HHPS), first defined by [13], which will be used in Section 5.

A HHPS $\mathsf{HHPS} = (\mathsf{Param}, \mathsf{Priv}, \mathsf{Pub})$ is described as follows. The *setup* algorithm $\mathsf{Param}(\cdot)$ takes as input a security parameter 1^λ and randomly outputs *public parameters* $\mathsf{HP} = (\mathsf{C}, \mathsf{C}_v, \mathsf{W}, \mathsf{K}, \mathsf{SK}, \mathsf{PK}, \mu : \mathsf{SK} \to \mathsf{PK}, \Lambda : \mathsf{SK} \times \mathsf{C} \to \mathsf{K})$, where, C is called the set of *ciphertexts*, $\mathsf{C}_v \subseteq \mathsf{C}$ the set of *valid ciphertexts*, W the set of *witnesses*, K the set of *plaintexts*, SK the set of *secret keys* and PK the set of *public keys*. We should point out that all sets in the output of Param are indeed descriptions of their actual sets; we make this fact implicit henceforth. Each $\mathsf{c} \in \mathsf{C}_v$ admits a witness $\mathsf{w} \in \mathsf{W}$ of its membership in C_v, meaning that there exists a PPT relation R such that

$$\mathsf{c} \in \mathsf{C}_v \Leftrightarrow \exists \mathsf{w} \in \mathsf{W} \text{ s.t. } \mathsf{R}(\mathsf{c}, \mathsf{w}) = 1.$$

We assume that it is efficiently possible to generate a uniform element from C_v along with a corresponding witness, and also to sample uniformly from the sets SK and K. The efficient *private evaluation* algorithm Priv takes as input $\mathsf{sk} \in \mathsf{SK}$ and $\mathsf{c} \in \mathsf{C}$, and deterministically computes $\mathsf{Priv}_{\mathsf{sk}}(\mathsf{c}) = \Lambda(\mathsf{sk}, \mathsf{c})$. The efficient *public evaluation* algorithm Pub, takes as input $\mathsf{pk} = \mu(\mathsf{sk})$, $\mathsf{c} \in \mathsf{C}_v$ and a witness w for c, and deterministically computes $\mathsf{Pub}_{\mathsf{pk}}(\mathsf{c}, \mathsf{w}_\mathsf{c}) = \Lambda(\mathsf{sk}, \mathsf{c})$. Finally, we require HHPS to satisfy the following properties.

Subset membership: For every adversary $\mathcal{A}$, given all the public parameters of the scheme, it holds that

$$|\Pr[\mathcal{A}(\mathsf{c_v}) = 1] - \Pr[\mathcal{A}(\mathsf{c_{inv}}) = 1]| = negl(\lambda),$$

where, $\mathsf{c_v} \leftarrow \mathsf{C}_v$, $\mathsf{c_{inv}} \leftarrow \mathsf{C} \setminus \mathsf{C}_v$ and the probabilities are computed over the random coins of the adversary and over C and C_v, which are taken from the output of $\mathsf{Param}(1^\lambda)$.

Smoothness: It holds that $\Delta[(\mathsf{pk}, \mathsf{Priv}_{\mathsf{sk}}(\mathsf{c}), \mathsf{c}), (\mathsf{pk}, \mathsf{k}, \mathsf{c})] = negl(\lambda)$, where $\mathsf{c} \leftarrow \mathsf{C} \setminus \mathsf{C}_v$, $\mathsf{k} \leftarrow \mathsf{K}$, $\mathsf{sk} \leftarrow \mathsf{SK}$ and $\mathsf{pk} = \mu(\mathsf{sk})$.

Homomorphism: $(\mathsf{C}, +)$, $(\mathsf{C}_v, +)$ and $(\mathsf{K}, +)$ admit groups (with efficient group operations), and, for every sk, $\Lambda(sk, \cdot)$ constitutes a homomorphism, i.e., for every $\mathsf{sk} \in \mathsf{SK}$ and $\mathsf{c}_1, \mathsf{c}_2 \in \mathsf{C}$, it holds that,

$$\Lambda(\mathsf{sk}, \mathsf{c}_1) + \Lambda(\mathsf{sk}, \mathsf{c}_2) = \Lambda(\mathsf{sk}, \mathsf{c}_1 + \mathsf{c}_2)^6.$$

5 Realizations from homomorphic weak PRFs

We now show how to realize our base encryption primitive under *homomorphic weak pseudorandom functions (PRFs)*, an extension of weak PRFs, introduced by Naor and Reingold [33]. Then we give instantiations of weak homomorphic PRFs under *homomorphic hash-proof systems* [13]. Finally, we weaken the pseudorandomness property even further, making it suitable for instantiations under the subgroup indistinguishability assumption of [9].

Let $K = \{K_\lambda\}_{\lambda \in \mathbb{N}}$, $D = \{D_\lambda\}_{\lambda \in \mathbb{N}}$ and $R = \{R_\lambda\}_{\lambda \in \mathbb{N}}$ be ensembles of sets. For each security parameter λ and each $k \in K_\lambda$ we have an associated function $f_k : D_\lambda \to R_\lambda$, and let $F_\lambda = \{f_k \mid k \in K_\lambda\}$ and $F = \{F_\lambda\}_{\lambda \in \mathbb{N}}$. We call F *homomorphic* if for every $\lambda \in \mathbb{N}$, every $k \in K_\lambda$, we have that both D_λ and R_λ are groups and that $f_k \in F_\lambda$ is a homomorphism from D_λ to R_λ.

[6] We remark that in many settings the homomorphism of C_v is implied by that of C: Especially in the standard setting, where the set of valid ciphertexts is defined as those, for which the value of $\Lambda(\mathsf{sk}, \cdot)$, for any sk is determined solely from the ciphertexts itself and $\mu(\mathsf{sk})$. However, we put it as a separate condition just to be as general as possible.

Definition 3. *[33] We call F a* weak pseudorandom function family *if for any polynomial function $p = p(\lambda)$, it holds that $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$, where*

$$\mathcal{DS}_1 \equiv (d_1, r_1), \ldots, (d_p, r_p)$$
$$\mathcal{DS}_2 \equiv (d_1, f_k(d_1)), \ldots, (d_p, f_k(d_p)),$$

for $k \leftarrow K_\lambda$, $d_1, \ldots, d_p \leftarrow D_\lambda$ and $r_1, \ldots, r_p \leftarrow R_\lambda$.[7]

Note that a PRF in the standard sense is trivially a weak PRF. Now we show that the standard method of constructing CPA-secure private-key encryption from a PRF, when applied to a homomorphic weak PRF, results in the kind of encryption primitive we need.

Lemma 3. *Assuming the existence of a homomorphic weak pseudorandom function family, there exists CPA-secure private-key encryption which is both reproducible and homomorphic, and also satisfies Property 1.*

Proof. Let F be a homomorphic weak PRF. Construct $\mathcal{E} = (G, Enc, Dec)$, whose plaintext and randomness spaces are the range and domain of F as follows. $G(1^\lambda)$ returns $k \leftarrow K_\lambda$; $Enc_k(p_1; d_1)$ returns $(d_1, f_k(d_1) + p_1)$; and $Dec_k(d, r)$ returns $r - f_k(d)$. CPA-security, homomorphism and reproducibility of $\mathcal{E}$ are clear. Finally, note that $E_k(p; 0) = (0, p)$, which verifies Property 1. □

We now show how to construct a homomorphic weak PRF from a HHPS.

Theorem 5. *Assuming the existence of a HHPS, there exists a homomorphic weak PRF.*

Proof. (Outline) Assume that $\mathsf{HHPS} = (\mathsf{Param}, \mathsf{Priv}, \mathsf{Pub})$ is a HHPS. Let $\mathsf{HP} = (\mathsf{C}, \mathsf{C}_v, \mathsf{W}, \mathsf{K}, \mathsf{SK}, \mathsf{PK}, \mu : \mathsf{SK} \to \mathsf{PK}, \Lambda : \mathsf{SK} \times \mathsf{C} \to \mathsf{K})$ be the public parameters of HHPS produced by running Param. The tuple HP will also be the public parameters of our PRF, F, constructed as follows. We set $K_\lambda = \mathsf{SK}$, $D_\lambda = \mathsf{C}_v$ and $R_\lambda = \mathsf{K}$, and define $f_{\mathsf{sk}}(c) = \Lambda_{\mathsf{sk}}(\mathsf{c})$. We have that both C_v and K admit groups and that $f_{\mathsf{sk}}(c_1) + f_{\mathsf{sk}}(c_2) = f_{\mathsf{sk}}(\mathsf{c}_1 + \mathsf{c}_2)$, which implies homomorphism for PRF F. We defer the proof of weak pseudorandomness security for F to Section A.5 (which uses standard facts about HHPS). □

For instantiations based on the subgroup indistinguishability (SG) assumption we further weaken the condition of weak pseudorandomness, but note that it may be possible to directly instantiate homomorphic weak PRFs, perhaps in more inefficient ways, under SG-related assumptions (say, under QR). We leave this to future work. Informally, the new weakened condition requires that, given random access to the underlying f_k which on every call returns $(d, f_k(d))$ for a random d, the output of f_k has high *pseudoentropy*, as opposed to being pseudorandom. Moreover, we assume that a new function is provided which allows us to hide a given element by extracting the pseudoentropy provided.

For the following definitions let $M = \{M_\lambda\}_{\lambda \in \mathbb{N}}$ and $R' = \{R'_\lambda\}_{\lambda \in \mathbb{N}}$ be ensembles of sets and $\{f^{\text{hid}} : R_\lambda \times M_\lambda \to R'_\lambda\}_{\lambda \in \mathbb{N}}$ be an ensemble of functions. Note that f^{hid} is not a keyed function, so we have an ensemble of functions.

Definition 4. *Let F and f^{hid} be as above. We call F a weak$_1$ pseudorandom function family with respect to f^{hid} (shortly (F, f^{hid}) is a weak$_1$ pseudorandom function family) if for any poly $p = p(n)$ and adversary $\mathcal{A}$, it holds that $\mathcal{A}$ has a negligible advantage in the following game: $\mathcal{A}(1^\lambda)$ outputs $(m_0, m_1) \in M_\lambda^2$ and its advantage is defined as*

$$\left|\Pr\left[\mathcal{A}\left((d_1, r_1), \ldots, (d_p, r_p), (d, r'_0)\right) = 1\right] - \Pr\left[\mathcal{A}\left((d_1, r_1), \ldots, (d_p, r_p), (d, r'_1)\right) = 1\right]\right|, \tag{4}$$

where $d, d_1, \ldots, d_p, d_{p+1} \leftarrow D_\lambda$, $k \leftarrow K_\lambda$, $r_1 = f_k(d_1)$, $\ldots, r_p = f_k(d_p)$ and $r'_b = f^{hid}(f_k(d), m_b)$, for $b \in \{0, 1\}$.

Definition 5. *Let F and f^{hid} be as in Definition 4. We call (F, f^{hid}) homomorphic if F is homomorphic, both M_λ and R'_λ, for any $\lambda \in \mathbb{N}$, form groups and that it holds $f^{hid}(r_1, m_1) + f^{hid}(r_2, m_2) = f^{hid}(r_1 + r_2, m_1 + m_2)$. Also, we call f^{hid} (or simply (F, f^{hid})) invertible if there is a function Inv such that for any $\lambda \in \mathbb{N}$, $r' \in R'_\lambda$ and $m \in M_\lambda$, it holds that $Inv(r', f^{hid}(r', m)) = m$.*

[7] Each of the domain space or the key space may come with an associated distribution, but we leave this point implicit for simplicity.

Theorem 6. *Assuming the existence of a homomorphic weak PRF F, there exists a weak$_1$ PRF (F', f^{hid}) that is both invertible and homomorphic, and assuming the existence of a weak$_1$ PRF (F, f^{hid}) that is both invertible and homomorphic, there exists a CPA-secure encryption scheme that is reproducible, homomorphic and satisfies Property 1.*

Proof. For the first part, assume F is a homomorphic weak PRF with input D_λ and output R_λ. Define $f^{\text{hid}}\colon R_\lambda \times R_\lambda \mapsto R_\lambda$ as $f^{\text{hid}}(r_1, r_2) = r_1 + r_2$. Now it is easy to see that (F, f^{hid}) is weak$_1$, homomorphic and invertible.

For the second part, assume that (F, f^{hid}) is a weak$_1$ PRF that is invertible (with the associated function Inv) and homomorphic. Assume that F's key set, input set and output sets are, respectively, K_λ, D_λ and R_λ, and that $f^{\text{hid}}\colon R_\lambda \times M_\lambda \mapsto R'_\lambda$. Define $\mathcal{E} = (G, E, Dec)$, with key space K_λ, plaintext space M_λ and randomness space D_λ, as follows. $G(1^\lambda)$ outputs $k \leftarrow K_\lambda$; $E_k(m; d) = (d, f^{\text{hid}}(f_k(d), m))$ and $Dec_k(d, r') = Inv(d, f^{\text{hid}}(f_k(d), m))$. Reproducibility follows since $\mathcal{E}$ reveals the randomness in the clear. Also, homomorphism of $\mathcal{E}$ follows from the homomorphism of (F, f^{hid}). Also, Property 1 follows by considering that $f_k(id)$ (here id is the identity randomness element) is indeed the identity element of R_λ (which we assume is publicly known). Finally, CPA security of $\mathcal{E}$ follows immediately from weak$_1$ security of (F, f^{hid}). (Recall that we work with the model of CPA security in which the adversary may make many single encryption queries but only one left-or-right query.) □

Finally, we show how to instantiate weak$_1$ PRFs under the subgroup indistinguishability assumption. See Section A.6 for SG-related definitions.

Theorem 7. *Assuming the subgroup indistinguishability (SG) assumption holds there exists a weak$_1$ PRF (F, f^{hid}) that is both invertible and homomorphic.*

Proof. (Outline) We sketch the proof for the special case of the QR assumption, and give the proof for the general SG case in Section A.6. We build PRF (F, f^{hid}) as follows. We run $N \leftarrow \mathsf{RSAGen}(1^\lambda)$ and let N be the public parameter of F, and let $K_\lambda = \mathbb{Z}_{N^2}$, $D_\lambda = R_\lambda = \mathcal{QR}_N$, $M_\lambda = \{0, 1\}$ and $R'_\lambda = \mathbb{Z}^*_N$. All operations below are modulo N. We define $f_k(d) = d^k$ and $f^{\text{hid}}(d', b) = (-1)^b \cdot d'$. Under the additive group for $\{0, 1\}$ we have that both f_k and f^{hid} are homomorphic. Invertibility of (F, f^{hid}) also follows immediately. Finally, for weak$_1$ security we need to show that $(d_1, \ldots, d_p, d, d_1^k, \ldots, d_p^k, d^k) \equiv^c (d_1, \ldots, d_p, d, d_1^k, \ldots, d_p^k, , -d^k)$, where $d_1, \ldots, d_p, d \leftarrow \mathcal{QR}_N$ and $k \leftarrow \mathbb{Z}_{N^2}$, which follows using standard techniques. (See, e.g., [13, 29] and also [9] for a stronger statement.) □

6 KDM amplification and leakage resilience

We show that Applebaum's KDM-amplification method [3], which, informally speaking, shows that projection security is sufficient for obtaining "rich-KDM" security, preserves both types of leakage resilience. For simplicity we focus on the case of bit encryption and 1-KDM security. The corollaries we obtain, however, hold if these conditions are relaxed.

We recall that all functions and all ensembles of sets of functions are in the PPT uniform model (as we have assumed throughout the paper). We identify an efficiently computable function $\{f_\lambda\colon \{0,1\}^l \mapsto \{0,1\}\}_\lambda$ with an ensemble of circuits $\{c_\lambda\}_\lambda$, and say that f has size p if, for any λ, the circuit c_λ has size at most $p(\lambda)$. We say an ensemble of sets of functions $F = \{F_\lambda\}_\lambda$ is p-bounded if for every λ and every $f \in F_\lambda$, f has size p. The following theorem is a special case of the results of [3].

Theorem 8. *([3]) Assume that $F = \{F_\lambda\}_\lambda$ is a fixed p-bounded ensemble of sets of functions and $\mathcal{E} = (G, E, Dec)$ is a 1-projection-secure encryption scheme. The encryption scheme $\mathcal{E}' = (G, E', D')$, constructed as follows, is F-KDM$^{(1)}$ secure: $E'_{pk}(b) = E_{pk}(Sim(b))$ and $D'_{sk}(C) = Rec(D_{sk}(C))$. Here Sim is a randomized function and Rec is a deterministic function, both of which are constructed based on F, through the procedure of* randomized encoding of functions*. The correctness of decryption follows by the properties of randomized encoding (see [3] for details).*

Theorem 9. *Let $\mathcal{E}$ and $\mathcal{E}'$ be as in Theorem 8. Then assuming that $\mathcal{E}$ is r-rate leakage resilient (resp., ϵ-auxiliary input secure) then $\mathcal{E}'$ is r-rate leakage resilient (resp., ϵ-auxiliary input secure).*

Proof. This follows by noting that the constructed scheme $\mathcal{E}'$ has the same key generation algorithm as that $\mathcal{E}$. We consier the leakage resilience case; the proof for the auxiliary-input case is entirely the same. Assume $\mathcal{A}'$ wins against ℓ-length leakage resilience of $\mathcal{E}'$; we build $\mathcal{A}$ that wins ℓ-length leakage resilience of $\mathcal{E}'$ by simulating $\mathcal{A}'$ as follows: $\mathcal{A}$ runs $\mathcal{A}'(pk)$, where pk is the public key that $\mathcal{A}$ receives; when $\mathcal{A}'$ asks the leakage query f, $\mathcal{A}$ makes the same query from its oracle and give $f(sk)$ to $\mathcal{A}'$; finally, when $\mathcal{A}'$ submits (b_0, b_1), $\mathcal{A}$ submits $(Sim(b_0), Sim(b_1))$ to its oracle and give the ciphertext that it receives to $\mathcal{A}'$. Thus, $\mathcal{A}$ achieves the same advantage as $\mathcal{A}'$ does, and the proof is complete. □

We now obtain the following corollary, by combining Theorems 1, 3, 6 and 8.

Corollary 1. *Assuming the existence of an invertible, homomorphic weak$_1$ PRF (implied by, in particular, the existence of a homomorphic weak PRF), for any poly p and any fixed p-bounded function family F, there exists a scheme $\mathcal{E}'$ which (at the same time) (1) is F-KDM secure, (2) achieves a $(1 - o(1))$ resilience rate, and (3) is auxiliary-input secure against subexponentially-hard functions.*

7 Conclusions and open problems

We introduced a general construction for building projection-secure, leakage resilient encryption from CPA-secure encryption schemes that satisfy some special properties. We showed how to realize our encryption primitive under homomorphic weak PRFs, and the latter under various specific assumptions. Finally, we showed that Applebaum's KDM amplification method preserves both types of leakage resilience, which allowed us, under our assumptions, to obtain schemes with rich-KDM security and strong forms of leakage security. Using our general approach, we were able to explain the results of [8, 9], and also to obtain an HHPS-based projection-secure scheme. We have not yet investigated the applicability of our framework to the LWE (and related) assumptions and we leave an investigation of this to future work. The main question arising from our work is if it is possible to construct circularly-secure encryption from even more general assumptions.

8 Acknowledgments

We would like to thank Dan Boneh for helpful suggestions.

References

1. Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *TCC*, volume 5444 of *LNCS*, pages 474–495. Springer, 2009.
2. Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-key encryption in the bounded-retrieval model. In Gilbert [21], pages 113–134.
3. Benny Applebaum. Key-dependent message security: Generic amplification and completeness. In *Advances in Cryptology - EUROCRYPT 2011*, pages 527–546, 2011.
4. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, 2009.
5. Boaz Barak, Iftach Haitner, Dennis Hofheinz, and Yuval Ishai. Bounded key-dependent message security. In Gilbert [21], pages 423–444.
6. Mihir Bellare, Alexandra Boldyreva, and Jessica Staddon. Randomness re-use in multi-recipient encryption schemeas. In *Public Key Cryptography - PKC 2003*, pages 85–99, 2003.
7. John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *SAC 2002, Selected Areas in Cryptography*, volume 2595 of *LNCS*, pages 62–75. Springer, 2002.
8. Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In *CRYPTO 2008*, pages 108–125, 2008.
9. Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In *Advances in Cryptology - CRYPTO 2010*, pages 1–20, 2010.

10. Zvika Brakerski, Shafi Goldwasser, and Yael Tauman Kalai. Black-box circular-secure encryption beyond affine functions. In Ishai [28], pages 201–218.
11. Zvika Brakerski and Gil Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In *CRYPTO 2011, volume 6841 of LNCS*, pages 543–560. Springer, 2011.
12. Seung Geol Choi and Hoeteck Wee. Lossy trapdoor functions from homomorphic reproducible encryption. *Inf. Process. Lett.*, 112(20):794–798, 2012.
13. Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, 2002.
14. Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *Theory of Cryptography, TCC 2010*, volume 5978 of *LNCS*, pages 361–381. Springer, 2010.
15. Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *STOC 2009*, pages 621–630, 2009.
16. Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.
17. Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *STOC 1991*, pages 542–552, 1991.
18. Cynthia Dwork, editor. *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*. ACM, 2008.
19. Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS 2008*, pages 293–302, 2008.
20. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Dwork [18], pages 197–206.
21. Henri Gilbert, editor. *Advances in Cryptology - EUROCRYPT 2010,*, volume 6110 of *LNCS*. Springer, 2010.
22. Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 25–32. ACM, 1989.
23. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.
24. Shai Halevi and Huijia Lin. After-the-fact leakage in public-key encryption. In Ishai [28], pages 107–124.
25. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.
26. Carmit Hazay, Adriana López-Alt, Hoeteck Wee, and Daniel Wichs. Leakage-resilient cryptography from minimal assumptions. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 160–176. Springer, 2013.
27. Dennis Hofheinz and Dominique Unruh. Towards key-dependent message security in the standard model. In *Advances in Cryptology - EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 108–126. Springer, 2008.
28. Yuval Ishai, editor. *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, volume 6597 of *LNCS*. Springer, 2011.
29. Eike Kiltz, Krzysztof Pietrzak, Martijn Stam, and Moti Yung. A new randomness extraction paradigm for hybrid encryption. In *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 590–609. Springer, 2009.
30. Venkata Koppula, Kim Ramchen, and Brent Waters. Separations in circular security for arbitrary length key cycles. *IACR Cryptology ePrint Archive*, 2013:683, 2013.
31. Tal Malkin, Isamu Teranishi, and Moti Yung. Efficient circuit-size independent public key encryption with kdm security. In *Advances in Cryptology – EUROCRYPT 2011*, pages 507–526, 2011.
32. Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *Theory of Cryptography, TCC 2004*, volume 2951 of *LNCS*, pages 278–296. Springer, 2004.
33. Moni Naor and Omer Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. *J. Comput. Syst. Sci.*, 58(2):336–375, 1999.
34. Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. *SIAM J. Comput.*, 41(4):772–814, 2012.
35. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC 1990*, pages 427–437, 1990.
36. Noam Nisan and David Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996.
37. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Dwork [18], pages 187–196.
38. Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology - CRYPTO 1991*, volume 576 of *LNCS*, pages 433–444. Springer, 1991.

39. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC 2005*, pages 84–93, 2005.
40. Ron Rothblum. Homomorphic encryption: From private-key to public-key. In Ishai [28], pages 219–234.
41. Ron Rothblum. On the circular security of bit-encryption. In *TCC*, pages 579–598, 2013.

Appendices

A Detailed proofs

In this section we provided detailed proofs for all the theorems which appeared in Section 3.

A.1 n-Projection security

For the proof of n-projection security we first need to give some standard entropy notions. We start by reviewing some facts related to randomness extraction, which we will use in the subsequent proof. We denote the *min-entropy* of a distribution D by $H_\infty(D)$, defined as $H_\infty(X) = \min_{d\in D}[\log(\frac{1}{\Pr[D=d]})]$. In cryptographic applications, when two random variables X and Y are jointly distributed, one typically considers the notion of *average min entropy*, formalized by Dodis et al. [16], which measures the expected unpredictability of X given a random value y of Y. Formally,

$$\tilde{H}_\infty(X|Y) = -\log\left(E_{y\leftarrow Y}(2^{-H_\infty(X|Y=y)})\right) = -\log\left(E_{y\leftarrow Y}(max_x \Pr[X=x|Y=y])\right).$$

Strong randomness extractors [36] allow one to extract almost uniformly-random bits from any source with sufficient min-entropy. We will work with the generalized notion of *average-case extractors*, formalized in [16], which allow one to extract randomness from sources about which some information may be leaked.

Definition 6. *([16]) We say that $Ext : \{0,1\}^n \times S \to R$ is an average case (k,ϵ)-extractor if for any pair of distributions (D,X), where D takes values in $\{0,1\}^n$ and $\tilde{H}_\infty(D|X) \geq k$, it holds that*

$$\Delta((Ext(D,U_S),U_S,X),(U_R,U_S,X)) \leq \epsilon,$$

where U_S is independent of (D,X).

Note that in the above definition it is necessary to sample the seed U_S independently of X because otherwise we will not be able to extract any randomness. (e.g., The adversary may set X to be some prefix of $Ext(D,U_S)$, trivially distinguishing between the two distributions.) A useful standard lemma proved in [16] is the following.

Lemma 4. *For any (X,Y) it holds that $\tilde{H}_\infty(X|Y) \geq H_\infty(X) - \log|Sup(Y)|$.*

A family of functions $\{h : D \to R\}_{h\in H}$ is called *universal* if for all $x_1, x_2 \in D$, with $x_1 \neq x_2$, it holds that

$$\Pr_{h\leftarrow H}[h(x_1) = h(x_2)] \leq \frac{1}{|R|}.$$

We may sometimes write a family of functions $\{h : D \to R\}_{h\in H}$ as a single function $\mathcal{H} : D \times H \to R$, where $\mathcal{H}(d,h) = h(d)$. Dodis et al. [16] show that universal hash functions, which are known to be a good strong extractor [25], act well in the average case as well, in the sense that, using a family of universal hash functions, one can extract as many random bits from D in the presence of leaked information X as from a random variable Y with min-entropy $\tilde{H}_\infty(D|X)$ (but without any leaked information).

Lemma 5. *([16]) If $Ext : \{0,1\}^n \times S \to R$ is a family of universal hash functions, then for any random variables D taking values in $\{0,1\}^n$ and random variable X, it holds that*

$$\Delta\left((Ext(D,U_S),U_S,X),(U_R,U_S,X)\right) \leq 1/2\sqrt{2^{-\tilde{H}_\infty(D|X)}|R|}$$

We are now ready to give the proof for n-projection security.

Theorem 2. *Let $\mathcal{E} = (G, E, Dec, Hom, Rep)$ be a CPA-secure private-key bit-encryption scheme providing homomorphism, reproducibility and Property 1. Then, for any constant $c > 1$, by taking $l = cn \log(|\mathcal{R}_\lambda|) + \omega(\log \lambda)$, the scheme built in Construction 1 based on l is n-projection secure.*

Proof. To represent the n-projection game more concisely, we use the following notation for denoting the adversary's queries.

- *enc-secret*(i, j, h) encrypt the jth bit of sk_i under the hth public key; and
- *enc-secret*$(i, \bar{j}, h)$ encrypt the negation of the jth bit of sk_i under the hth public key.

We also need the following notation. In all the games below we denote the ith secret key as $\mathbf{s}_i$ and define $\mathbf{s}_{ij}$ to be the jth-bit of the ith secret key. If $\mathbf{v}$ is a vector of size l, and $\mathbf{s} \in \{0,1\}^l$ we define $\mathbf{v}[\mathbf{s}]$ to be $(\mathbf{v}_{i_1}, \ldots, \mathbf{v}_{i_w})$, where $i_1 < \cdots < i_w$ are the indices of nonzero bits of $\mathbf{s}$. We also recall the notation $\mathbf{e}_i^l$, $\mathbf{0}^l$, and $E_{sk}(\mathbf{b}; \mathbf{r})$, defined at the beginning of Section 3.1.

.

Game-0: real encryption. This game provides a view for the adversary that is identical to that under the projection security game in which the challenge bit is zero. The idential view is produced by using the algorithm Hom to produce the public keys and to reply to encryption queries. (See the first paragraph following Construction 1.) Generate $\mathbf{s}_1, \ldots, \mathbf{s}_n \leftarrow \{0,1\}^l$ and $\mathbf{r}_1, \ldots, \mathbf{r}_n \leftarrow \mathcal{R}_\lambda^l$. (Here $\mathbf{s}_i$ will be the secret key of the ith "user" and $\mathbf{r}_i$ the randomness values used to produce the public key for the ith user.) For each h, where $1 \leq h \leq n$, do:

- the adversary is given $(\mathbf{c}, Hom(\mathbf{c}[\mathbf{s}_h]))$ as the public key, where $\mathbf{c} = E_{sk}(\mathbf{0}^l, \mathbf{r}_h)$, for $sk \leftarrow G(1^\lambda)$.
- In response to *enc-secret*$((i,j),h)$ (resp., *enc-secret*$((i,\bar{j}),h)$), we return $(\mathbf{c}', Hom(\mathbf{c}'[\mathbf{s}_h], E_{sk'}(\mathbf{s}_{ij}; 0)))$ (resp., $(\mathbf{c}', Hom(\mathbf{c}'[\mathbf{s}_h], E_{sk'}(\bar{\mathbf{s}}_{ij}; 0)))$, where $sk' \leftarrow G(1^\lambda)$, $\mathbf{c}' = E_{sk'}(\mathbf{0}^l; \mathbf{r}_h)$.

Game-1: In this game, we generate the public keys exactly as in *Game*-0, but we reply to *enc-secret* queries in the following way. For each h, where $1 \leq h \leq n$, in response to *enc-secret*$((i,j),h)$ (resp., *enc-secret*$((i,\bar{j}),h)$), we return $(\mathbf{c}', Hom(\mathbf{c}'[\mathbf{s}_h], E_{sk'}(\mathbf{s}_{ij}; 0)))$ (resp., $(\mathbf{c}', Hom(\mathbf{c}'[\mathbf{s}_h], E_{sk'}(\bar{\mathbf{s}}_{ij}; 0)))$, where $sk' \leftarrow G(1^\lambda)$ and $\mathbf{c}' = E_{sk'}(\mathbf{e}_h^l; \mathbf{r}_h)$.

Now, as in the proof for 1-projection security, it is not hard to show that any adversary that can distinguish between *Game*-0 and *Game*-1 can be reduced to break the l-RS security of $\mathcal{E}$ (which is a contradiction by Lemma (1)).

Moreover, notice that, for each h, under this game, the distributions of the public key, the response to *enc-secret*$((i,j),h)$ and to *enc-secret*$((i,\bar{j}),h)$ queries, respectively, may be formed equivalently as:

$$\begin{aligned}
&(E_{sk_1}(\mathbf{0}^l; \mathbf{r}_h), E_{sk_1}(0; r_{h,l+1})) && \text{public key}\\
&(E_{sk_2}(\mathbf{e}_j^l; \mathbf{r}_h), E_{sk_2}(\mathbf{s}_{ij} + \mathbf{s}_{hj}; r_{h,l+1})) && \text{response to } \textit{enc-secret}((i,j),h)\\
&(E_{sk_3}(\mathbf{e}_j^l; \mathbf{r}_h), E_{sk_3}(\bar{\mathbf{s}}_{ij} + \mathbf{s}_{hj}; r_{h,l+1})) && \text{response to } \textit{enc-secret}((i,\bar{j}),h),
\end{aligned} \tag{5}$$

where $r_{h,l+1} = \mathbf{s}_h \cdot \mathbf{r}_h$ and each sk_p, for $1 \leq p \leq 3$ is chosen freshly for each query. Defining $\mathbf{d}_w = \mathbf{s}_1 + \mathbf{s}_\mathbf{w}$, i.e., bitwise binary addition, for $2 \leq w \leq n$, by having only the tuple $D = (\mathbf{r}_1, \ldots, \mathbf{r}_n, r_{1,l+1}, \ldots, r_{n,l+1}, \mathbf{d}_2, \ldots, \mathbf{d}_n)$, one can perfectly simulate the view of the adversary for this game, i.e., for each $1 \leq h \leq n$, produce the distributions of Equation 5.

Game-2: This game proceeds exactly as in *Game*-1, except that we now sample $r_{h,l+1}$, for all $1 \leq h \leq n$, independently of $\mathbf{r}_h$; the rest of the game remains unchanged. Now we show that the following two distributions are statistically indistinguishable, which by the last statement of *Game*-1, it implies that *Game*-1 $\equiv^s$ *Game*-2;

$$\begin{aligned}
D &= (\mathbf{r}_1, \ldots, \mathbf{r}_n, \mathbf{s}_1 \cdot \mathbf{r}_1, \ldots, \mathbf{s}_n \cdot \mathbf{r}_n, \mathbf{d}_2, \ldots, \mathbf{d}_n),\\
D' &= (\mathbf{r}_1, \ldots, \mathbf{r}_n, r_1, \ldots, r_n, \mathbf{d}_2, \ldots, \mathbf{d}_n),
\end{aligned} \tag{6}$$

where $\mathbf{r}_1, \ldots, \mathbf{r}_n \leftarrow \mathcal{R}_\lambda^l$, $\mathbf{s}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n \leftarrow \{0,1\}^l$, and, for $2 \le i \le n$, $\mathbf{s}_i = \mathbf{s}_1 \oplus \mathbf{d}_i$. To show Equation 6, observe that

$$\begin{aligned} y &= \tilde{H}_\infty(\mathbf{s}_1 | \mathbf{s}_2 \cdot \mathbf{r}_2, \ldots, \mathbf{s}_n \cdot \mathbf{r}_n, \mathbf{r}_2, \ldots, \mathbf{r}_n, \mathbf{d}_2, \ldots, \mathbf{d}_n) \\ &\ge \tilde{H}_\infty(\mathbf{s}_1 | \mathbf{r}_2, \ldots, \mathbf{r}_n, \mathbf{d}_2, \ldots, \mathbf{d}_n) - (n-1)\log|\mathcal{R}_\lambda| \\ &= \tilde{H}(\mathbf{s}_1) - (n-1)\log|\mathcal{R}_\lambda| \\ &\ge \log|\mathcal{R}_\lambda| + \omega(\log\lambda), \end{aligned}$$

which implies $D \equiv^s D'$ by Lemma (5).

Game-3: This game proceeds as in *Game*-2, but we reply to all queries as "encryptions" of zero. That is, generate $\mathbf{r}_1, \ldots, \mathbf{r}_n \leftarrow \mathcal{R}_\lambda^n$, and for for each h, where $1 \le h \le n$, form the hth public key as $(E_{sk}(\mathbf{0}^l; \mathbf{r}_h), E_{sk}(0; r_{h,l+1}))$, where $r_{h,l+1}$ is chosen independently of $\mathbf{r}_h$, and respond to every query under this public key as $(E_{sk'}(\mathbf{0}^l; \mathbf{r}_h), E_{sk'}(0; r_{h,l+1}))$. (Note that sk''s is chosen freshly for each query.)

Since $r_{h,l+1}$ for every h is sampled independently of $\mathbf{r}_h$, again we can easily see that any adversary that can distinguish between *Game*-2 and *Game*-3 can be reduced to break the $(l+1)$-RS security of $\mathcal{E}$ (which is a contradiction by Lemma (1)), implying *Game*-2 $\equiv^s$ *Game*-3.

Game-4: This game runs exactly like *Game*-4, except that, for every $1 \le h \le n$, we now define $r_{h,l+1} = \mathbf{s}_h \cdot \mathbf{r}_h$. Using the same information-theoretic arguments as in *Game*-2, we obtain that *Game*-3 $\equiv^s$ *Game*-4. The view of the adversary under this game is exactly that under the n-projection security game when the challenge bit is one. This concludes the proof. □

A.2 Detailed proof for leakage resilience

Theorem 3. *Let $\mathcal{E} = (G, E, Dec, Hom, Rep)$ be a CPA-secure private-key bit-encryption scheme providing homomorphism, reproducibility and Property 1. Then, the scheme built in Construction 1 is $(l - \log|\mathcal{R}_\lambda| - u)$-length leakage resilient, for any $u \in \omega(\log\lambda)$. Moreover, by taking $l = \omega(\log|\mathcal{R}_\lambda|) + \omega(\log\lambda)$, the constructed scheme achieves a $(1 - o(1))$ resilience rate.*

Proof. Note that the second statement readily follows from the first one. To prove the first statement, observe that since the constructed scheme is bit encryption, we may assume, without loss of generality, that the adversary always outputs $(0,1)$ as its challenge query. Now we prove the first statement through a series of games, where the first game matches the real leakage game, and in the last game the view of the adversary is independent of the challenge bit, b. We conclude the proof by showing that the views of the adversary under any two adjacent games under the same $b \in \{0,1\}$ are computationally indistinguishable. In each game below, we let f be the leakage query of the adversary, and, letting x_i be the adversary's output in *Game-i*, we write *Game-i* $\equiv^c$ *Game-j* to indicate that $|\Pr[x_i = 1] - \Pr[x_j = 1]| = negl$. Fix $b \in \{0,1\}$ for the rest of the proof.

Game-0: In this game we answer to the adversary's queries exactly as in the real leakage game, where the challenge bit is b. Thus, at the end of the game, the view of the adversary is $(c_1, \ldots, c_l, c_{l+1}, f(\mathbf{s}), c'_1, \ldots, c'_l, c'_{l+1})$, produced as follows: $\mathbf{s} \leftarrow \{0,1\}^l$, $\mathbf{r} = (r_1, \ldots, r_l) \leftarrow \mathcal{R}_\lambda^l$, $r_{l+1} = \mathbf{s} \cdot \mathbf{r}$, $sk \leftarrow G(1^\lambda)$, $sk' \leftarrow G(1^\lambda)$, $c_i = E_{sk}(0; r_i)$, for $1 \le i \le l+1$, $c'_j = E_{sk'}(0; r_j)$, for $1 \le j \le l$, and $c'_{l+1} = E_{sk'}(b; r_{l+1})$.

Notice that the view of the adversary may identically be produced as

$$(c_1, \ldots, c_l, c''_{l+1}, f(\mathbf{s}), c'_1, \ldots, c'_l, c'''_{l+1}), \tag{7}$$

where all c_i's and c'_i's are produced as above, and $c''_{l+1} = Hom(c_{h_1}, \ldots, c_{h_w})$ and $c'''_{l+1} = Hom(c'_{h_1}, \ldots, c'_{h_w}, E_{sk'}(b; 0))$ with $\langle h_1, \ldots, h_w \rangle$ being the indices of non-zero bits of $\mathbf{s}$.

Game-1. In this game we generate the secret key, the public key and the response to the leakage query exactly as in *Game*-0, but we reply to the encryption challenge query in a special way. Formally, choose $\mathbf{s} \leftarrow \{0,1\}^l$, $\mathbf{r} = (r_1, \ldots, r_l) \leftarrow \mathcal{R}_\lambda^l$, and return $(c_1, \ldots, c_l, c_{l+1})$ as the public key, $f(\mathbf{s})$ as the response to leakage query and $(c'_1, \ldots, c'_l, c'_{l+1})$ as the challenge ciphertext, where c_i's and c'_i's are produced as follows: generate $sk \leftarrow G(1^\lambda)$, $sk' \leftarrow G(1^\lambda)$, $\mathbf{b} = (b_1, \ldots, b_l) \leftarrow \{0,1\}^l$, $c_i = E_{sk}(0; r_i)$, for $1 \le i \le l$, $c'_j = E_{sk'}(b_j; r_j)$, for $1 \le j \le l$, and finally $c_{l+1} = Hom(c_{h_1}, \ldots, c_{h_w})$ and $c'_{l+1} = Hom(c'_{h_1}, \ldots, c'_{h_w}, E_{sk'}(b; 0))$, where again $\langle h_1, \ldots, h_w \rangle$ are the indices of non-zero bits of $\mathbf{s}$.

To show that *Game*-0 $\equiv^c$ *Game*-1, note that both games can be simulated (in exactly the same way) by only having $\mathcal{D} = (c_1, \ldots, c_l, c'_1, \ldots, c'_l)$ (see Equation 7), and that the distributions produced for $\mathcal{D}$ under the two games are computationally indistinguishable. (This latter follows from the reproducibility and CPA security of the scheme.) Thus, we conclude *Game*-0 $\equiv^c$ *Game*-1. Notice that, under Game-1, the view of the adversary is

$$\big(E_{sk}(0;r_1), \ldots, E_{sk}(0;r_l), E_{sk}(0;\mathbf{s}\cdot\mathbf{r}), f(\mathbf{s}), \\ (E_{sk'}(b_1;r_1), \ldots, E_{sk'}(b_l;r_l), E_{sk'}(b_{l+1}+b;\mathbf{s}\cdot\mathbf{r})\big), \quad (8)$$

where $b_{l+1} = \mathbf{s}\cdot\mathbf{b}$.

Game-2. This game runs exactly as in *Game*-2 (Equation 8), except that now we generate $b_{l+1} \leftarrow \{0,1\}$. First, notice that both *Game*-1 and *Game*-2 can be simulated (in exactly the same manner) by only having

$$\mathcal{D} = (\mathbf{r}, r_{l+1}, b_1, \ldots, b_l, b_{l+1}, f(\mathbf{s})), \quad (9)$$

where under both games we sample $\mathbf{s} \leftarrow \{0,1\}^l$ and $r_{l+1} = \mathbf{s}\cdot\mathbf{r}$, while under one game, we sample b_{l+1} freshly at random, and under the other game we set $b_{l+1} = \mathbf{s}\cdot\mathbf{b}$. Thus, it suffices to show that the distributions of $\mathcal{D}$ induced under the two games are indistinguishable. We have,

$$\begin{aligned}\tilde{H}_\infty(\mathbf{s}|\mathbf{r}, \mathbf{s}\cdot\mathbf{r}, f(\mathbf{s})) &\geq \tilde{H}_\infty(\mathbf{s}|\mathbf{r}, f(\mathbf{s})) - log|\mathcal{R}_\lambda| \\ &= \tilde{H}_\infty(\mathbf{s}|f(\mathbf{s})) - log|\mathcal{R}_\lambda| \\ &\geq H_\infty(\mathbf{s}) - log|\mathcal{R}_\lambda| - l + \log|\mathcal{R}_\lambda| + u \\ &= u,\end{aligned}$$

and by assumption we have $u \in \omega(\log\lambda)$. Now, since all r_i's and b_i's, for $1 \leq i \leq l$ are independent, and that the family of hash functions corresponding to the inner product is universal, we may use Lemma 5 to deduce that the distribution of $\mathcal{D}$ under *Game*-1 and *Game*-2 are statistically indistinguishable, and hence *Game*-1 $\equiv^s$ *Game*-2. To apply Lemma 5, take $D = \mathbf{s}$, $U_S = (b_1, \ldots, b_l)$ and $X = (\mathbf{r}, \mathbf{s}\cdot\mathbf{r}, f(\mathbf{s}))$. Notice that *Game*-2 produces the same views for the adversary under $b = 0$ and $b = 1$, and hence the proof is complete. □

A.3 Proof of auxiliary-input security

The proof of Theorem 4 follows similarly to that of Theorem 3, except for one step, where we replace real-randomness extraction with pseudorandomness extraction, which is made possible considering that the inner product function (which is the basis of our public key and encryption construction) allows to extract pseudorandomness, as formalized below. The following theorem is due to Goldreich and Levin [22], where we follow the presentation of [14], adapted to the binary field.

Theorem 10. *([22]) Assume that $l = l(\lambda)$ and $h\colon \{\{0,1\}^l \to \{0,1\}^*\}$ is a (possibly randomized) function and $\mathcal{D}$ is a distinguisher, where*

$$|\Pr[D(\mathbf{b}, b, h(\mathbf{s})) = 1] - \Pr[D(\mathbf{b}, b', h(\mathbf{s})) = 1]| = \delta(l), \quad (10)$$

where $\mathbf{s}, \mathbf{b} \leftarrow \{0,1\}^l$, $b \leftarrow \{0,1\}$ and $b' = \mathbf{s}\cdot\mathbf{b}$. Then there exists an inverter $\mathcal{A}$, for which it holds that

$$\Pr[\mathcal{A}(y) = \mathbf{s}] \in \Omega(\frac{\delta^3}{l}), \quad (11)$$

where $\mathbf{s} \leftarrow \{0,1\}^l$ and $y \leftarrow h(\mathbf{s})$.

We now give the proof of auxiliary-input security, which uses ideas from [14].

Theorem 4. *Let $\mathcal{E} = (G, E, Dec, Hom, Rep)$ be a CPA-secure private-key bit-encryption scheme providing homomorphism, reproducibility and Property 1. Let $\mathcal{E}'$ be the scheme constructed from $\mathcal{E}$ according to Construction 1. For any poly-bounded $l = l(\lambda)$ and negligible function $\epsilon = \epsilon(\lambda)$, it holds that $\mathcal{E}'$ is ϵ-weakly-auxiliary-input secure.*

Proof. The proof follows by introducing *Game*-0, *Game*-1 and *Game*-2 exactly as in the proof of Theorem 3 (except that now the function f is applied on both the secret key and the public key), and deriving *Game*-0 $\equiv^c$ *Game*-1 exactly as in there. To prove *Game*-1 $\equiv^c$ *Game*-2, however, we proceed as below. To prove *Game*-1 $\equiv^c$ *Game*-2, it suffices to show that

$$(b_1,\ldots,b_l,b_{l+1},f(PK,\mathbf{s}),\overbrace{E_{sk}(0;r_1),\ldots,E_{sk}(0;r_l),E_{sk}(0;r_{l+1})}^{PK}) \equiv^c \\ (b_1,\ldots,b_l,b'_{l+1},f(PK,\mathbf{s}),\underbrace{E_{sk}(0;r_1),\ldots,E_{sk}(0;r_l),E_{sk}(0;r_{l+1})}_{PK}), \tag{12}$$

where $\mathbf{s}\leftarrow\{0,1\}^l$, $b_1,\ldots,b_l,b_{l+1}\leftarrow\{0,1\}$, $b'_{l+1}=\mathbf{s}\cdot(b_1,\ldots,b_l)$, $\mathbf{r}=(r_1,\ldots,r_l)\leftarrow\mathcal{R}^l_\lambda$, $r_{l+1}=\mathbf{s}\cdot\mathbf{r}$ and $sk\leftarrow G(1^\lambda)$. (The reason that proving Equation 12 suffices to conclude *Game*-1 $\equiv^c$ *Game*-2 is because of the descriptions of *Game*-1 and *Game*-2, which are different only in the way that b_{l+1} is sampled, and the reproducibility property of the base scheme $\mathcal{E}$.)

By the assumption of the theorem, we know that it is ϵ-hard to recover $\mathbf{s}$ from $(PK,f(PK,\mathbf{s}))$. Now Equation 12 follows from Theorem 10, by defining the randomized function $h(\mathbf{s})=(PK,f(PK,\mathbf{s}))$, where all the values are sampled and defined as above, (note that the randomness of h comes from choosing $(r_1,\ldots,r_l)\leftarrow\mathcal{R}^l_\lambda$ and $sk\leftarrow G(1^\lambda)$), and considering that $\epsilon=negl(\lambda)$. (I.e., If there exists an adversary that can distinguish between the distributions in Equation 12 with a non-negligible probability, then there exists an adversary that, with a non-negligible probability, recovers $\mathbf{s}$ from $h(\mathbf{s})$, which is a contradiction by the choice of ϵ.) □

A.4 A scheme that preserves the plaintext space

Construction 1 yields a bit-encryption scheme, even when the scheme we start with has a larger plaintext space. We now describe a scheme that maintains the plaintext space of the original scheme, at the cost of making two additional syntactic assumption, aimed at making the decryption algorithm efficient. We stress here that these new conditions are required solely for the efficiency of decryption, and are not used anywhere in security proofs. We show that all these assumptions are satisfied for encryptions schemes built based on homomorphic weak PRFs.

Definition 7. *Assume $\mathcal{E}=(G,E,Dec,Hom,Rep)$ is a private-key scheme that provides homomorphism and reproducibility. We define two new conditions for the scheme.*

1. *There exists an efficient algorithm $RandInv$, such that for every $sk\in G(1^\lambda)$ and $r\in\mathcal{R}_\lambda$, it holds that*
$$RandInv(E_{sk}(0;r))=E_{sk}(0;-r)).$$
2. *For every $sk\in G(1^\lambda)$ and $m\in\mathcal{M}_\lambda$, one can efficiently compute m from the encryption of m under the identity element, $E_{sk}(m;0)$.*

We now give the plaintext-preserving construction.

Construction 2 *(Plaintext-preserving encryption)*
Assume that $\mathcal{E}=(G,E,Dec,Hom,Rep)$ satisfies the conditions in Definition 7.

- *Key generation G': Choose the secret key as $\mathbf{s}\leftarrow\{0,1\}^l$, and the public key as*
$$\left(E_{sk}(0;r_1),\ldots,E_{sk}(0;r_l),E_{sk}(0;\mathbf{s}\cdot\mathbf{r})\right),$$
where $sk\leftarrow G(1^\lambda)$, $r_1,\ldots,r_l\leftarrow\mathcal{R}_\lambda$ and $\mathbf{r}=(r_1,\ldots,r_l)$.
- *Encryption E': To encrypt plaintext $m\in\mathcal{M}_\lambda$ under public key $(c_1,\ldots,c_l,c_{l+1})$, do the following: choose $sk'\leftarrow G(1^\lambda)$ and return $(c'_1,\ldots,c'_l,c'_{l+1})$, where $c'_i=Rep(c_i,0,sk')$, for $1\le i\le l$, and $c'_{l+1}=Rep(c_{l+1},m,sk')$.*
- *Decryption D': To decrypt $(c'_1,\ldots,c'_l,c'_{l+1})$ under secret key s, letting $(i_1,\ldots,i_w)$ be the indices of non-zero bits of s, compute $c'=Hom(c_{i_1},\ldots,c_{i_w},c''_{l+1})$, where $c''_{l+1}=RandInv(c'_{l+1})$. Now recover m from c', using Condition 2 of Definition 7.*

In the following theorem we state the modified version of Theorems 2, 3 for the case where the initial scheme is not bit encryption. The proofs are entirely similar (except for a few simple changes) to the bit-encryption case, and so we omit them here.

Theorem 11. *Let $\mathcal{E} = (G, E, Dec, Hom, Rep)$ be a CPA-secure private-key encryption scheme providing homomorphism and reproducibility. Moreover, assume that $\mathcal{E}$ provides the conditions in Definition 7 (aimed at having efficient decryption) and Property 1. Then, the scheme $\mathcal{E}' = (G', E', D')$ built in Construction 2 has the following properties.*

- *for any constant $c > 1$, by taking $l = cn \log(|\mathcal{R}_\lambda|) + \omega(\log \lambda)$, $\mathcal{E}'$ is n-projection secure; and*
- *$\mathcal{E}'$ is $(l - \log|\mathcal{R}_\lambda| - \log|\mathcal{M}_\lambda| - u)$-length leakage resilient, for any $u \in \omega(\log \lambda)$. Moreover, by taking $l = \omega(\log|\mathcal{R}_\lambda|) + \omega(\log|\mathcal{M}_\lambda|) + \omega(\log \lambda)$, the constructed scheme achieves a leakage rate of $(1 - o(1))$.*

Corollary 2. *Assuming the existence of an invertible, homomorphic weak$_1$ PRF (F, f^{hid}) (which is implied by a homomorphic weak PRF), with $F_\lambda = \{f_k \colon D_\lambda \mapsto R_\lambda\}_{k \in K_\lambda}$ and $f^{hid} \colon R_\lambda \times M_\lambda \mapsto R'_\lambda$, there exists an encryption scheme with plaintext space M_λ and randomness space D_λ satisfying all the properties stated in Theorem 11.*

The corollary above follows from considering that encryption schemes built based on invertible, homomorphic weak$_1$ PRF, as per the proof of Theorem 6, satisfy both Conditions spelled out in Definition 7.

A.5 From HHPS to homomorphic weak PRFs

Theorem 5. *Assuming the existence of a HHPS, there exists a homomorphic weak PRF.*

Proof. Assume that $\mathsf{HHPS} = (\mathsf{Param}, \mathsf{Priv}, \mathsf{Pub})$ is a HHPS. Let

$$\mathsf{HP} = (\mathsf{C}, \mathsf{C}_v, \mathsf{W}, \mathsf{K}, \mathsf{SK}, \mathsf{PK}, \mu : \mathsf{SK} \to \mathsf{PK}, \Lambda : \mathsf{SK} \times \mathsf{C} \to \mathsf{K})$$

be the public parameters of HHPS produced by running Param. The tuple HP will also be the public parameters of our PRF, F, constructed as follows. We set $K_\lambda = \mathsf{SK}$, $D_\lambda = \mathsf{C}_v$ and $R_\lambda = \mathsf{K}$, and define $f_{(\mathsf{sk}}(c) = \Lambda_{\mathsf{sk}}(\mathsf{c})$. We have that both C_v and K admit groups and that $f_{\mathsf{sk}}(c_1) + f_{\mathsf{sk}}(c_2) = f_{\mathsf{sk}}(\mathsf{c}_1 + \mathsf{c}_2)$, which implies homomorphism for PRF F. To prove weak pseudorandomness for F we need to show that, for any $p = p(\lambda)$, it holds that $\mathcal{DS} \equiv^c \mathcal{DS}'$, where

$$\mathcal{DS} = (\mathsf{c}_1, \Lambda_{\mathsf{sk}}(\mathsf{c}_1)), \ldots, (\mathsf{c}_\mathsf{p}, \Lambda_{\mathsf{sk}}(\mathsf{c}_\mathsf{p}))$$
$$\mathcal{DS}' = (\mathsf{c}_1, \mathsf{k}_1), \ldots, (\mathsf{c}_\mathsf{p}, \mathsf{k}_\mathsf{p}),$$

for $\mathsf{c}_1, \ldots, \mathsf{c}_\mathsf{p} \leftarrow \mathsf{C}_v$, $\mathsf{k}_1, \ldots, \mathsf{k}_\mathsf{p} \leftarrow \mathsf{K}$ and $sk \leftarrow \mathsf{SK}$. To this end, for $0 \le i \le p$, we define $\mathcal{DS}_i$ as follows (for the rest of the proof we fix the way of sampling of the random variables that appear above):

$$\mathcal{DS}_i = ((\mathsf{c}_1, \Lambda_{\mathsf{sk}}(\mathsf{c}_1)), \ldots, (\mathsf{c}_\mathsf{i}, \Lambda_{\mathsf{sk}}(\mathsf{c}_\mathsf{i})), (\mathsf{c}_{\mathsf{i}+1}, \mathsf{k}_{\mathsf{i}+1}), \ldots, (\mathsf{c}_\mathsf{p}, \mathsf{k}_\mathsf{p}))\,. \tag{13}$$

Note that $\mathcal{DS} = \mathcal{DS}_0$ and $\mathcal{DS}' = \mathcal{DS}_p$. Now for each $0 \le i \le p$ we show that $\mathcal{DS}_i = \mathcal{DS}_{i+1}$, and this concludes the proof. Now $\mathcal{DS}_i = \mathcal{DS}_{i+1}$ easily follows from the fact that $(\mathsf{pk}, \mathsf{c}_{\mathsf{i}+1}, \Lambda_{\mathsf{sk}}(\mathsf{c}_{\mathsf{i}+1})) \equiv^c (\mathsf{pk}, \mathsf{c}_{\mathsf{i}+1}, \mathsf{k}_{\mathsf{i}+1})$ (which follows by combining the subset membership and smoothness properties of HHPS): Namely, given $(\mathsf{pk}, \mathsf{c}_{\mathsf{i}+1}, *)$, which either corresponds to $(\mathsf{pk}, \mathsf{c}_{\mathsf{i}+1}, \Lambda_{\mathsf{sk}}(\mathsf{c}_{\mathsf{i}+1}))$ or to $(\mathsf{pk}, \mathsf{c}_{\mathsf{i}+1}, \mathsf{k}_{\mathsf{i}+1})$, we simply form

$$((\mathsf{c}_1, \mathsf{Pub}_{\mathsf{pk}}(\mathsf{c}_1, \mathsf{w}_1)), \ldots, (\mathsf{c}_\mathsf{i}, \mathsf{Pub}_{\mathsf{pk}}(\mathsf{c}_\mathsf{i}, \mathsf{w}_i)), (\mathsf{c}_{\mathsf{i}+1}, *)\,(\mathsf{c}_{\mathsf{i}+1}, \mathsf{k}_{\mathsf{i}+2})), \ldots, (\mathsf{c}_\mathsf{p}, \mathsf{k}_\mathsf{p}))\,, \tag{14}$$

where w_j, for $1 \le j \le i$, is a witness for c_j. □

A.6 From subgroup indistinguishability to homomorphic weak$_1$ PRFs

We first start by reviewing the *quadratic residuosity* assumption. For an *RSA number* N (i.e., $N = pq$, where p and q are distinct odd primes) we use $\mathcal{QR}_N$ to denote the subset of $\mathbb{Z}^*_\mathbb{N}$ consisting of quadratic residues modulo N, and let $\mathcal{J}_N$ denote the set of elements in $\mathbb{Z}^*_\mathbb{N}$ with Jacobi symbol, $\left(\frac{\cdot}{N}\right)$, one. Finally, we define $\mathcal{QNR}_N = \mathcal{J}_N \setminus \mathcal{QR}_N$.

Assume that $\mathsf{RSAGen}(1^\lambda)$ is a PPT algorithm that on input 1^λ generates a *Blum integer* N, i.e., $N = pq$ with p and q being distinct primes satisfying $p, q \equiv 3 \pmod 4$. We stress here that we do not need $\mathsf{RSAGen}(1^\lambda)$ to output the factorization of N as well. We say that the quadratic residuosity (QR) problem is hard under RSAGen if $\{N, U(\mathcal{QR}_N)\}_{\lambda\in\mathbb{N}}$ is computationally indistinguishable from $\{N, U(\mathcal{QNR}_N)\}_{\lambda\in\mathbb{N}}$, where N is generated according to $RSAGen(1^\lambda)$.

A weak$_1$ PRF based on SG. We refer the reader to [9] for the definition of subgroup indistinguishability assumptions. We use the same notation as in [9] and refer the reader to [9] for a detailed definition.
.

Theorem 7. *Assuming the subgroup indistinguishability assumption holds there exists a weak$_1$ PRF (F, f^{hid}) that is both invertible and homomorphic.*

Proof. We build PRF (F, f^{hid}) as follows. Run the subgroup indistinguishability scheme to obtain $(G_U, G_{\mathsf{M}}, G_L, h, T)$ and let the generated tuple be the public parameter of the PRF. (Here G_U is a description of the direct product of G_{M} and G_L.) We let $K_\lambda = \mathbb{Z}_{T^2}$, $D_\lambda = R_\lambda = G_L$, $M_\lambda = G_{\mathsf{M}}$ and $R'_\lambda = G_U$. We define $f_k(d) = d^k$ and $f^{\text{hid}}(d', m) = m \cdot d'$. Invertibility and homomorphism of (F, f^{hid}) follow immediately.[8] Finally, for weak$_1$ security we need to show that $(d_1, \ldots, d_p, d_1^k, \ldots, d_p^k, d, m_1 \cdot d^k) \equiv^c (d_1, \ldots, d_p, d_1^k, \ldots, d_p^k, d, m_2 \cdot d^k)$, where $d_1, \ldots, d_p, d \leftarrow G_L$, $k \leftarrow \mathbb{Z}_{T^2}$ and $m_1, m_2 \in G_{\mathsf{M}}$; The proof of this follows immediately from [9, Lemma B.1]. □

[8] To do inversion we need to recover m from d' and the description of (m, d'). This can be guaranteed by, e.g., assuming that (m, id), i.e., the element of G_U that is the description of (m, id), with id being the identity of G_L, reveals m. Note that we can compute (m, id) from d' and (m, d').