Reproducible Circularly-Secure Bit Encryption: Applications and Realizations

Mohammad Hajiabadi, Bruce M. Kapron

Dept. of Computer Science, University of Victoria, Victoria, Canada {mhaji,bmkapron@cs.uvic.ca}

Abstract. We give generic constructions of several fundamental cryptographic primitives based on a new encryption primitive that combines *circular security* for bit encryption with the so-called *reproducibility property* (Bellare et al. PKC 2003). At the heart of our constructions is a novel technique which gives a way of de-randomizing reproducible public-key bit-encryption schemes and also a way of reducing one-wayness conditions of a constructed trapdoor-function family (TDF) to circular security of the base scheme. The main primitives that we build from our encryption primitive include *k-wise one-way* TDFs (Rosen and Segev TCC 2009), CCA2-secure encryption and deterministic encryption. Our results demonstrate a new set of applications of circularly-secure encryption beyond fully-homomorphic encryption and symbolic soundness. Finally, we show the plausibility of our assumptions by showing that the DDH-based circularly-secure scheme of Boneh et al. (Crypto 2008) and the subgroup indistinguishability based scheme of Brakerski and Goldwasser (Crypto 2010) are both reproducible.

Keywords: Circular security, correlated-input security, trapdoor functions, (non-)shielding CCA construction, deterministic encryption

1 Introduction

A central problem in cryptography is delineating the assumptions required for the existence of cryptographic primitives. One way to differentiate assumptions is by whether they refer to the hardness of a *specific* computational problem (e.g., factoring), or refer *generically* to a class of problems (e.g., inverting efficiently computable functions). Assumptions of the former sort often lead to primitives which are more practical, e.g., in terms of efficiency or levels of security achieved. Those of the latter sort are useful for gaining deeper insights into the security requirements of a primitive, and also as a means of unifying specific assumptions. However, these approaches are not mutually exclusive. In particular, in cases where we have not been able to obtain constructions based on generic assumptions, we may consider strengthening an assumption with some more specific properties. This is the approach we take in this paper. By adding a syntactic property to *circularly-secure* bit encryption, we are able to obtain constructions of several powerful cryptographic primitives.

More precisely, we give constructions of various cryptographic primitives based on a general encryption primitive, which combines *circular security* with a property called *reproducibility* [5], which, the latter, gives a way of functionally reusing randomness across independent public keys. We show the following results.

(1) We give a novel generic construction of TDFs from reproducible bit encryption, and under this construction we show that successively stronger circular-security conditions result in successively stronger one-wayness conditions: we give a hierarchy of circular security notions, called *k-rec circular security*, all of which are weaker than those of [11,12,2], and show if the base scheme is k-rec circularly secure, the constructed TDF is k-wise one-way, in the sense of [31].
(2) We show how to extract many hardcore bits for our constructed TDFs, and by applying the results of [31] we obtain a blackbox (BB) construction of CCA2-secure encryption from our assumptions. Our CCA2 construction is *non-shielding* in the sense of [19]. We partially justify this fact

by showing wrt a weaker encryption primitive than ours, a non-shielding BB CCA2 construction is possible, while a shielding CCA2 construction is BB impossible.
(3) By slightly extending our base primitive, we show how to obtain deterministic encryption schemes secure under *block-source* inputs, as defined by [9].
(4) We realize our base encryption primitive by showing the circularly-secure schemes of [11,12] are reproducible.

In what follows, we provide some background, give a more detailed exposition of our results and describe our constructions and proof techniques. First of all, we assume the following notation and conventions throughout the introduction. Unless otherwise stated, an encryption scheme is bit encryption with randomness space $\{0,1\}^\rho$ and secret-key space $\{0,1\}^l$, where $l = l(n)$ and $\rho = \rho(n)$; by $E_{pk}(m)$, for $m \in \{0,1\}^*$, we mean bitwise encryption of m.

Trapdoor functions Central to public-key cryptography is the notion of *injective trapdoor one-way function*, which refers to a family of functions, where each function in the family is easy to compute, but a randomly chosen function is hard to invert without a *trapdoor key*. A related notion is *witness-recovering CPA-secure encryption*: CPA-secure public-key encryption (PKE) where the decryption algorithm also recovers the randomness used for encryption. It is well-known that these two primitives are equivalent. However, as shown by Gertner et al. [20], there is a BB separation between CPA-secure PKE and TDFs. An interpretation of this result is that a construction of a TDF from PKE should either be non-blackbox, or should rely on specific properties of the PKE. Indeed, under specific assumptions, TDFs may be constructed "directly" (e.g., under the factoring assumption), or may be constructed by using the specifics of a particular PKE scheme (e.g., the strong homomorphisms, among other properties, of ElGamal encryption [29]).

A folklore attempt to build a TDF from PKE is to encrypt a message x under a randomness string derived deterministically from x. However, by [20], such a methodology is in general not sound. A naturally arising question is what properties of PKE enable sound realizations of this approach. The starting point of our work is a related question, namely: when does a PKE scheme allow "secure" encryption of r, using r itself as randomness? By security we mean it be hard to recover r from $(E_{pk_1}(r_1; r), \ldots, E_{pk_\rho}(r_\rho; r))$. Note that this immediately yields a TDF.

To address this question we first review a property of PKE schemes, called *reproducibility* [5]: $\mathcal{E} = (Gen, E, D)$ is reproducible if there exists an efficient deterministic function R, which given a ciphertext $c = E_{pk}(m; r)$, a message m_1, and public/secret keys (pk_1, sk_1), computes $E_{pk_1}(m_1; r)$, which we denote by $R(c, m_1, sk_1)$. Namely, there is an efficient way to transfer the randomness underlying a given encryption to another, provided the secret key for the second encryption is known. Although this notion may seem overly strong, natural cryptosystems (e.g., ElGamal, hash-proof-system-based cryptosystems) do satisfy this property. Indeed, under ElGamal a group element q is encrypted as $(g^r, g^{r \cdot sk} \cdot q)$, allowing the (encoded) randomness g^r be reused under a new secret key. Let $\mathcal{E} = (Gen, E, D, R)$ be a reproducible PKE scheme. Define $\mathcal{E}' = (Gen', E', D')$ as follows: $(pk', sk') \leftarrow Gen'$, where $sk' = r$ and $pk' = c = E_{pk}(0; r)$ (i.e., the secret key is a randomness string r and the public key is a dummy ciphertext formed under r); $E'_c(b)$ samples $(pk_1, sk_1) \leftarrow Gen$, computes $c' = R(c, b, sk_1)$ and returns (pk_1, c') (i.e., E'_c encrypts b by reusing the randomness underlying c); and $D'_r(pk_1, c')$ returns the bit b that $E_{pk_1}(b; r) = c'$. Intuitively, CPA security of $\mathcal{E}'$ follows from reproducibility and CPA security of $\mathcal{E}$. Moreover, the construction swaps the key and randomness spaces of $\mathcal{E}$, and so the task of securely encrypting randomness in $\mathcal{E}'$ reduces to that of securely self-encrypting the secret key in $\mathcal{E}$; this latter is the problem of *circular security*, a special case of the well-studied problem of *key-dependent-message* security [8,11,3,2,12,25,1,13].

The discussion above suggests a general technique for de-randomizing reproducible bit-encryption schemes, sketched below, which is the basis for all our subsequent constructions.
For $\mathcal{E} = (Gen, E, D, R)$ define $\mathcal{F} = C(\mathcal{E}) = (G, F, F^{-1})$ as follows. The domain space of F is the set of all pairs of public/secret keys generated under $Gen(1^n)$.

- G: To produce index/trapdoor keys (ik, tk), generate $(pk, sk) \leftarrow Gen(1^n)$, set

$$ik = (pk, E_{pk}(0; r_1), \ldots, E_{pk}(0; r_l)),$$

  for random r_i's, and set $tk = (r_1, \ldots, r_l)$.
- $F(\cdot, \cdot)$: On key $ik = (pk, c_1, \ldots, c_l)$ and domain input (pk', sk'), return $(pk', c'_1, \ldots, c'_l)$, where $c'_i = R(c_i, sk'_i, sk')$. (Here, sk'_i denotes the ith bit of sk'.)
- $F^{-1}(\cdot, \cdot)$: given trapdoor key $tk = (r_1, \ldots, r_l)$ and image point $(pk', c'_1, \ldots, c'_l)$, form the output as $(pk', b_1 \ldots b_l)$, where b_i is the bit which satisfies $c'_i = E_{pk'}(b_i; r_i)$.

Correctness of $\mathcal{F}$ follows by the reproduction property of R. Also, since R is deterministic, so is the evaluation algorithm F. Finally, we take advantage of the fact that $\mathcal{E}$ is bit encryption to ensure efficient inversion for $\mathcal{F}$.

To discuss one-wayness we need the following definitions. For (pk, sk) output by Gen we refer to $E_{pk}(sk)$ as an *sk-self-encryption*. We call $\mathcal{E}$ *k-rec circularly secure* if no adversary can recover (with a nonnegligible chance) a random sk from k independent sk-self-encryptions, and call $\mathcal{E}$ *k-ind circularly secure* if no adversary can distinguish between k independent sk-self-encryptions and encryptions of, say, zero. The notion of circular security in the literature is that of k-ind circular security, for unbounded k. For the construction above we show the following *tight* reduction.

Theorem 1. If $\mathcal{E}$ is reproducible and 1-rec circularly secure then $C(\mathcal{E})$ is one-way.

The reduction above is "security preserving" in the following sense: assuming $\mathcal{E}$ is reproducible, then $\mathcal{E}$ is 1-rec circularly secure iff $C(\mathcal{E})$ is one-way. Indeed, as we show next, by strengthening the condition of 1-rec circular security we achieve stronger forms of one-wayness.

A family of TDFs is called *k-wise one-way* [31] if one-wayness holds even if the given input is evaluated under k independently chosen functions.[1] More formally, $\mathcal{F} = (G, F, F^{-1})$ is called k-wise one-way, if $\mathcal{F}$'s *k-wise product*, defined as $F_{ik_1,\ldots,ik_k}(x) = (F_{ik_1}(x), \ldots, F_{ik_k}(x))$ is one-way. Rosen and Segev [31] showed the utility of this notion by giving a blackbox construction of CCA2-secure encryption based on k-wise one-way TDFs, for a sufficiently large k, simplifying a prior construction [29] based on lossy TDFs (LTDFs). Despite their utility, k-wise one-way TDFs (even for $k = 2$) are very strong primitives, whose only generic constructions so far have been based on LTDFs. Indeed, as shown by Vahlis [33], even 2-wise one-way TDFs cannot be constructed in a blackbox way from trapdoor permutations (TDPs).

Our TDF construction provides an easy means for obtaining k-wise one-way TDFs: we can generalize Theorem 1 to show the following

If $\mathcal{E}$ is reproducible and k-rec circularly secure then $C(\mathcal{E})$ is k-wise one-way.

To put our construction of k-wise one-way TDFs in context, we compare it to the LTDF-based construction [31]: the security reduction of [31] involves both statistical and computational arguments, allowing one to obtain only k-wise one-way TDFs for a priori fixed but arbitrarily large values of k (which does suffice for CCA2 encryption) from sufficiently lossy TDFs. Our reduction

[1] Actually, [31] chose another name for this particular notion, but we refer to it as k-wise one-wayness for simplicity.

argument, on the other hand, is entirely computational, allowing us to obtain unbounded k-wise one-way TDFs (i.e., a TDF that is k-wise one-way for any value of k) from the full circular security assumption.

As for the base assumptions, the relationships among the circular-security notions we described is not well-understood (beyond the trivial ones). Under certain assumptions these notions become equivalent. For example, any *re-randomizable* 1-rec circularly-secure scheme is poly-ind circularly secure: this follows by considering that a 1-rec circularly-secure scheme is already poly-rec circularly secure (because of re-randomizability), and that any poly-rec circularly-secure scheme is also poly-ind circularly secure [32, Theorem 8]. For the rest of the introduction, however, for ease of exposition, we describe the results wrt full circular security.

We extend Construction C for the case in which the base scheme is t-circularly secure (i.e., circularly-secure wrt t keys): the input of each TDF is t pairs of public/secret keys, the index key contains $l \cdot t$ dummy ciphertexts, and the evaluation algorithm on $(pk_0, sk_0, \ldots, pk_{t-1}, sk_{t-1})$ returns $(pk_0, \ldots, pk_{t-1})$ along with $t \cdot l$ ciphertexts formed by encrypting each bit of sk_i under $pk_{(i+1 \mod t)}$ (deterministically) by reusing the randomness of the corresponding ciphertext of the index key.

Extracting hardcore bits. Given the TDFs built above, we may apply the general Goldreich-Levin (GL) theorem [21] to extract a hardcore bit. We would like to, however, avoid the use of the GL theorem for several reasons. First, the GL reduction, due to its generality, is not tight, while we would like to achieve CCA security with tight reductions. Second, for our deterministic encryption results we need to be able to extract many hardcore bits. Finally, since our base assumptions are strictly BB-stronger (by Vahlis's result) than one-way TDFs, we should look for more specialized methods. We sketch below two deterministic methods for extracting many hardcore bits with tight security reductions for our constructed TDFs. The first method applies to t-circular security and allows us to extract $\log((t-1)!)$ bits, with the advantage that it only increases the domain size. The second method allows us to extract any, a priori fixed, number of bits, but it enlarges other spaces as well.

First method: a cycle hides its ordering. For simplicity, we describe the idea for 3-circular security, showing how to extract a single hardcore bit. The idea is 3-circularly security implies no adversary can distinguish between the sequences $(pk_1, pk_2, pk_3, E_{pk_1}(sk_2), E_{pk_2}(sk_3), E_{pk_3}(sk_1))$ and $(pk_1, pk_2, pk_3, E_{pk_1}(sk_3), E_{pk_2}(sk_1), E_{pk_3}(sk_2))$. Now we augment our TDF construction described above (for the t-circular security case), so that the evaluation algorithm, besides the input $(pk_1, sk_1), (pk_2, sk_2), (pk_3, sk_3)$, also receives an additional bit b, used to dictate the ordering used to form the cycle. The inversion algorithm can open the ciphertexts, as before, and recover the bit b, by checking, say, whether the key encrypted under pk_1 is a secret key for pk_2 or for pk_3.[2] This technique extends to the t-circular security case for any $t > 3$, allowing us to "hide" a random ordering, providing $\log((t-1)!)$ hardcore bits.

Second method. We describe the idea for 1-circular security. We extend construction C above to be parameterized over an integer $m = m(n)$ and to result in a TDF whose input now consists of triples (pk, sk, x), as opposed to (pk, sk) alone, where $x \in \{0,1\}^m$. Moreover, we augment the index key to contain m added ciphertexts and let the trapdoor key contain their underlying randomness strings. Now $F(ik, (pk, sk, x))$ proceeds as before, but it also "encrypts" x in the process by again reusing randomness. For this TDF, we show that x remain pseudorandom even knowing $F(ik, (pk, sk, x))$. Finally, assuming the property that public keys under the base scheme are com-

[2] This, however, imposes a negligible inversion error.

puted deterministically from their secret keys (plus perhaps some public parameters), we show how to obtain TDFs that hide a $(1 - o(1))$ fraction of their input bits.

CCA-secure encryption. Using results on k-wise one-way TDFs with many hardcore bits,[3] we may now use the BB construction of Rosen and Segev [31] to build a many-bit CCA2-secure PKE from a reproducible, circularly secure bit-encryption scheme. Specifically, [31] gives a BB construction of CCA2-secure encryption from k-wise one-way TDFs, for $k \in \Omega(n)$; they also show that $k \in \omega(\log n)$ suffices for CCA1 encryption. Our CCA constructions, by relying on that of [31], result in schemes whose decryption functions query the encryption function of the base scheme. Gertner, Malkin and Myers [19] refer to such constructions as *non-shielding*, and show that there exists no *shielding* BB construction of CCA-secure from CPA-secure encryption. Since our base assumptions are BB-stronger than CPA security, it is natural to ask whether the non-shielding nature of our CCA2 construction is just an artifact of the construction of [31], or whether it is inherent. We were not able to answer this question for our encryption primitive, mainly because of the presence of the reproduction function. However, we are able to answer this wrt a weaker primitive than ours, which is a special case of *randomness-dependent-message-secure (RDMS)* encryption [7], which allows secure multiple bitwise-encryptions of a randomness string r under r itself as randomness (Formalized in Definition 2). Calling this new primitive RDMS encryption, we show that RDMS encryption is implied by our base assumptions, and also that it enables a non-shielding construction of CCA-secure encryption. We prove the latter by directly constructing k-wise one-way TDFs using RDMS encryption. Next we observe that the shielding-BB impossibility result of [19] extends even if the base scheme is an RDMS encryption primitive (Theorem 6). Indeed, it seems that this latter statement is true for most encryption primitives whose security requirements are defined wrt passive indistinguishability (i.e., no decryption oracles); see Section 4 for more details. Thus, we obtain an encryption primitive, wrt which a non-shielding BB CCA-secure construction is possible, but under which a shielding CCA-secure construction is BB impossible.

Deterministic encryption (DE) Following [9], a deterministic l-bit-encryption scheme is called (λ, l)-IND secure if encryptions of any two (efficient) λ-sources (i.e., distributions with min-entropy λ) result in computationally indistinguishable ciphertexts. We formulate two extended notions of circular security, called (λ, l)-*entropy circular security* and *strong-*(λ, l)-*entropy circular security*, both of which require circular security hold even if the secret key $sk \in \{0,1\}^l$ is sampled from a λ-source distribution, while the strong-entropy version requires one more assumption, related to the public-key distribution.[4]

We show our TDF construction immediately gives us a (λ, l)-IND-secure DE scheme if the base scheme satisfies strong (λ, l)-entropy circular security. We also show, by appropriately choosing the parameters, the schemes of [11,12] provide strong-entropy circular security, meaning our generic transformation applies to these two schemes to obtain secure DE schemes, which explains the striking similarities between (especially) the DDH-based DE scheme of [9] and the scheme of Boneh et al. [11]. We also note that the extra condition of strong-entropy circular security may be satisfied if, informally, the key-generation algorithm acts as a *strong extractor*, producing the public key from the secret key (taken as the source) based on a public parameter (taken as the seed). Similar structural assumptions are made in other settings, e.g., [35], to obtain DE schemes.

[3] We note that our hardcore-security results hold not only for $\mathcal{F} = C(\mathcal{E})$, but also for $\mathcal{F}$'s k-wise products, under the corresponding assumptions. See Section 3.2.

[4] The notion of weak-entropy circular security was also considered by [13] in the context of KDM amplification.

For weak-entropy circular security we also show how to obtain a secure DE scheme but with looser parameters, i.e., the (λ, l)-parameters of the base scheme are not maintained. We follow the so-called *encrypt-with-hardcore* technique, implicitly used in [6,4,9], and formalized in [18]. A high-level description of the idea is as follows. Assume $\mathcal{F} = (G, F, F^{-1})$ is a TDF with an associated hardcore function h producing $\Omega(n)$ hardcore bits, and we want to make $\mathcal{F}$ a secure DE scheme. Suppose we have the bonus that h preserves hardcore security even if x is sampled from a biased, high min-entropy distribution. Now we can build a DE scheme by encrypting the output of F using the hardcore bitstring under a randomized-encryption scheme $\mathcal{E}'$: namely, $E_{ik,pk}(x) = E'_{pk}(F(ik, x), h(x))$; decryption can be done using ik's trapdoor key and pk's secret key. Security of E comes from the fact that $(F(ik, x), h(x))$ is computationally indistinguishable from $(F(ik, x), r)$, so $h(x)$ is as good as a fresh randomness string. The only remaining issue is that E may require a longer randomness string, which, however, can be handled by applying a pseudorandom generator to $h(x)$.

Further discussion. Since LTDFs [29] are the only generic assumption (to the best of our knowledge) that imply k-wise one-way TDFs, it is natural to ask about the relationship between LTDFs and our base primitive. We believe these notions are incomparable. First, under our encryption primitive, we are able to obtain a TDF that is k-wise one-way for unbounded k's; LTDFs are known to achieve bounded k-wise one-way TDFs, but this does not seem to generalize to the unbounded case, mainly due to the nature of LTDF-based proof techniques that also rely on statistical arguments. On the other hand, LTDFs have powerful statistical properties (i.e., losing information in lossy mode) which do not seem to be realizable under our assumptions. Choi and Wee [15], by abstracting the DDH-based TDF construction of [29], show how to obtain LTDFs from reproducible encryption that is homomorphic wrt both messages and randomness. For similar reasons our assumptions seem incomparable to those of [29]. Interestingly, all constructions of circularly-secure schemes in the literature rely on homomorphic properties of their underlying assumptions [11,12,2]. A circularly-secure scheme, however, as a general primitive, does not necessarily provide such homomorphic structures. For example, under standard assumptions one may construct a CCA2-secure, circularly-secure scheme [14,22], while such homomorphic structures would violate the CCA2-security of the same scheme.

We note that almost all BB CCA2-constructions, based on encryption or TDFs, are non-shielding [26,31,29], except for a few cases which rely on very powerful (and structurally different) primitives, e.g., [10]. Intuitively, the non-shielding property of those constructions is used to do consistency checks on ciphertexts. It would be interesting to explore if there exist weaker encryption primitives (than those we consider) for which the BB separation of [19] is the best possible.

Our results show an alternative way (to those presented in [29,17]) of constructing DDH-based TDFs. Right now, by instantiating our TDF construction under the DDH-based circularly-secure scheme [11], we obtain no improvement in efficiency over existing constructions. This motivates the search for more efficient DDH-based circularly-secure schemes.

Finally, we discuss adaptations of Construction $C(\mathcal{E})$ to the case in which the secret-key space of $\mathcal{E}$ is a subset of the plaintext space $\mathcal{M}$ (which allows the secret key to be encrypted as a whole) and reproducibility holds wrt $\mathcal{M}$. For this case we may substantially improve efficiency by having each index key contain only one ciphertext, whose randomness will be reused to self-encrypt the secret key (as a whole) given as input to the evaluation algorithm. To perform inversion, however, we would need to rely on one more assumption: it is efficiently possible to recover m from pk, $E_{pk}(m; r)$ and r, for all pk, m and r. This last property by itself is satisfied by natural cryptosystems,

e.g., ElGamal. Moreover, there is a standard (and straightforward) way to make any CPA-secure scheme (for which $\{0,1\}^l \subseteq \mathcal{M}$) "one-shot" circularly secure; this transformation, however, does not (necessarily) maintain this last property. Thus, our results suggest that the CPA-to-one-shot-circular transformation may be non-trivial (and interesting) if it is to maintain the last property.

2 Basic notation and definitions

Notation. For a finite set S we use $x \leftarrow S$ to denote sampling x uniformly at random from S and denote by U_S the uniform distribution on S. If D is a distribution then $x \leftarrow D$ denotes choosing x according to D. We use the word PPT in this paper in the standard sense. We use $A(\ldots;r)$ to denote the deterministic output of PPT function A when the randomness is fixed to r, and use $x \leftarrow A(a_1, a_2, \ldots)$ to denote the distribution formed by outputting $A(a_1, a_2, \ldots; r)$ for a uniformly-random r. If $A(x_1, \ldots, x_m; r)$ outputs a tuple of strings, we let $A_i(x_1, \ldots, x_m)$ be the distribution formed by outputting the ith component of $A(x_1, \ldots, x_m)$. We denote the support set of a distribution D by $Sup(D)$, and write $x \in D$ to indicate $x \in Sup(D)$. We call $f : \mathbb{N} \to \mathbb{R}$ negligible if $f(n) < 1/P(n)$, for any polynomial P and sufficiently large n. We write $negl$ to denote unspecified negligible functions. We denote by f^{-1} the inverse of an injective function f. 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 function D, we have $|\Pr[D(X_n) = 1] - \Pr[D(Y_n) = 1]| = negl(n)$. We write $X \equiv Y$ to mean X and Y are identically distributed. All functions, adversaries, distributions, etc., that appear in this paper, if not otherwise stated, are assumed to be efficiently computable/samplable. For $x, y \in \{0,1\}^*$ we use $|x|$ to denote the bit length of x , use x_i, for $1 \le i \le |x|$, to denote the ith bit of x, and use $x||y$ to denote the concatenation of x and y.

Trapdoor functions We first start by giving the standard definitions related to trapdoor functions and hardcore bits.

A collection, $\mathcal{F} = (G, F)$, of functions is defined as follows. The algorithm $G(1^n)$ returns a function index s, and the deterministic algorithm $F(s, \cdot)$ computes a function $f_s : D_n \to R_n$. We stress both the domain and range of f_s only depend on the security parameters, 1^n. We call $\{D_n\}$ the domain space of $\mathcal{F}$. Assuming that $\mathcal{D} = \{\mathcal{D}_n\}$ is a distribution over $\{D_n\}$ and $h : D_n \to \{0,1\}^{p(n)}$ is a deterministic function, we define the following notions. We say $\mathcal{F}$ is $\mathcal{D}$-*one-way* if for any adversary $\mathcal{A}$, $\Pr[f_s(\mathcal{A}(s, f_s(x))) = f_s(x)] = negl(n)$, where the probability is computed over $s \leftarrow G(1^n)$, $x \leftarrow \mathcal{D}_n$ and $\mathcal{A}$'s coins.

We say that h is a $\mathcal{D}$-*hardcore function* for $\mathcal{F}$ if for any adversary $\mathcal{A}$,

$$\left|\Pr[\mathcal{A}(s, f_s(x), h(x)) = 1] - \Pr\left[\mathcal{A}(s, f_s(x), U_{\{0,1\}^{p(n)}}) = 1\right]\right| = negl(n),$$

where $s \leftarrow G(1^n)$ and $x \leftarrow \mathcal{D}_n$. We often omit $\mathcal{D}$, from $\mathcal{D}$-hardcore, etc., when it is clear from context. Next, we define TDFs and their k-wise products, following the notation used in [31].

A *collection of injective trapdoor functions* (TDFs)[5] is given by three algorithms $\mathcal{F} = (G, F, F^{-1})$, where $G(1^n)$ randomly produces a pair (ik, tk) of *index/trapdoor keys*, the deterministic algorithm $F(ik, \cdot)$ computes an injective function $f_{ik} \colon D_n \to R_n$, and $F^{-1}(tk, \cdot)$ computes $f_{ik}^{-1}()$—We may sometimes allow $F^{-1}(tk, \cdot)$ to err with a negligible probability, computed over $(ik, tk) \leftarrow G(1^n)$ and the random input to $F(ik, \cdot)$. We stress that the input domain of f_{ik} only depends on the security parameter 1^n. The k-wise product of $\mathcal{F}$, denoted $\mathcal{F}^{(k)} = \left(G^{(k)}, F^{(k)}\right)$, is defined as follows. The

[5] We use TDF to refer to a collection of injective trapdoor functions henceforth.

algorithm $G^{(k)}(1^n)$ runs $G(1^n)$ independently k times to output k index keys, $(ik_1, \ldots, ik_k)$; on input x, $F^{(k)}((ik_1, \ldots, ik_k), \cdot)$ returns $(F(ik_1, x), \ldots, F(ik_k, x))$.

Assume $\mathcal{F}$ is a TDF with domain $D = \{D_n\}$ and $\mathcal{D} = \{\mathcal{D}_n\}$ is a distribution on D. We say $\mathcal{F}$ is *k-wise $\mathcal{D}$-one-way* if $\mathcal{F}^{(k)}$ is $\mathcal{D}$-one-way.

Bit encryption schemes All encryption schemes that appear throughout, if not explicitly stated, are *bit-encryption* schemes. In our applications we need to work with a more general notion of encryption schemes involving *public parameters*. A bit-encryption scheme $\mathcal{E} = (Param, Gen, E, Dec)$ is defined as follows. $Param$ on input 1^n outputs a random parameter, par. The *key-generation algorithm*, Gen on inputs 1^n and par generates a public/secret key $(pk, sk) \leftarrow Gen(1^n, par)$; we assume pk includes par, so we do not include par as input to other algorithms. The *encryption algorithm*, E, on inputs 1^n, pk, bit b and randomness $r \in \mathcal{R}_n$, outputs ciphertext $c = E_{pk}(b; r)$. The *decryption algorithm*, Dec, takes a secret key sk and ciphertext c, and deterministically outputs a bit $b = Dec_{sk}(c)$. For correctness, we require $\Pr[Dec_{sk}(E_{pk}(b)) = b] = 1$, for $par \leftarrow Param(1^n)$, $(pk, sk) \leftarrow Gen(1^n, par)$ and $b \leftarrow \{0,1\}$. We will typically use $\mathcal{R}_n$ to denote the underlying encryption-randomness space of a scheme under consideration. We assume the following: for any fixed par, all secret keys output by $Gen(1^n)$ are bitstrings of the same length, and, whenever we are generating many public keys, all keys are generated wrt a single initial par. Thus, we make $Param$ implicit henceforth.

We say $\mathcal{E} = (Gen, E, Dec)$ is *CPA secure* if $(pk, E_{pk}(0)) \equiv^c (pk, E_{pk}(1))$, where pk is chosen according to $Gen(1^n)$. For $m \in \{0,1\}^*$, we extend E to define $E_{pk}(m) = (E_{pk}(m_1), \ldots, E_{pk}(m_{|m|}))$. If $\mathbf{r} = (r_1, \ldots, r_t)$ and $m \in \{0,1\}^t$ we write $E_{pk}(m; \mathbf{r}) = (E_{pk}(m_1; r_1), \ldots, E_{pk}(m_t; r_t))$.
We now give definitions for circular security. We say $\mathcal{E} = (Gen, E, Dec)$ is *k-rec t-circularly secure* if for every adversary $\mathcal{A}$, it holds

$$\Pr[\mathcal{A}(pk_1, \ldots, pk_t, \mathbf{c}_1, \ldots, \mathbf{c}_k) = sk_1] = negl(n),$$

where $(pk_1, sk_1), \ldots, (pk_t, sk_t) \leftarrow Gen(1^n)$ and for every $1 \leq i \leq k$

$$\mathbf{c}_i \leftarrow (E_{pk_2}(sk_1), E_{pk_3}(sk_2), \ldots, E_{pk_1}(sk_t)).$$

We say $\mathcal{E}$ is *k-ind t-circularly secure* if $\mathcal{E}$ is CPA secure and also it holds that

$$(\mathbf{c}_1, \ldots, \mathbf{c}_k) \equiv^c (\mathbf{c}'_1, \ldots, \mathbf{c}'_k),$$

where $\mathbf{c}_i$'s are generated as above and

$$\mathbf{c}'_i \leftarrow \left(E_{pk_2}(0^l), E_{pk_3}(0^l), \ldots, E_{pk_t}(0^l), E_{pk_1}(0^l)\right),$$

for every $1 \leq i \leq k$, and $l = |sk_1|$. Note that we add CPA security as a separate condition because otherwise the definition may be satisfied trivially, e.g., consider the encryption scheme under which the secret key is always the all-zero string and the encryption function is the identity function.

Henceforth, when we say k-rec circular security (or k-ind circular security) we are referring to the definition wrt a single pair of public/secret keys.

Definition 1. *We call $\mathcal{E} = (Gen, E, Dec)$* reproducible *if there exists a deterministic function R, called the* reproduction function*, s.t. for any (pk_1, sk_1), $(pk_2, sk_2) \in Gen(1^n)$, $r \in \mathcal{R}_n$ and $b_1, b_2 \in \{0,1\}$,*

$$R(pk_1, E_{pk_1}(b_1; r), b_2, pk_2, sk_2) = E_{pk_2}(b_2; r).$$

For simplicity we omit the inclusion of pk_1 and pk_2 as inputs to R.

3 Constructing TDFs and hardcore bits

In this section we give the constructions for TDFs and hardcore bits.

3.1 TDFs from reproducible encryption

We begin by giving a construction that takes as input a reproducible bit-encryption scheme and produces a TDF. We then show how to achieve increasingly stronger guarantees of one-wayness for the constructed TDF from corresponding assumptions about the base encryption primitive. We present the construction adapted to the t-circular security case (i.e., circular security wrt t keys), meaning that we will obtain guarantees of one-wayness for the constructed TDF from t-circular security assumptions.

We use D^t to denote the t'th Cartesian power of a set D. If $\mathcal{D}$ is a distribution, $\mathcal{D}^t$ denotes the t-tuple formed by sampling t times independently from $\mathcal{D}$.

Construction 1 *Construction C_1 takes in a reproducible bit-encryption scheme $\mathcal{E} = (Gen, E, Dec, R)$ and $t = t(n)$ and it outputs a TDF, $\mathcal{F} = (G, F, F^{-1})$, with domain space D^t, where $D = Sup(Gen(1^n))$. Let $l = l(n)$ be the length of a secret keys output by $Gen(1^n)$.*

- *$G(1^n)$: Let $(pk, sk) \leftarrow Gen(1^n)$, and form $tk = (r_{1,1}, \ldots, r_{1,l}, \ldots, r_{t,1}, \ldots, r_{t,l})$, for independent $r_{i,j}$'s, and $ik = (pk, c_{1,1}, \ldots, c_{1,l}, \ldots, c_{t,1}, \ldots, c_{t,l})$, where for $1 \leq i \leq t$ and $1 \leq j \leq l$, we set $c_{i,j} = E_{pk}(0; r_{i,j})$. Return (ik, tk)*
- *$F\left((pk, c_{1,1}, \ldots, c_{t,l}), (pk_1, sk_1, \ldots, pk_t, sk_t)\right)$: return the tuple $(pk_1, \ldots, pk_t, c'_{1,1}, \ldots, c'_{t,l})$, where for $1 \leq i \leq t-1$ and $1 \leq j \leq l$ we set $c'_{i,j} = R(c_{i,j}, b_{i,j}, sk_{i+1})$, and $c'_{t,j} = R(c_{t,j}, b_{t,j}, sk_1)$, with $b_{i,j}$ being the jth bit of sk_i.*
- *$F^{-1}((r_{1,1}, \ldots, r_{t,l}), (pk_1, \ldots, pk_t, c'_{1,1}, \ldots, c'_{t,l}))$: Retrieve each sk_i, for $1 \leq i \leq t$, bit-by-bit by encrypting back both 0 and 1 with the provided randomness (and under the appropriate public key) and finding the matching bit.*

The TDF's completeness follows by reproducibility. We point out a few remarks. First, the efficiency of the search performed by the inversion algorithm relies on the fact that each ciphertext is hiding a single bit, encrypted under the randomness known to the inverter. Second, our construction is entirely blackbox, also accessing (during evaluation) the reproduction function. Third, our construction extends to the non-bit-encryption case, by still continuing to encrypt the secret key bit-by-bit, but by fixing a mapping from bits to two fixed plaintext messages; for this case, the one-wayness of the constructed TDF reduces to bit-wise circular security of the base scheme (wrt the fixed mapping).

Theorem 1. *Assume $\mathcal{E}$ is a reproducible bit-encryption scheme and $\mathcal{F}$ is the TDF built from $\mathcal{E}$ in Construction 1 based on integer $t = t(n)$. Then, $\mathcal{E}$ is k-rec t-circularly secure if and only if $\mathcal{F}$ is k-wise $\mathcal{D}$-one-way, where $\mathcal{D} = (Gen(1^n))^t$. Moreover, the reductions are tight.*

Proof. We give the proof for the case in which the base encryption scheme is k-rec 1-circularly secure, i.e., wrt a single key. The proof for the general case follows similarly.

We begin by introducing the following notation. For $\mathbf{r} = (r_1, \ldots, r_l)$ and $m \in \{0,1\}^l$, we define $E_{pk}(m; \mathbf{r}) = (E_{pk}(m_1; r_1), \ldots, E_{pk}(m_l; r_l))$.

($\Rightarrow$) Assume that $\mathcal{E}$ is k-rec t-circularly secure and $\mathcal{A}$ is an adversary against the k-wise $\mathcal{D}$-one-wayness of $\mathcal{F}$, achieving advantage ϵ, namely

$$\Pr\left[\mathcal{A}\left(\underbrace{pk', E_{pk'}(sk';\mathbf{r}_1),\ldots,E_{pk'}(sk';\mathbf{r}_k)}_{\textbf{image}},\underbrace{pk, E_{pk}(0^l;\mathbf{r}_1),\ldots,E_{pk}(0^l;\mathbf{r}_k)}_{\textbf{tk}}\right)=(pk',sk')\right]=\epsilon(n),$$

where $(pk, sk), (pk', sk') \leftarrow Gen(1^n)$ and $\mathbf{r}_1, \ldots, \mathbf{r}_k \leftarrow \mathcal{R}_n^l$. We first note that if $\mathcal{A}$ were only given **image**, it could generate **tk** by itself, by sampling $(pk, sk) \leftarrow Gen(1^n)$ independently and appropriately using the reproduction function R. Thus, $\mathcal{A}$'s ability to invert the TDF with advantage ϵ reduces to a new adversary, $\mathcal{B}$, recovering sk' from $\big(pk', E_{pk'}(sk';\mathbf{r}_1),\ldots,E_{pk'}(sk';\mathbf{r}_k)\big)$ with the same advantage, ϵ. This completes the proof for this part.

($\Leftarrow$) This part follows trivially. That is, any adversary that recovers sk' from

$$(pk', E_{pk'}(sk';\mathbf{r}_1),\ldots,E_{pk'}(sk';\mathbf{r}_k))$$

with probability γ can also trivially recover sk' from

$$\Big(pk', E_{pk'}(sk';\mathbf{r}_1),\ldots,E_{pk'}(sk';\mathbf{r}_k), pk, E_{pk}(0^l;\mathbf{r}_1),\ldots,E_{pk}(0^l;\mathbf{r}_k)\Big),$$

with the same probability, by simply discarding the second half of the sequence. □

3.2 Extracting many hardcore bits

We present two deterministic methods for extracting many hardcore bits from the TDF presented in Construction 1, with tight and efficient reductions to the indistinguishability variants of circular security assumptions. The first method applies to t-circular security for $t \geq 2$, allowing us to directly extract $\log((t-1)!)$ bits, by expanding only the domain space by the same number of bits (but without affecting the sizes of the other system's parameters). The second method is less restrictive, allowing us to extract (from t-circular security, for any $t \geq 1$,) $m(n)$ hardcore bits, where m is an arbitrary but a priori fixed poly function, by increasing the domain space by $m(n)$ bits and the image, index-key and trapdoor-key spaces by poly factors of $m(n)$. In particular, by choosing the parameter m appropriately we obtain TDFs that hide a $1 - o(1)$ fraction of their input bits.

First hardcore extraction method. We begin with some notation. Define $[t] = \{1, \ldots, t\}$. Let

$$S = \{f\colon [t] \to [t] \mid f \text{ is injective } \&\ \forall X, \text{ s.t. } \emptyset \subsetneq X \subsetneq [t], \{f(y) \mid y \in X\} \neq X\},$$

for which we have $|S| = (t-1)!$. Intuitively, each $f \in S$ defines a possible circular ordering of encrypting a sequence of t pairs of keys, by having pk_i encrypt $sk_{f(i)}$. The condition $\forall X \subsetneq [t], \{f(y) \mid y \in X\} \neq X$ guarantees that we have a single, full cycle. For example, it is not the case that pk_1 encrypts sk_2, pk_2 encrypts sk_1 and the remaining keys encrypt each other in a circular manner. Fix $\mathcal{O} : \mathbb{Z}_{(t-1)!} \to S$ to be an efficient index function defined using a canonical ordering of the elements of S. We will also write $\mathcal{O}(i, x)$ to denote $f_i(x)$, where f_i is the ith function according to the ordering. We also require that, for any $f \in S$, given $sq = \{(x, f(x)) \mid x \in [t]\}$, it is possible to efficiently compute the index of f according to the ordering[6], which we (by slightly abusing the notation) denote by $\mathcal{O}^{-1}(sq)$. We now proceed to describe the modified TDF construction and the associated hardcore function.

[6] Such an ordering for which we have such a function $\mathcal{O}$ can be defined by fixing an efficient way of enumeration.

Construction 2 *Let $\mathcal{E} = (Gen, E, Dec, R)$, t and D^t be as in Construction 1. The domain space of the TDF, $\mathcal{F} = (G, F, F^{-1})$, we build is now $(D^t, \mathbb{Z}_{(t-1)!})$.*

- $G(1^n)$*: As in Construction 1.*
- $F((pk, c_{1,1}, \ldots, c_{t,l}), (pk_1, sk_1, \ldots, pk_t, sk_t, u))$ *is computed as follows. First, define the indices* $(ind_1, \ldots, ind_t) = (\mathcal{O}(u, 1), \ldots, \mathcal{O}(u, t))$*. Informally, the output will be* $pk_1, \ldots, pk_t$ *together with a chain of encryptions, where* pk_i *encrypts the bits of* sk_{ind_i}*. Return* $(pk_1, \ldots, pk_t, c'_{1,1}, \ldots, c'_{t,l})$*, where, for* $1 \leq i \leq t$ *and* $1 \leq j \leq l$*,* $c'_{i,j} = R(c_{i,j}, b_{i,j}, sk_i)$*, with* $b_{i,j}$ *being the* j*th bit of* sk_{ind_i}*.*
- $F^{-1}((r_{1,1}, \ldots, r_{t,l}), (pk_1, \ldots, pk_t, c'_{1,1}, \ldots, c'_{t,l}))$*: do the following steps:*
  - *for each* $1 \leq i \leq t$*, recover the bitstring,* x_i*, encrypted under* pk_i *bit-by-bit as follows: to retrieve the* j*th bit of* x_i*, encrypt both* 0 *and* 1 *under* pk_i *using randomness* $r_{i,j}$ *and check the result against* $c'_{i,j}$*;*
  - *for each* $1 \leq i \leq t$*, let* ind_i*, where* $1 \leq ind_i \leq t$*, be the index for which it holds that* pk_{ind_i} *is the matching public key of* x_i*,*[7] *and let* $sk_{ind_i} = x_i$*. Form* $sq = \{(1, ind_1), \ldots, (t, ind_t)\}$*; return* $(pk_1, sk_1, \ldots, pk_t, sk_t, \mathcal{O}^{-1}(sq))$*.*

Hardcore function: *For $\mathcal{F}$ given above we define $h\colon (D^t, \mathbb{Z}_{(t-1)!}) \to \mathbb{Z}_{(t-1)!}$ as*

$$h(pk_1, sk_1, \ldots, pk_t, sk_t, u) = u.$$

Correctness of the new TDF follows immediately. Note that Construction 1 is a special case of Construction 2, by forming the encrypted cycle wrt the fixed function $f\colon f(1) = t; f(2) = 1; \ldots, f(t) = t - 1$. In contrast, Construction 2 forms the encrypted cycle according to a random f (provided as input to the TDF), where, as we show below, the random choice of f is what is computationally hidden by the output. We now prove the following theorem.

Theorem 2. *Let $\mathcal{F}$ and h be the TDF and hardcore function constructed according to Construction 2 based on $\mathcal{E} = (Gen, E, Dec, Rep)$ and $t = t(n)$. Assuming $\mathcal{E}$ is k-ind t-circularly-secure, $\mathcal{F}$ is k-wise one-way and h is a hardcore function for $\mathcal{F}^k$.*

Proof. We first start by proving the first statement of the theorem. Assume that for an adversray $\mathcal{A}$ it holds that $\Pr[\mathcal{A}(\mathbf{DS}_1) = sk_1] = \epsilon(n)$, where

$$\mathbf{DS}_1 = E_{pk}(0^l; \mathbf{r}_1), \ldots, E_{pk}(0^l; \mathbf{r}_t), \underbrace{pk_1, \ldots, pk_t, E_{pk_1}(sk_{f(1)}; \mathbf{r}_1), \ldots, E_{pk_t}(sk_{f(t)}; \mathbf{r}_t), (f(1), \ldots, f(t))}_{\textbf{image}}, \quad (1)$$

where $f \leftarrow S$ (the set S was defined above), (pk, sk), $(pk_1, sk_1), \ldots,$ $(pk_t, sk_t) \leftarrow Gen(1^n)$ and $\mathbf{r}_1, \ldots, \mathbf{r}_t \leftarrow \mathcal{R}_n^l$. We show how to construct adversary $\mathcal{B}$, where

$$\Pr[\mathcal{B}(\underbrace{pk_1, \ldots, pk_t, c_1, \ldots, c_t}_{\textbf{circ}})] = \epsilon(n), \quad (2)$$

where pk_i's are sampled as above and $c_i \leftarrow E_{pk_i}(sk_{(i \mod t)+1})$, for $1 \leq i \leq t$. Since $\mathcal{E}$ is reproducible, to construct $\mathcal{B}$ from $\mathcal{A}$, it would suffice to show how to produce **image** from **circ**. We now show the distribution $(pk'_1, \ldots, pk'_t, c'_1, \ldots, c'_t, s_1, \ldots, s_t)$ generated based on **circ** according to the following procedure is identically distributed to **image**:

[7] This can be done by encrypting many bits under the public key and decrypting them under a candidate secret key. This, however, results in a negligible inversion error.

– $s_1 \leftarrow \{1, \ldots, n\}$;
– for $(2 \leq i \leq n)$, sample $s_i \leftarrow (\{1, \ldots, n\} \setminus \{s_1, \ldots, s_{i-1}\})$, where $\setminus$ denotes set difference; and
– for $1 \leq i \leq t$, let $pk'_i = pk_{s_i}$ and $c'_i = c_{s_i}$.

The fact that $\mathbf{image} \equiv (pk'_1, \ldots, pk'_t, c'_1, \ldots, c'_t)$ is proved in a straightforward way, and we omit the details. This concludes the proof of the first part.

For the second statement, we give the proof for $k = 1$ (i.e., 1-ind t-circular security), since the proof for the general case follows in a straightforward way. To argue about hardcore security it suffices to show $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$, for and

$$\mathbf{DS}_2 = E_{pk}(0^l; \mathbf{r}_1), \ldots, E_{pk}(0^l; \mathbf{r}_t), \\ pk_1, \ldots, pk_t, E_{pk_1}(sk_{f(1)}; \mathbf{r}_1), \ldots, E_{pk_t}(sk_{f(t)}; \mathbf{r}_t), (f'(1), \ldots, f'(t)), \quad (3)$$

where $f' \leftarrow S$ and everything else is sampled as in $\mathcal{DS}_1$. From 1-ind t-circular security we can deduce

$$pk_1, \ldots, pk_t, E_{pk_1}(sk_{f(1)}; \mathbf{r}_1), \ldots, E_{pk_t}(sk_{f(t)}; \mathbf{r}_t), (f(1), \ldots, f(t)) \equiv^c \\ pk_1, \ldots, pk_t, E_{pk_1}(0^l; \mathbf{r}_1), \ldots, E_{pk_t}(0^l; \mathbf{r}_t), (f(1), \ldots, f(t)), \quad (4)$$

where all the random variables are sampled as shown above. Now using the reproducibility property, from Equation 4, we obtain that, $\mathbf{DS}_1 \equiv^c \mathbf{DS}_3$, where

$$\mathbf{DS}_3 = E_{pk}(0^l; \mathbf{r}_1), \ldots, E_{pk}(0^l; \mathbf{r}_t), \\ pk_1, \ldots, pk_t, E_{pk_1}(0^l; \mathbf{r}_1), \ldots, E_{pk_t}(0^l; \mathbf{r}_t), (f(1), \ldots, f(t)). \quad (5)$$

Now using the reproducibility property and Equation 4 again, we obtain $\mathbf{DS}_3 \equiv^c \mathbf{DS}_2$, which implies $\mathbf{DS}_1 \equiv^c \mathbf{DS}_2$, as desired. □

Second hardcore extraction method. The second construction allows us to extract any (a priori fixed) number of pseudorandom bits, where these bits are the last input block of the TDF.

Construction 3 *Let $\mathcal{E} = (Gen, E, Dec, R)$, t and D^t be as in Construction 1, and let $m = m(n)$ be an integer. The domain space of the TDF we build is $(D^t, \{0,1\}^m)$. We define $\mathcal{F} = (G, F, F^{-1})$ as follows.*

– *$G(1^n)$: Let $(pk, sk) \leftarrow Gen(1^n)$, and form $tk = (r_{1,1}, \ldots, r_{t,l}, r_1, \ldots, r_m)$, where $r_{i,j}$'s and r_h's are independent randomness values, and form $ik = (pk, \mathbf{c})$, where $\mathbf{c}$ consists of $t \cdot l + m$ encryptions of zero under pk using $r_{i,j}$'s and r_h's as randomness. Return (ik, tk).*
– *Define $F((pk, c_{1,1}, \ldots, c_{t,l}, c_1, \ldots, c_m), (pk_1, sk_1, \ldots, pk_t, sk_t, x))$ to be equal to*

$$(pk_1, \ldots, pk_t, c'_{1,1}, \ldots, c'_{t,l}, c'_1, \ldots, c'_m),$$

where $c'_{i,j} = R(c_{i,j}, b_{i,j}, sk_{i+1})$ for $1 \leq i \leq t-1$, $c'_{t,j} = R(c_{t,j}, b_{t,j}, sk_1)$ and $c'_h = R(c_h, x_h, sk_1)$, where $1 \leq h \leq m$, $1 \leq j \leq l$ and $b_{w,j}$ is the jth bit of sk_w, for $1 \leq w \leq t$.
– *$F^{-1}((r_{1,1}, \ldots, r_{t,l}, r_1, \ldots, r_m), (pk_1, \ldots, pk_t, c'_{1,1}, \ldots, c'_{t,l}, c'_1, \ldots, c'_m))$: as in the previous constructions.*

Hardcore function: *For $\mathcal{F}$ given above, we let $h\colon (D^t, \{0,1\}^m) \to \{0,1\}^m$ be defined as*

$$h(pk_1, sk_1, \ldots, pk_t, sk_t, x) = x.$$

Correctness of inversion is again evident and we have security as follows.

Theorem 3. *Let $\mathcal{F}$ and h be the TDF and hardcore function constructed according to Construction 3 based on $\mathcal{E} = (Gen, E, Dec, Rep)$, $m = m(n)$ and $t = t(n)$. Assuming $\mathcal{E}$ is k-ind t-circularly-secure, $\mathcal{F}$ is k-wise one-way and h is a hardcore function for $\mathcal{F}^k$.*

Proof. The first statement of the theorem follows immediately from Theorem 1, by having an inverter (attacking in the sense of Construction 1) simulate a hypothetical inverter against Construction 3 by generating $c_1, \ldots, c_m, c'_1, \ldots, c'_m$ on its own, where (by following notation used in Construction 3) c_i's are the auxiliary ciphertexts put in the index key, and c'_j's are encryptions of the last m bits of the input to the TDF (which are independent of the secret-keys part). This inverter against Construction 1 would essentially have the same advantage as that of the hypothetical inverter against Construction 3. Thus, in the sequel we focus on proving the second part of the theorem. We again give the proof for $k = 1$, since the proof for the general case follows similarly. Also, since the proof is similar to those presented before, we omit most of the details and present only a sketch of the proof.

To prove hardcore security we need to show that the following two distributions

$$\mathbf{DS}_1 = pk, E_{pk}(0^l; \mathbf{r}_1), \ldots, E_{pk}(0^l; \mathbf{r}_t), E_{pk}(0^m; \mathbf{r}_{t+1}), \\ pk_1, \ldots, pk_t, E_{pk_2}(sk_1; \mathbf{r}_1), \ldots, E_{pk_1}(sk_t; \mathbf{r}_t), E_{pk_1}(x; \mathbf{r}_{t+1}), x$$

$$\mathbf{DS}_2 = pk, E_{pk}(0^l; \mathbf{r}_1), \ldots, E_{pk}(0^l; \mathbf{r}_t), E_{pk}(0^m; \mathbf{r}_{l+1}), \\ pk_1, \ldots, pk_t, E_{pk_2}(sk_1; \mathbf{r}_1), \ldots, E_{pk_1}(sk_t; \mathbf{r}_t), E_{pk_1}(x; \mathbf{r}_{t+1}), x',$$

are computationally indistinguishable, where

$$(pk, sk), (pk_1, sk_1), \ldots, (pk_t, sk_t) \leftarrow Gen(1^n),\ \mathbf{r}_1, \ldots, \mathbf{r}_t \leftarrow \mathcal{R}_n^l,\ \mathbf{r}_{t+1} \leftarrow \mathcal{R}_n^m \text{ and } x, x' \leftarrow \{0,1\}^m.$$

The statement $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$ follows from the following two statements:

$$E_{pk_1}(x; \mathbf{r}_{t+1}), x' \equiv^c E_{pk_1}(x; \mathbf{r}_{t+1}), x', \tag{6}$$

and

$$pk, E_{pk}(0^l; \mathbf{r}_1), \ldots, E_{pk}(0^l; \mathbf{r}_t), pk_1, \ldots, pk_t, E_{pk_2}(sk_1; \mathbf{r}_1), \ldots, E_{pk_1}(sk_t; \mathbf{r}_t) \equiv^c \\ pk, E_{pk}(0^l; \mathbf{r}_1), \ldots, E_{pk}(0^l; \mathbf{r}_t), pk_1, \ldots, pk_t, E_{pk_2}(0^l; \mathbf{r}_1), \ldots, E_{pk_1}(0^l; \mathbf{r}_t), \tag{7}$$

where all the random variables are sampled as shown above; Equation 6 follows from semantic security of the scheme, and Equation 7 follows from the 1-ind t-circular security and reproducibility properties of the scheme. The proof is now complete. □

Remark 1. In many concrete settings, for a PKE $(Param, Gen, E, Dec)$, we have $Gen(1^n, par) \equiv (Pub_{par}(sk), sk)$, for a deterministic function Pub (recall that par is output by $Param$): namely, the public key is obtained deterministically from the secret key and public parameters. We may now easily modify Construction 3, so that the index key also includes par and that the evaluation function no longer takes pk as input (so its entire input is a bitstring), by computing $pk = Pub_{par}(sk)$ on its own. Now by taking $m \in \omega(t \cdot l)$ we would obtain a TDF (from the assumptions stated in Theorem 3) that hides a $(1 - o(1))$-fraction of its input bits.

4 Construction of CCA secure encryption

Rosen and Segev [31, Theorem 1] give a BB construction of CCA1-secure encryption from any $\omega(\log n)$-wise TDF and a BB CCA2-secure encryption from any $\Omega(n)$-wise TDFs. We may use our results and those of [31] to build CCA-secure encryption. For concreteness, we give the CCA1 construction here, which simplifies that obtained by directly instantiating [31] with our base encryption primitive. The construction for the CCA2 case is obtained similarly.

We fix the following notation. For $\mathbf{c} = (c_1, \ldots, c_m)$, $\mathbf{b} = (b_1, \ldots, b_m)$ we extend the reproduction function R so that $R(\mathbf{c}, \mathbf{b}, sk)$ denotes $(R(c_1, b_1, sk), \ldots, R(c_m, b_m, sk))$. In this section, for convenience, we present all our results wrt $t = 1$; the general case follows similarly. We give the CCA1 construction below.

Suppose $\mathcal{E} = (Gen, E, Dec, R)$ has randomness space $\mathcal{R}_n$ and secret-key space $\{0,1\}^l$. We build a many-bit scheme $\hat{\mathcal{E}} = (\hat{Gen}, \hat{E}, \hat{Dec})$ as follows.

- $\hat{Gen}(1^n)$ samples $\mathbf{r}_0^1, \mathbf{r}_1^1, \ldots, \mathbf{r}_0^k, \mathbf{r}_1^k \leftarrow \mathcal{R}_n^l$, $(pk, sk) \leftarrow Gen(1^n)$ and returns $(\mathbf{pk}, \mathbf{sk})$, where $\mathbf{pk} = (pk, E_{pk}(0^l; \mathbf{r}_0^1), E_{pk}(0^l; \mathbf{r}_1^1), \ldots, E_{pk}(0^l; \mathbf{r}_0^k), E_{pk}(0^l; \mathbf{r}_1^k))$ and $\mathbf{sk} = (\mathbf{r}_0^1, \mathbf{r}_1^1, \ldots, \mathbf{r}_0^k, \mathbf{r}_1^k)$;
- $\hat{E}_{\mathbf{pk}}(m)$ parses $\mathbf{pk} = (pk, \mathbf{c}_0^1, \mathbf{c}_1^1, \ldots, \mathbf{c}_0^k, \mathbf{c}_1^k)$, samples $(pk', sk') \leftarrow Gen(1^n)$, $u \leftarrow \{0,1\}^k$ and returns $(u, pk', E_{pk'}(m), \mathbf{c'}_{u_1}^1, \ldots, \mathbf{c'}_{u_k}^k)$, where, for $1 \le i \le k$, $\mathbf{c'}_{u_i}^i = R(\mathbf{c}_{u_i}^i, sk', sk')$ (Note that each $\mathbf{c'}_{u_i}^i$ is a self-encryption of sk');
- $\hat{Dec}_{\mathbf{sk}}(u, pk', c, \mathbf{c'}_{u_1}^1, \ldots, \mathbf{c'}_{u_k}^k)$ parses $\mathbf{sk} = (\mathbf{r}_0^1, \mathbf{r}_1^1, \ldots, \mathbf{r}_0^k, \mathbf{r}_1^k)$, lets sk_i, for each $1 \le i \le k$, be the plaintext obtained bit-by-bit by "opening" $\mathbf{c'}_{u_i}^i$ relative to public key pk' and randomness vector $\mathbf{r}_{u_i}^i$, checks whether $sk_1 = \cdots = sk_k$ (if this check fails it returns $\perp$), and returns $Dec_{sk_1}(c)$. Here by opening we mean finding the corresponding bit that encrypts to the given ciphertext under the specified randomness and public key.

In words, $\hat{E}$, given plaintext m, generates an index string u and a pair (pk', sk') under the base scheme, and returns u, an encryption of m under pk' as well as k self-encrypted versions of sk', where the ith version reuses the randomness embedded in $\mathbf{c'}_{u_i}^i$. The decryption algorithm opens the self-encrypted keys under the corresponding randomness determined by u, ensures all recovered secret keys are the same (consistency check), and, if so, decrypts the message-hiding ciphertext to recover the plaintext. The main idea behind the CCA1-security proof of $\hat{\mathcal{E}}$ is that, for $u' \leftarrow \{0,1\}^k$, as long as we possess only $(\mathbf{r}_{u'_1}^1, \mathbf{r}_{u'_2}^2, \ldots, \mathbf{r}_{u'_k}^k)$—as opposed to the entire randomness vector $\mathbf{sk}$—then we can efficiently do the consistency check for (and decrypt) any (even adversarially chosen) ciphertext $(u, \ldots)$, for which $u \neq \bar{u'}$. (Here $\bar{u'}$ denotes the bitwise complement of u.) Also, for any efficient adversary the probability that it ever happens that $u = \bar{u'}$ is negligible—Here is where we use the fact that $k \in \omega(\log n)$. This idea enables us to simulate a CCA1 experiment using a k-ind 1-circular security experiment as follows. Assume we are given $(pk', \mathbf{c}'_1, \ldots, \mathbf{c}'_k)$ as part of the k-ind 1-circular security: choose $u' \leftarrow \{0,1\}^k$, generate $\mathbf{r}_{u'_1}^1, \mathbf{r}_{u'_2}^2, \ldots, \mathbf{r}_{u'_k}^k \leftarrow \mathcal{R}_n^l$ and $(pk, sk) \leftarrow Gen(1^n)$, publish the public key as $(pk, \mathbf{c}_0^1, \mathbf{c}_1^1, \ldots, \mathbf{c}_0^k, \mathbf{c}_1^k)$, where $\mathbf{c}_{\bar{u}_i}^i = Rep(\mathbf{c}'_i, 0, sk)$ and $\mathbf{c}_{u_i}^i = E_{pk}(0^l; \mathbf{r}_i)$, for any $1 \le i \le k$. As argued above, we would be able to reply to all decryption queries of the adversary (except with negligible probability). Finally, in response to the left-or-right query (m_0, m_1) of the adversary, we return $(\bar{u'}, pk', E_{pk'}(m_b), \mathbf{c}'_1, \ldots, \mathbf{c}'_k)$, where $b \leftarrow \{0,1\}$. Now if $\mathbf{c'}_i$'s are self-encrypted keys, the adversary would achieve exactly the same advantage of distinguishing between $b = 0$ and $b = 1$ under this simulated experiment as that it would achieve under the normal CCA1 game. On the other hand, if $\mathbf{c'}_i$'s are encryptions of zeros, the ability of any adversary to distinguish between $b = 0$ and $b = 1$ would easily reduce to an attack against the CPA-security of $\mathcal{E}$.

The proof of the following theorem essentially follows the line of arguments given above.

Theorem 4. *If $k \in \omega(\log n)$ and $\mathcal{E}$ is a reproducible, k-ind circularly-secure bit-encryption scheme, then $\hat{\mathcal{E}}$, constructed above, is CCA1 secure.*

The construction above is *non-shielding*, in the sense of [19], since the constructed decryption function queries the base encryption function.[8] By [19], there are no BB shielding constructions of CCA1-secure encryption from CPA-secure encryption. Since our base assumptions are strictly stronger than CPA security (at least in a BB sense), a natural question is whether or not it is possible to give a shielding construction based on our assumptions. At this point, we do not know the answer to this question, but as we show below, there exists an encryption primitive implied by our assumptions, based on which a non-shielding CCA1-construction is possible, but from which no *fully-blackbox*[9] shielding CCA1-construction is possible. Our new encryption primitive is an extension of CPA-secure encryption, asking that security holds even when encrypting certain *randomness-dependent messages.*

Definition 2. *A bit-encryption scheme $\mathcal{E} = (Gen, E, Dec)$ with randomness space $\{0,1\}^\rho$ is q-randomness-dependent-message (RDM) secure if*

$$\{E_{pk_1^1}(r_1; r), \ldots, E_{pk_\rho^1}(r_\rho; r)\}, \ldots, \{E_{pk_1^q}(r_1; r), \ldots, E_{pk_\rho^q}(r_\rho; r)\} \\ \equiv^c \{E_{pk_1^1}(0; r), \ldots, E_{pk_\rho^1}(0; r)\}, \ldots, \{E_{pk_1^q}(0; r), \ldots, E_{pk_\rho^q}(0; r)\},$$

where $r \leftarrow \{0,1\}^\rho$ and all public keys are chosen at random according to Gen. For better readability, we made the inclusion of the public keys implicit.

In the definition above, since we are encrypting the randomness string bitwise, we should use independent public keys for each encryption. Otherwise, an adversary can easily distinguish between the two distributions. Our definition is basically an adaptation of those of [7] to the bit-encryption case. We show below that this primitive is implied by our assumptions.

Given $\mathcal{E} = (Gen, E, Dec, R)$, define $\mathcal{E}' = (Gen', E', Dec')$, whose randomness space is the key space of $\mathcal{E}$, as follows: $Gen'(1^n)$ samples $(pk, sk) \leftarrow Gen(1^n)$ and $r \leftarrow \mathcal{R}_n$ and returns $pk = E_{pk}(0; r)$ and $sk = r$. The encryption $E'_c(b; (pk', sk'))$ returns $(pk', R(c, b, sk'))$; and, finally, $Dec'_r(pk', c')$ returns the bit b for which $E_{pk'}(b; r) = c'$. Using ideas described in Section 3 we can show, for any poly q, if $\mathcal{E}$ is q-ind circularly secure, then $\mathcal{E}'$ is q-RDM secure.

Next, we show q-RDM-secure encryption easily implies q-wise one-way TDFs. Let $\mathcal{E}$'s randomness space be $\{0,1\}^\rho$, and define TDF $\mathcal{TF} = (G, F, F^{-1})$ as follows. G runs $Gen(1^n)$ ρ times and returns $ik = (pk_1, \ldots, pk_\rho)$ and $tk = (sk_1, \ldots, sk_\rho)$; let F's domain space be $\mathcal{R}_n$ and define $F_{pk_1,\ldots,pk_\rho}(r)$ to equal $(F_{pk_1}(r_1; r), \ldots, F_{pk_\rho}(r_\rho, r))$. The inversion algorithm F^{-1} is defined in an obvious way. Now it is not hard to show if $\mathcal{E}$ is q-RDM secure, $\mathcal{TF}$ is q-wise one-way. A summary of the discussion above is the following.

Corollary 1. *For any $q \in \omega(\log n)$ there exists a shielding BB construction of CCA1-secure encryption from q-RDM-secure bit-encryption.*

[8] A non-shielding construction of an encryption scheme from another is one in which the constructed decryption algorithm does not query the encryption function of the base scheme. See [19] for a more formal definition.

[9] We are using the notion of fully-blackbox reductions as defined in [30].

We now show the BB separation of [19], stating that there are no shielding BB constructions of CCA1-secure encryption from CPA-secure encryption, extends even if the base scheme is RDM-secure, for any poly-bounded q. Combined with the corollary above, this gives us an encryption primitive which permits a non-shielding BB CCA1-secure construction, but from which no shielding BB CCA1-secure construction is possible. We first start with an informal description of the separation model of [19] and then give the formal definitions.

Specifically, [19] introduces a tuple of oracles $\mathcal{O} = (\mathbf{g}, \mathbf{e}, \mathbf{d}, \mathbf{w}, \mathbf{u})$, where $\mathcal{O}_1 = (\mathbf{g}, \mathbf{e}, \mathbf{d})$ model an idealized encryption scheme (when the oracle is chosen at random), and $\mathcal{O}_2 = (\mathbf{d}, \mathbf{w})$ are two security-weakening components, defined based on $\mathcal{O}_1$. They show that (*) for any candidate oracle-construction $\mathcal{E} = (Gen^{\mathcal{O}_1}, Enc^{\mathcal{O}_1}, Dec^{\mathbf{g},\mathbf{d}})$ there exists an oracle-adversary $\mathcal{A}^{\mathcal{O}}$, which is unbounded in time but poly-bounded in the number of oracle calls, that breaks the CCA1 security of $\mathcal{E}$ *almost always* (i.e., except for measure-zero of oracles) [19, Theorem 2]. To establish their separation result, they show that for *almost any* selection of $\mathcal{O}$ (i.e., measure-one of oracles), $(\mathbf{g}, \mathbf{e}, \mathbf{d})$ is CPA-secure against any oracle-adversary $\mathcal{A}^{\mathcal{O}}$ with constraints mentioned above, i.e., poly-bounded in the number of queries but otherwise unbounded [19, Theorem 1].[10] Therefore, to rule out shielding fully-BB constructions of CCA1-secure encryption from a new encryption primitive, it suffices to prove (**) with respect to the new primitive. This is what we do below wrt RDM secure encryption. We first give the formal description of the oracles as used in [19].

Definition 3. *([19]) Define ψ, a distribution on oracles $(\mathbf{g}, \mathbf{e}, \mathbf{d}, \mathbf{w}, \mathbf{u})$, defined for each $n \in \mathbb{N}$, as follows.*

- $\mathbf{g}\colon \{0,1\}^n \mapsto \{0,1\}^{3n}$ *is a random one-to-one function. Function* $\mathbf{g}$ *is considered as a key generator, with sk being the secret key and $pk = \mathbf{g}(sk)$ as the public key.*
- $\mathbf{e}\colon \{0,1\}^{3n} \times \{0,1\} \times \{0,1\}^n \mapsto \{0,1\}^{3n}$ *is a random one-to-one function.*
- $\mathbf{d}\colon \{0,1\}^n \times \{0,1\}^{3n} \mapsto \{0,1,\perp\}$ *is the unique function specified based on $(\mathbf{g}, \mathbf{e})$, where $\mathbf{d}(sk, c) = b$ if there exists $r \in \{0,1\}^n$ such that $\mathbf{e}(\mathbf{g}(sk), b, r) = c$; otherwise, $\mathbf{d}(sk, c) = \perp$.*
- $\mathbf{w}\colon \{0,1\}^{3n} \to \{0,1\}^{3n \times n} \cup \{\perp\}$ *is a random function sampled as follows. For $\mathbf{w}(pk)$, if $\mathbf{g}^{-1}(pk) = \emptyset$ then $\mathbf{w}(pk) = \perp$; otherwise, sample $r_1, \ldots, r_n \leftarrow \{0,1\}^n$ and return*

$$(\mathbf{e}(pk, sk_1, r_1), \ldots, \mathbf{e}(pk, sk_n, r_n)),$$

  where $sk = \mathbf{g}^{-1}(pk)$.
- $\mathbf{u}\colon \{0,1\}^{3n} \times \{0,1\}^{3n} \mapsto \{\top, \perp\}$ *is a deterministic function which returns $\top$ if there exists sk, b and r such that $\mathbf{g}(sk) = pk$ and $\mathbf{e}(pk, b, r) = c$, and returns $\perp$, otherwise.*

For consistency, we may sometimes write $\mathbf{e}(pk, b, r)$ and $\mathbf{d}(sk, c)$, respectively, as $\mathbf{e}_{pk}(b; r)$ and $\mathbf{d}_{sk}(c)$.

The following theorem, which is from [19], shows that, informally speaking, for any candidate shielding construction, there exists an inefficient adversary (which, however, makes a polynomial number of queries) that "almost-always" breaks the constructed scheme in a CCA1 sense.

[10] We abuse notation somewhat here. By scheme $(\mathbf{g}, \mathbf{e}, \mathbf{d})$ we mean the oracle-aided scheme $(G^{\mathbf{g}}, E^{\mathbf{e}}, D^{\mathbf{d}})$ which merely copies its oracle's functionality, e.g., $Gen(s)$ simply returns $\mathbf{g}(s)$.

Theorem 5. *([19]) Fix a shielding bit-encryption construction $(Gen^{\circ}, E^{\circ}, Dec^{\circ})$. There exists a CCA1 adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, where $\mathcal{A}$ is poly-bounded in the bumber of queries but unbounded otherwise, for which it holds that*

$$\Pr\left[(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1^{Dec^{\mathbf{g},\mathbf{d}}(SK),\mathcal{O}}(PK)\ ;\ \mathcal{A}_2^{\mathcal{O}}(c, \sigma) = b\right] \geq 1 - \frac{1}{n}, \tag{8}$$

where the probability is computed over $\mathcal{A}$'s random coins, $\mathcal{O} = (\mathbf{g}, \mathbf{e}, \mathbf{d}, \mathbf{w}, \mathbf{u}) \leftarrow \psi$, $b \leftarrow \{0,1\}$ and the random coins involved in computing $(PK, SK) \leftarrow Gen^{\mathcal{O}}(1^n)$ and $c \leftarrow E^{\mathcal{O}}(m_b)$.

We give and prove the following theorem, a CPA version of which was proved in [19].

Theorem 6. *For any (possibly) inefficient adversary $\mathcal{A}$, which is poly-bounded in the number of queries, and poly-bounded q, there exists a negligible function $negl$ such that*

$$\Pr_{\mathcal{O}=(\mathbf{g},\mathbf{e},\mathbf{d},\mathbf{w},\mathbf{u})\leftarrow\psi}\left[\Pr\left[\mathcal{A}^{\mathcal{O}}(ds_b) = b\right] \leq \tfrac{1}{2} + negl(n)\right] \geq 1 - \tfrac{1}{2^{n/2}}, \tag{9}$$

where the inner probability is over $b \leftarrow \{0,1\}$, the randomness of $\mathcal{A}$ and $ds_b \leftarrow \mathcal{DS}_b$, for

$$\mathcal{DS}_0 \equiv \{\mathbf{e}_{pk_1^1}(r_1; r), \ldots, \mathbf{e}_{pk_n^1}(r_n; r)\}, \ldots, \{\mathbf{e}_{pk_1^q}(r_1; r), \ldots, \mathbf{e}_{pk_n^q}(r_n; r)\} \tag{10}$$

$$\mathcal{DS}_1 \equiv \{\mathbf{e}_{pk_1^1}(0; r), \ldots, \mathbf{e}_{pk_n^1}(0; r)\}, \ldots, \{\mathbf{e}_{pk_1^q}(0; r), \ldots, \mathbf{e}_{pk_n^q}(0; r)\}, \tag{11}$$

in which $r \leftarrow \{0,1\}^n$ and the tuples $(pk_1^1, \ldots, pk_n^1)\ \ldots\ (pk_1^q, \ldots, pk_n^q)$ are formed, for every $1 \leq i \leq n$ and $1 \leq j \leq q$, by sampling $sk_i^j \leftarrow \{0,1\}^n$ and setting $pk_i^j = \mathbf{g}(sk_i^j)$.

We point out a few comments. First, the choice of $1 - \frac{1}{2^{n/2}}$ for the quantity of Equation 9 is not strict; we made that choice just to be consistent with that of [19]. It can in fact be, for any constant $c < 1$, as large as $1 - \frac{1}{2^{n/c}}$ by choosing appropriately the negligible function used to bound the inner probability in Equation 9.

Proof. The way to go about proving Equation 9 is now standard, so we only sketch the main ingredients of the proof, without being strict about the choice of parameters. Basically, the main idea of the proof is that $(\mathbf{g}, \mathbf{e}, \mathbf{d})$ represents an ideal encryption scheme (and in the absence of oracles $\mathbf{w}$ and $\mathbf{u}$ the theorem is indeed a standard fact). Fix adversary $\mathcal{A}$. To prove that there exists a negligible function *negl* for which Equation 9 holds, it suffices to show that there exists a poly $p = p(n)$ such that the inner probability— i.e., $\Pr\left[\mathcal{A}^{\mathcal{O}}(ds_b) = b\right]$— if still conditioned on the choice of $\mathcal{O} = (\mathbf{g}, \mathbf{e}, \mathbf{d}, \mathbf{w}, \mathbf{u})$, is upperbounded by $1/2 + p/2^n$; Equation 9 then follows using a simple averaging argument (e.g., applying Markov's inequality) by defining *negl* appropriately based on p.

We first fix some notation. Let PK_1 be the set of pubic keys given to $\mathcal{A}$ as part of its input, ds_b, and let PK_2 be the set of public keys that $\mathcal{A}$ obtains by querying $\mathbf{g}$. Also, let C be the set of pairs of ciphertexts/public keys which $\mathcal{A}$ can retrieve as part of ds_b, as well as those $\mathcal{A}$ obtains by querying $\mathbf{e}$ or by querying $\mathbf{w}$. First, as in [19], we may assume that (1) $\mathcal{A}$ never queries $\mathbf{u}$, (2) $\mathcal{A}$ only queries $\mathbf{w}$ on inputs $pk \in PK_1$ and (3) $\mathcal{A}$ never queries $\mathbf{d}$. Indeed, for any $(pk, c) \notin C$, the query $\mathbf{u}(pk, c)$ is answered with $\perp$ except with an inverse-exponential probability (since $\mathbf{g}$ is length tripling and also $\mathbf{e}$ is "almost" length-tripling for fixed pk); if $(pk, c) \in C$, however, then $\mathcal{A}$ already knows the answer and there is no point in calling $\mathbf{u}$. Similarly, we can argue that, or any $pk \notin PK_1 \cup PK_2$, the query $\mathbf{w}(pk)$ is answered with $\perp$ unless with an inverse-exponential

probability; also, if $pk \in PK_2$, then $\mathcal{A}$ itself can sample the answer by querying $\mathbf{e}$. Using similar reasoning, we can show that any query $\mathbf{d}(c)$ that $\mathcal{A}$ trivially does not know the answer to is replied with $\perp$ except with an inverse-exponential probability. Now assuming $\mathcal{A}$ makes at most p_1 queries, by observation (2), we may now assume that $\mathcal{A}$ never queries $\mathbf{w}$, but instead $\mathcal{A}$'s input includes $p_1 \times n^2 \times q$ more ciphertexts, where for each public key pk_i^j, we include p_1 bit-by-bit encryptions of pk_i^j's secret key (so $p_1 \times n$ encryptions for each public key). We also let C contain all these new ciphertexts/public keys. Now since $\mathcal{A}$ only queries $\mathbf{g}$ and $\mathbf{e}$, we consider the events that (a) one of $\mathbf{g}$ queries results in a public key which is in PK_1, and (b) $\mathcal{A}$ makes a query $\mathbf{e}(pk, b, r)$, which is answered with a ciphertext c, for which it holds that $(pk, c) \in C$. If none of these events occur, then we can show that the probability that $\mathcal{A}$ can determine b is greater than $1/2$ with at most an inverse-exponential quantity. Moreover, both these two events can, in turn, be bounded by the same quantity. □

Using standard techniques (especially applying the Borel-Cantelli lemma) [23], the inequality of Equation 9 may then be used to conclude that for measure-one of oracles $\mathcal{O} = (\mathbf{g}, \mathbf{e}, \mathbf{d}, \mathbf{w}, \mathbf{u})$, the scheme $(\mathbf{g}, \mathbf{e}, \mathbf{d})$ is q-RDM secure against all oracle-adversaries $\mathcal{A}^{\mathcal{O}}$. Similarly, Theorem 5 implies that for any candidate shielding construction, there exists an oracle adversary $\mathcal{A}$ that, for measure one of oracles $\mathcal{O}$, $\mathcal{A}^{\mathcal{O}}$ breaks the CCA1-security of the construction. Therefore, we obtain the following corollary.

Corollary 2. *For any $q \in \omega(\log n)$ there exists a non-shielding blackbox construction of CCA1 encryption from q-RDM-secure encryption. Moreover, for any poly-bounded q, there exists no shielding blackbox construction of CCA1 encryption from q-RDM-secure encryption.*

We note that it seems that one can generalize Corollary 2 to rule out the existence of shielding BB CCA1 constructions from a large class of encryption primitives whose security is defined in terms of indistinguishability against passive attacks (i.e., no decryption oracles). In other words, the BB separation generalizes to any (base) security requirement that is realized by an ideal encryption scheme $(\mathbf{g}, \mathbf{e}, \mathbf{d})$ in the presence of $(\mathbf{w}, \mathbf{u})$; for example, Corollary 2 still holds true if RDM security is replaced with circular security.

5 Deterministic encryption (DE) and instantiations

We start by reviewing some basic facts related to entropy. The *min-entropy* of a distribution $\mathcal{D}$ is defined as $H_\infty(\mathcal{D}) = \min_{d \in \mathcal{D}} \log(1/\Pr[\mathcal{D} = d])$. If $l = H_\infty(\mathcal{D})$ we call $\mathcal{D}$ an *l-source*. We also recall the notion of *average min entropy*, formalized by Dodis et al. [16], defined as $\tilde{H}_\infty(\mathcal{X}|\mathcal{Y}) = -\log\left(E_{y \leftarrow Y}(2^{-H_\infty(\mathcal{X}|\mathcal{Y}=y)})\right)$.

DE schemes. Since a DE scheme is syntactically the same as a TDF, we denote a DE scheme as $\mathcal{DE} = (G, F, F^{-1})$. We make a few assumptions in this section. We assume that the conditions stated in Remark 1 hold for any randomized encryption (RE) scheme used in this section: that is, $Gen_1(1^n) \equiv Pub_{par}(sk)$, where Pub is a deterministic function; we often drop par. Throughout this section, we use $l = l(n)$ to denote the length of a secret key of a RE scheme, and also the message length of a DE scheme.

We start by defining an extended notion of circular security, requiring circular security hold even if the secret key is sampled from a non-full-entropy distribution. For technical reasons, we need to allow some information about the secret key to be leaked, assuming the average min entropy of

the secret key conditioned on the leaked information is high. The following definition generalizes a similar definition of [13] to the average case. We note it is possible to prove our results wrt the weaker definition of [13], but the proofs become more complex.

Definition 4. *We say a bit-encryption scheme $\mathcal{E} = (Gen, E, Dec)$ is (λ, l)-entropy circularly secure if for any joint distribution $(\mathcal{SK}, \mathcal{X})$, with $\tilde{H}_\infty(\mathcal{SK}|\mathcal{X}) \geq \lambda$, we have*

$$(pk, E_{pk}(sk), E_{pk}(1), x) \equiv^c (pk, E_{pk}(0^l), E_{pk}(0), x),$$

where $(sk, x) \leftarrow (\mathcal{SK}, \mathcal{X})$ and $pk = Pub(sk)$.

Next we define a strengthening of the notion of [13], which adds the requirement that the public key distributions formed under high-entropy secret keys be computationally indistinguishable. This may be guaranteed if, e.g., *Pub* is a *strong randomness extractor* [27], as is the case with known circularly-secure schemes [11,12].

Definition 5. *We say a bit-encryption scheme $\mathcal{E} = (Gen, E, Dec)$ is strongly-(λ, l)-entropy circularly secure if*

- *(a) for any λ-source $\mathcal{SK}$,*

$$(pk, E_{pk}(sk), E_{pk}(1)) \equiv^c (pk, E_{pk}(0^l), E_{pk}(0)),$$

  where $sk \leftarrow \mathcal{SK}$ and $pk = Pub(sk)$; and
- *(b) for any λ-sources $\mathcal{SK}_1$ and $\mathcal{SK}_2$, it holds that $Pub(\mathcal{SK}_1) \equiv^c Pub(\mathcal{SK}_2)$.*

We now define our DE security notion, which is essentially the single-message, indistinguishability-based notion of [9]. See [9] for definitional equivalences.

Definition 6. *We say $\mathcal{DE} = (G, F, F^{-1})$ is secure wrt indistinguishability of λ-source inputs (shortly, (λ, l)-IND secure) if for any λ-sources $\mathcal{M}_0$ and $\mathcal{M}_1$, it holds $(ik, F_{ik}(\mathcal{M}_0)) \equiv^c (ik, F_{ik}(\mathcal{M}_1))$ where $(ik, tk) \leftarrow G(1^n)$.*

Now we show that by starting from a reproducible encryption scheme which provides strong (λ, l)-entropy circular security, Construction 1 immediately gives us a (λ, l)-IND secure deterministic scheme—i.e., it preserves the parameters.

Theorem 7. *Let $\mathcal{E} = (Gen, E, Dec, R)$ be a reproducible bit-encryption scheme and $\mathcal{DE} = C_1(\mathcal{E}, 1)$ be the DE scheme built in Construction 1 based on $\mathcal{E}$ and $t = 1$.*[11] *If $\mathcal{E}$ is strongly-(λ, l)-entropy circularly secure $\mathcal{F}$ is (λ, l)-IND secure.*

Proof. Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two arbitrary λ-sources on $\{0,1\}^l$. We need to prove that the following two distributions,

$$\mathcal{DS}_1 \equiv (pk, E_{pk}(0^l; \mathbf{r}), pk_1, E_{pk_1}(sk_1; \mathbf{r})),$$

$$\mathcal{DS}_2 \equiv (pk, E_{pk}(0^l; \mathbf{r}), pk_2, E_{pk_2}(sk_2; \mathbf{r}))$$

[11] Here we are working with a modified version of Construction 1 stated in Remark 1.

are computationally indistinguishable, where $(pk, sk) \leftarrow Gen(1^n)$, $sk_1 \leftarrow \mathcal{D}_1$, $pk_1 = Pub(sk_1)$, $sk_2 \leftarrow \mathcal{D}_2$, $pk_2 = Pub(sk_2)$ and $\mathbf{r} \leftarrow \mathcal{R}_n^l$. From strong-$(\lambda, l)$-entropy circularly security and reproducibility of $\mathcal{E}$, we may deduce

$$\begin{aligned} \mathcal{DS}_1 &\equiv^c (pk, E_{pk}(0^l; \mathbf{r}), pk_1, E_{pk_1}(0^l; \mathbf{r})), \\ \mathcal{DS}_2 &\equiv^c (pk, E_{pk}(0^l; \mathbf{r}), pk_2, E_{pk_2}(0^l; \mathbf{r})), \end{aligned} \tag{12}$$

where all the random variables are sampled as above. Now given that $pk_1 \equiv^c pk_2$ (which is implied by strong (λ, l)-entropy circularly security of $\mathcal{E}$) we conclude from Equations 12 that $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$, and the proof is complete. □

Next we show the "weaker" entropy circular security assumption also gives rise to DE schemes, but with looser security bounds. Our construction employs the encrypt-with-hardcore (EWH) technique, described in the introduction. To this end, we assume that the ciphertext space of our (base) encryption scheme is also a bitstring space, since our construction (by employing the EWH technique) results in doubly-encrypted ciphertexts. We give the main theorem below.

Theorem 8. *Let $\mathcal{E} = (Gen, E, Dec, R)$ be a reproducible (λ, l)-entropy circularly secure encryption scheme, with randomness space $\mathcal{R}_n = \{0,1\}^{p_r}$. There exists an $(l + p_r + u, 2l + p_r - \lambda)$-IND-secure deterministic encryption scheme, where $u \in \omega(\log n)$ is an arbitrary function.*

We give an outline of the proof, and leave the full proof to a later version of the paper. The first step is to show that we can use reproducibility of $\mathcal{E}$ to encrypt any arbitrarily-long message using a p_r-bit-long randomness string, by reusing randomness across different public keys. Now consider the TDF given by Construction 3, based on $t = 1$ and $m = l + p_r - \lambda$, and define the function $hc(sk, x) = (h, h(x))$, where the function $h\colon \{0,1\}^m \mapsto \{0,1\}^{p_r}$ is chosen from a *family of universal hash functions*; as the next step we show that hc is a hardcore function for the TDF. Now to be able to apply the EWH method, we need to show that $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$, for

$$\begin{aligned} \mathcal{DS}_1 &\equiv (h, h(x), E_{pk}(sk; \mathbf{r}_1), E_{pk}(x; \mathbf{r}_2), E_{pk}(0^l; \mathbf{r}_1), E_{pk}(0^{|x|}; \mathbf{r}_1)), \text{ and} \\ \mathcal{DS}_2 &\equiv (h, y, E_{pk}(sk; \mathbf{r}_1), E_{pk}(x; \mathbf{r}_2), E_{pk}(0^l; \mathbf{r}_1), E_{pk}(0^{|x|}; \mathbf{r}_1)), \end{aligned}$$

where $y \leftarrow \{0,1\}^{p_r}$, $(sk, x) \leftarrow (\mathcal{SK}, \mathcal{X})$, $pk = Pub(sk)$ and $H_\infty(\mathcal{SK}, \mathcal{X}) \geq l + p_r + u$. (Also, $\mathbf{r}_1$ and $\mathbf{r}_2$ are chosen independently.) Now since $\tilde{H}_\infty(\mathcal{SK}|\mathcal{X}) \geq \lambda + u$ (which follows from standard average min-entropy facts) we may appeal to (λ, l)-entropy circular security of $\mathcal{E}$ to replace $E_{pk}(sk; \mathbf{r}_1)$, in both $\mathcal{DS}_1$ and $\mathcal{DS}_1$, with an all-zero encryption; in the next step we do the same for $E_{pk}(x; \mathbf{r}_1)$ (i.e., we get rid of the occurrences of x as a plaintext); and finally, using the facts that $\tilde{H}_\infty(\mathcal{X}|\mathcal{SK}) \geq p_r + u$, that h is an *average-case extractor* and that $u \in \omega(\log n)$, we replace $h(x)$ with a random string.

6 Realizations

Throughout this section we will be working with multiplicative notation for groups. For a group element g, we denote the inverse of g by g^{-1} and define $g_1/g_2 = g_1 \cdot g_2^{-1}$. We also denote the identity element by 1, and we define $g^0 = 1$, and for $x > 1$, $g^x = g \cdot g^{x-1}$. If $\mathbf{g} = (g_1, \ldots, g_l)$ and r is an integer we define $\mathbf{g}^r = (g_1^r, \ldots, g_l^r)$. Finally we define $(b_1, \ldots, b_l) \cdot (g_1, \ldots, g_l) = \prod_{1 \leq i \leq l} g_i^{b_i}$.

6.1 From the decisional Diffie -Hellman (DDH) assumption

Let $\mathcal{G}$ be *group generator*, that is a PPT algorithm that on input 1^n, outputs $(\mathbb{G}, g, o)$, where $\mathbb{G}$ is the description of a group, $g \in \mathbb{G}$ and $o = |\mathbb{G}|$ is a prime number. We say that $\mathcal{G}$ is *DDH hard* if

$$\left\{\mathbb{G}, |\mathbb{G}|, g_1, g_2, g_1^d, g_2^d\right\}_{n\in\mathbb{N}} \equiv^c \{\mathbb{G}, |\mathbb{G}|, g_1, g_2, g_3, g_4\}_{n\in\mathbb{N}},$$

where $\mathbb{G}$ is chosen by running $\mathcal{G}(1^k)$, and $g_1, \ldots, g_4 \leftarrow \mathbb{G}$ and $d_1, d_2 \leftarrow \mathbb{Z}_o$.

We present the encryption scheme of [11], which we refer to as the BHHO scheme, below.

Definition 7. *(From [11]) Define $\mathcal{E} = (S, Gen, E, Dec)$, which is parameterized over an integer l (which we instantiate later), as follows.*

- *$S(1^n)$: runs $(\mathbb{G}, g, o) \leftarrow \mathcal{G}(1^n)$ and returns $param = (\mathbb{G}, g, \mathbf{g})$, where $\mathbf{g} \leftarrow \mathbb{G}^l$;*
- *$Gen(1^n)$: samples the secret key as $sk \leftarrow \{0,1\}^l$ and sets the public key $pk = sk \cdot \mathbf{g}$;*
- *$E_{pk}(g_1; r))$: samples $r \leftarrow \mathbb{Z}_q$ and returns $(\mathbf{g}^r, pk^r \cdot g_1)$; and*
- *$D_{sk}((\mathbf{g}', g'))$: clear from the encryption algorithm.*

Reproducibility. We now verify the reproducibility property for every fixed choice of *param*. To do this, we need to show that, for every $g_1, g_2 \in \mathbb{G}$ and $sk_1, sk_2 \in\leftarrow \{0,1\}^l$ and $r \in \mathbb{Z}_o$, from

$$(sk_1 \cdot \mathbf{g}, (\mathbf{g}^r, pk^r \cdot g_1), g_2, \mathbf{g}, sk_2),$$

one can compute $(\mathbf{g}^r, (sk_2 \cdot \mathbf{g})^r \cdot g_2)$, which is trivial considering that the last quantity is indeed $(\mathbf{g}^r, (sk_2 \cdot \mathbf{g}^r) \cdot g_2)$.

Next, we show that the proof of security in [11] easily extends to prove strong-(λ, l)-entropy circular security, where the ratio $\frac{\lambda}{l}$ can get as small as we want. (Formally, for any a priori fixed polynomial $p = p(n)$ we can have an instantiation of the above scheme which is $(\frac{l}{p}, l)$-entropy circularly secure.) The proof of entropy circular security is, however, immediate from the proof of [11] and we include it here only for self-containment.

Proposition 1. *(From [11]) Assuming DDH-hardness for $\mathcal{G}$, the following holds: for any polynomials $l = l(n)$ and $v = v(n)$ and (any efficiently computable) sequence of group elements, $(g_{1,1}, \ldots, g_{1,l}, \ldots, g_{v,1}, \ldots, g_{v,l})$, it holds that $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$, for*

$$\mathcal{DS}_1 = \begin{pmatrix} g_1 & g_2 & \cdots & g_l \\ g_1^{r_1} \cdot g_{1,1} & g_2^{r_1} \cdot g_{1,2} & \cdots & g_l^{r_1} \cdot g_{1,l} \\ \vdots & \vdots & \ddots & \vdots \\ g_1^{r_v} \cdot g_{v,1} & g_2^{r_v} \cdot g_{v,2} & \cdots & g_l^{r_v} \cdot g_{v,l} \end{pmatrix} \tag{13}$$

$$\mathcal{DS}_2 = \begin{pmatrix} g_1 & g_2 & \cdots & g_l \\ g_1^{r_1} & g_2^{r_1} & \cdots & g_l^{r_1} \\ \vdots & \vdots & \ddots & \vdots \\ g_1^{r_v} & g_2^{r_v} & \cdots & g_l^{r_v} \end{pmatrix} \tag{14}$$

where $\mathbb{G}$ is chosen by running $\mathcal{G}(1^n)$, $g_1, \ldots, g_l \leftarrow \mathbb{G}$ and $r_1, \ldots, r_v \leftarrow \mathbb{Z}_{|\mathbb{G}|}$.

Theorem 9. *(Implicit in [11]) Let $o = o(n)$ be an upper-bound on the size of any group output by $\mathcal{G}(1^n)$. Letting $\lambda = \log o + h$, where $h \in \omega(\log n)$ is an arbitrary function, and $l > \lambda$ be an arbitrary value, the scheme of Definition 7, when parameterized with l, is strong-(λ, l)-entropy circularly secure.*

Proof. We first show Condition (a) of Definition 5 for the BHHO scheme. To encrypts the bits of the secret key, we encrypt bit b by encrypting g^b. Recall that, for a general scheme, (λ, l)-entropy circular security asks that

$$pk, E_{pk}(sk), E_{pk}(1), x \equiv^c pk, E_{pk}(0^l), E_{pk}(0), x,$$

where $(\mathcal{SK}, \mathcal{X})$ are arbitrary joint distributions with $\hat{H}_\infty(\mathcal{SK}|\mathcal{X}) \geq \lambda$, $(sk, x) \leftarrow (\mathcal{SK}, \mathcal{X})$ and $pk = Pub(sk)$. For the rest of the proof fix $(\mathcal{SK}, \mathcal{X})$, where $\hat{H}_\infty(\mathcal{SK}|\mathcal{X}) \geq \lambda = \log o + h$. To prove the desired indistinguishability we introduce the following distributions, where in all of them, $\mathbb{G}$ is chosen by running $\mathcal{G}(1^n)$, $g_1, \ldots, g_l, g_{l+1} \leftarrow \mathbb{G}$, $r_1, \ldots, r_l, r \leftarrow \mathbb{Z}_{|\mathbb{G}|}$, $sk \leftarrow \mathcal{SK}$ and $x \leftarrow \mathcal{X}$.

$$\mathcal{DS}_1 = \{g_1, \ldots, g_l, sk \cdot \mathbf{g}\}, \{g_1^{r_i}, \ldots, g_l^{r_i}, (sk \cdot \mathbf{g})^{r_i} \cdot g^{sk_i}\}_{1 \leq i \leq l}, \{g_1^r, \ldots, g_l^r, (sk \cdot \mathbf{g})^r \cdot g\}, x \tag{15}$$

$$\mathcal{DS}_2 = \{g_1, \ldots, g_l, sk \cdot \mathbf{g}\}, \{g_1^{r_i}, \ldots, g_{i-1}^{r_i}, \frac{g_i^{r_i}}{g}, g_{i+1}^{r_i}, \ldots, g_l^{r_i}, (sk \cdot \mathbf{g})^{r_i}\}_{1 \leq i \leq l}, \{g_1^r, \ldots, g_l^r, (sk \cdot \mathbf{g})^r \cdot g\}, x \tag{16}$$

$$\mathcal{DS}_3 = \{g_1, \ldots, g_l, g_{l+1}\}, \{g_1^{r_i}, \ldots, g_{i-1}^{r_i}, \frac{g_i^{r_i}}{g}, g_{i+1}^{r_i}, \ldots, g_l^{r_i}, g_{l+1}^{r_i}\}_{1 \leq i \leq l}, \{g_1^r, \ldots, g_l^r, g_{l+1}^r \cdot g\}, x \tag{17}$$

$$\mathcal{DS}_4 = \{g_1, \ldots, g_l, g_{l+1}\}, \{g_1^{r_i}, \ldots, g_{i-1}^{r_i}, g_i^{r_i}, g_{i+1}^{r_i}, \ldots, g_l^{r_i}, g_{l+1}^{r_i}\}_{1 \leq i \leq l}, \{g_1^r, \ldots, g_l^r, g_{l+1}^r\}, x \tag{18}$$

$$\mathcal{DS}_5 = \{g_1, \ldots, g_l, sk \cdot \mathbf{g}\}, \{g_1^{r_i}, \ldots, g_l^{r_i}, (sk \cdot \mathbf{g})^{r_i}\}_{1 \leq i \leq l}, \{g_1^r, \ldots, g_l^r, (sk \cdot \mathbf{g})^r\}, x \tag{19}$$

We now briefly sketch how to derive the indistinguishability relations. The facts that $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$ and $\mathcal{DS}_3 \equiv^c \mathcal{DS}_4$ follow by Proposition 1, and the facts that $\mathcal{DS}_2 \equiv^c \mathcal{DS}_3$ and $\mathcal{DS}_4 \equiv^c \mathcal{DS}_5$ follows by considering that the inner product is a universal hash function and that it holds $\hat{H}_\infty(sk|x) \geq \log o + h$.

Finally, it is easy to verify that the second condition of strong-(λ, l)-circular security (i.e., Condition (b), Definition 5) holds for the BHHO scheme by considering the fact that the inner product, used in the key-generation algorithm, acts as universal hashing.

□

6.2 From the quadratic residuosity (QR) and related assumptions

Brakerski and Goldwasser [12] construct a circularly-secure encryption scheme (to which we refer as the BG scheme) from a general assumption that they call the *subgroup indistinguishability assumption*, which is in particular implied by the QR and Paillier's decisional composite residuosity (DCR) [28] assumptions. We show that the QR-based circularly-secure bit-encryption scheme of Brakerski and Goldwasser satisfies the reproducibility property; the analyses for the other schemes follow similarly.

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 $\mathbb{Z}_N^*$ consisting of quadratic residues modulo N, and let $\mathcal{J}_N$ denote the set of elements in $\mathbb{Z}_N^*$ with Jacobi symbol one. Finally, we define $\mathcal{QNR}_N = \mathcal{J}_N \setminus \mathcal{QR}_N$.

Assume that $RSAGen(1^k)$ is a PPT algorithm that on input 1^k generates a *Blum integer* N, i.e., $N = pq$ with p and q being distinct primes satisfying $p, q \equiv 3 \pmod 4$. We say that the *quadratic residuosity* (QR) problem is hard under RSAGen if $\{N, U(\mathcal{QR}_N)\}_{k\in\mathbb{N}}$ is computationally indistinguishable from $\{N, U(\mathcal{QNR}_N)\}_{k\in\mathbb{N}}$, where N is generated according to $RSAGen(1^k)$.

We now describe the BG scheme.

Definition 8. *(From [12])*

- $S(1^n)$*: returns* $(N, \mathbf{g})$*, where* $N \leftarrow RSAGen(1^n)$ *and* $\mathbf{g} \leftarrow \mathcal{QR}_N^l$*;*
- $Gen(1^n)$*: samples the secret key as* $sk \leftarrow \{0,1\}^l$ *and sets the public key* $pk = (sk \cdot \mathbf{g})^{-1}$*;*
- $E_{pk}(b))$*: samples* $r \in \mathbb{Z}_{N^2}$ *and returns* $(\mathbf{g}^r, pk^r \cdot (-1)^b)$*; and*
- $D_{sk}((\mathbf{g}^r, pk^r \cdot (-1)^b)$*: clear.*

The proof of reproducibility of the scheme above follows exactly as in the proof of the BHHO scheme. We omit the details. We also note that a similar statement to that of Theorem 9 may be given for the BG scheme.

7 Conclusions and open problems

We gave generic constructions of several cryptographic primitives based on a general technique for de-randomizing reproducible bit-encryption schemes. For all the primitives we built it is already known that a BB construction from CPA-secure encryption alone is either impossible, or very difficult to find. We mention two main open problems that arise from our work. First, it would be interesting to see if the BB result of [20] already separates TDFs from circularly-secure encryption; showing this would imply that our reliance on an additional property, i.e., reproducibility, is unavoidable. Second, we would like to see whether the LWE-based circularly-secure scheme of Appleabaum et al. [2] can be used to instantiate our base assumptions.

References

1. Benny Applebaum. Key-dependent message security: Generic amplification and completeness. In *Advances in Cryptology - EUROCRYPT 2011*.
2. 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*.
3. Boaz Barak, Iftach Haitner, Dennis Hofheinz, and Yuval Ishai. Bounded key-dependent message security. In *EUROCRYPT*, 2010.
4. Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, volume 4622 of *Lecture Notes in Computer Science*, pages 535–552. Springer, 2007.
5. 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.
6. Mihir Bellare, Marc Fischlin, Adam O'Neill, and Thomas Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In Wagner [34], pages 360–378.
7. Eleanor Birrell, Kai-Min Chung, Rafael Pass, and Sidharth Telang. Randomness-dependent message security. In Amit Sahai, editor, *Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*, volume 7785 of *Lecture Notes in Computer Science*, pages 700–720. Springer, 2013.
8. 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.

9. Alexandra Boldyreva, Serge Fehr, and Adam O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In Wagner [34], pages 335–359.
10. Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007.
11. Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In Wagner [34], pages 108–125.
12. 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.
13. Zvika Brakerski, Shafi Goldwasser, and Yael Tauman Kalai. Black-box circular-secure encryption beyond affine functions. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011.*
14. Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *LNCS*, pages 351–368. Springer, 2009.
15. Seung Geol Choi and Hoeteck Wee. Lossy trapdoor functions from homomorphic reproducible encryption. *Inf. Process. Lett.*, 112(20):794–798, October 2012.
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. David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More constructions of lossy and correlation-secure trapdoor functions. *J. Cryptology*, 26(1):39–74, 2013.
18. Benjamin Fuller, Adam O'Neill, and Leonid Reyzin. A unified approach to deterministic encryption: New constructions and a connection to computational entropy. In Ronald Cramer, editor, *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, volume 7194 of *Lecture Notes in Computer Science*, pages 582–599. Springer, 2012.
19. Yael Gertner, Tal Malkin, and Steven Myers. Towards a separation of semantic and CCA security for public key encryption. In Salil P. Vadhan, editor, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*, pages 434–455. Springer, 2007.
20. Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 126–135. IEEE Computer Society, 2001.
21. Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In Johnson [24], pages 25–32.
22. Dennis Hofheinz. Circular chosen-ciphertext security with compact ciphertexts. 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. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 520–536. Springer, 2013.
23. Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In Johnson [24], pages 44–61.
24. David S. Johnson, editor. *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*. ACM, 1989.
25. 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.
26. Steven Myers and Abhi Shelat. Bit encryption is complete. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 607–616. IEEE Computer Society, 2009.
27. Noam Nisan and David Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996.
28. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.
29. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Cynthia Dwork, editor, *STOC*, pages 187–196. ACM, 2008.
30. Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge,*

MA, USA, February 19-21, 2004, Proceedings, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2004.
31. Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. *SIAM J. Comput.*, 39(7):3058–3088, 2010.
32. Ron Rothblum. On the circular security of bit-encryption. In *TCC*, pages 579–598, 2013.
33. Yevgeniy Vahlis. Two is a crowd? A black-box separation of one-wayness and security under correlated inputs. In Daniele Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*, pages 165–182. Springer, 2010.
34. David Wagner, editor. *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*. Springer, 2008.
35. Hoeteck Wee. Dual projective hashing and its applications – lossy trapdoor functions and more. In *Proceedings of the 31st Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'12, pages 246–262, Berlin, Heidelberg, 2012. Springer-Verlag.