Reproducible Circularly-Secure Bit Encryption: Applications and Realizations*

Mohammad Hajiabadi§ Bruce M. Kapron†

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), chosen-ciphertext-attack (CCA) 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 decisional Diffie-Hellman (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 products of large primes), or refer to the hardness of some *general* problem (e.g., the existence of one-way 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* [6]. 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

*An extended abstract of this work was published in the proceedings of Crypto 2015. Work supported in part by the NSERC Discovery Grant "Foundational Studies in Privacy and Security."

§Department of Computer Science, University College London, UK. Work completed while a PhD student at the University of Victoria. Email: M.Hajiabadi@cs.ucl.ac.uk.

†Department of Computer Science, University of Victoria, Victoria, BC, Canada. Email: bmkapron@uvic.ca.

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 [12,13,3], and show if the base scheme is k-rec circularly secure, the constructed TDF is k-wise one-way, in the sense of [37].
2. We show how to extract many hardcore bits for our constructed TDFs, and by applying the results of [37] we obtain a blackbox construction of CCA2-secure encryption from our assumptions. Our CCA2 construction is *non-shielding* in the sense of [24]. We partially justify this fact by showing with respect to a weaker encryption primitive than ours, a non-shielding blackbox CCA2 construction is possible, while a shielding CCA2 construction is blackbox impossible.
3. By slightly extending our base primitive, we show how to obtain deterministic encryption schemes secure under *block-source* inputs, as defined by [10].
4. We realize our base encryption primitive by showing the circularly-secure schemes of [12,13] 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 (for encryption) $\{0,1\}^\rho$ 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. Also, we use $E_{pk}(b;r)$ to denote encryption of bit b under randomness r.

Trapdoor functions. Central to public-key cryptography is the notion of *injective trapdoor one-way functions*, 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. [25], there is a blackbox 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 [35]).

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 [25], 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 a random $r = r_1 \dots r_\rho \in \{0,1\}^\rho$ from

$$(E_{pk_1}(r_1;r), \dots, E_{pk_\rho}(r_\rho;r)),$$

where all pk_i's are chosen at random. Note that this immediately yields a TDF.

To address this question we first review a property of PKE schemes, called *reproducibility* [6]: $\mathcal{E} = (Gen, E, D)$ 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 $R(c, m_1, sk_1)$. 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:

- Gen' samples (pk', sk'), where

$$sk' = r \text{ and } pk' = c = E_{pk}(0; r)$$

  That is, the secret key is a (random) randomness string r and the public key is a dummy $\mathcal{E}$-ciphertext formed under randomness 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 for which $E_{pk_1}(b; r) = c'$.

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 problem is exactly that of *circular security*, a special case of the well-studied problem of *key-dependent-message* security [9,12,4,3,13,30,2,14]. 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 a trapdoor function $\mathcal{F} = C(\mathcal{E}) = (G, F, F^{-1})$, where G, F and F^{-1}, are respectively the key-generation, evaluation and inversion algorithms as follows. (See Section 2.2 for formal definitions and notation.) 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 satisfying $c'_i = E_{pk'}(b_i; r_i)$.

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 non-negligible 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* [37] if one-wayness holds even if the given input is evaluated under k independently chosen functions.* More formally, we say that $\mathcal{F} = (G, F, F^{-1})$ is 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 [37] 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, generalizing a prior construction [35] 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 have so far been based on LTDFs. Indeed, as shown by Vahlis [39], even 2-wise one-way TDFs cannot be constructed in a blackbox way from trapdoor permutations (TDPs).

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

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 [37]: the security reduction of [37] involves both statistical and computational arguments, allowing one to obtain only k-wise one-way TDFs for an *a priori* fixed but arbitrarily large value of k (which does suffice for CCA2 encryption) from sufficiently lossy TDFs. Our reduction 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 are 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 [38, Theorem 8]. For the rest of the introduction, however, for ease of exposition, we describe the results with respect to full circular security.

We extend Construction C to the case in which the base scheme is t-circularly secure (i.e., circularly-secure with respect to 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 [26] 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 blackbox-stronger (by Vahlis' 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 variants of our basic constructed TDFs. The first method applies

*Actually, [37] chose another name for this particular notion, but we refer to it as k-wise one-wayness for simplicity.

to t-circular security and allows us to extract $\log((t-1)!)$ bits, with the advantage that it only increases the domain size of the basic TDF. 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 that 3-circular security implies that 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.[§] 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 remains pseudorandom even knowing $F(ik, (pk, sk, x))$. Finally, assuming the property that public keys under the base scheme are computed 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,[†] we may now use the blackbox construction of Rosen and Segev [37] to build a many-bit CCA2-secure PKE from a reproducible, circularly secure bit-encryption scheme. Specifically, [37] gives a blackbox 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 [37], result in schemes whose decryption functions query the encryption function of the base scheme. Gertner, Malkin and Myers [24] refer to such constructions as *non-shielding*, and show that there exists no *shielding* blackbox construction of CCA-secure from CPA-secure encryption. Since our base assumptions are blackbox-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 [37], 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 with respect to a weaker primitive than ours, which is a special case of *randomness-dependent-message-secure (RDMS)* encryption [8], which allows secure multiple bitwise-encryptions of a randomness string r under r itself as randomness (Formalized in Definition 7). 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 blackbox impossibility result of [24] extends even if the base scheme is an RDMS encryption primitive (The-

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

[†]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 4.

orem 6). Indeed, it seems that this latter statement is true for most encryption primitives whose security requirements are defined with respect to passive indistinguishability (i.e., no decryption oracles); see Section 5.1 for more details. Thus, we obtain an encryption primitive, with respect to which a non-shielding blackbox CCA-secure construction is possible, but under which a shielding CCA-secure construction is blackbox impossible.

Deterministic encryption (DE). Following [10], 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 that 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.[‡]

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 that, by appropriately choosing the parameters, the schemes of [12,13] provide strong-entropy circular security, meaning that 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 [10] and the scheme of Boneh et al. [12]. 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., [40], to obtain DE schemes.

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 [7,5,10] and formalized in [23]. 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 its own associated hardcore bitstring under a randomized encryption scheme $\mathcal{E}'$: that is, $E_{ik,pk}(x) = E'_{pk}(F(ik, x), h(x))$; 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 $h(x)$.

1.1 Further discussion

The possibility of obtaining lossy trapdoor functions. Since LTDFs [35] 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. (See [37, Theorem 3.3].) On the other hand, LTDFs

[‡]The notion of weak-entropy circular security was also considered by [14] in the context of KDM amplification.

have powerful statistical properties (i.e., losing information in lossy mode) which do not seem to be realizable under our assumptions. In particular, we were not able to define "lossy" keys (in the sense of [35]) under our constructions; those lossy keys should be vectors of encryptions under the base scheme (as in injective keys) in such a way that when one applies the reproduction function to them (as in the evaluation algorithm) this results in loss of information. This idea does not seem to be implementable without making additional assumptions. The work of Hemenway and Ostrovsky [28] shows how to build LTDFs from a form of *lossy encryption*. It might be possible to obtain LTDFs by formulating and assuming an appropriate form of lossy encryption in our setting; we have not, however, investigated this direction.

Comparison with [19]. Choi and Wee [19], by abstracting the DDH-based TDF construction of Peikert and Waters [35], show how to obtain LTDFs from reproducible encryption that is homomorphic with respect to both messages and randomness. In what comes below we first compare our construction to that of Choi and Wee and then compare our underlying assumptions.

The construction of [19] results in (a) public keys that consist of $(\log |Rand| + \omega(\log n))^2$ base-ciphertexts (i.e., ciphertexts under the base scheme) and (b) ciphertexts that consist of $(\log |Rand| + \omega(\log n))$ base-ciphertexts. (Here $Rand$ is the randomness space of the base encryption function.) Assuming $\log |Rand| \in \Theta(n)$ this translates into quadratically large public keys and linearly large ciphertexts. Under our basic TDF construction both constructed public keys and ciphertexts consist of $\log |SK|$ base-ciphertexts, where SK is the secret-key space of the base scheme. For a concrete comparison, DDH-based instantiations of [35,19,22] give us schemes whose public keys and ciphertexts contain, respectively, $\Theta(n^2)$ and $\Theta(n)$ group elements. On the other hand, the DDH-based circularly-secure scheme of Boneh et al. has ciphertexts with $\Theta(n)$ group elements and secret keys with $\Theta(n)$ bits. Thus, we obtain a DDH-based TDF with public keys and ciphertexts both consisting of $\Theta(n^2)$ group elements. (The size of ciphertexts can be cut down to $\Theta(n)$ by removing redundancies; see Construction 4.) Thus, we obtain no improvements in efficiency, despite the fact that our generic construction offers public keys and ciphertexts each containing a linear (in $\log |SK|$) number of base-ciphertexts. The same phenomenon also holds for concrete deterministic encryption schemes. However, our work shows that progress in improving the efficiency of BHHO might lead to improvements in efficiency of existing DDH-based TDFs or DE schemes.

Homomorphism versus circular security. The notions of homomorphism (in the sense of [19]) and circular security for an encryption scheme are qualitatively different as they concern structural versus security properties. Interestingly though, all constructions of circularly-secure schemes in the literature rely on certain homomorphic properties of their underlying algebraic assumptions [12,13,3]. However, it is not clear whether the existence of reproducible circularly-secure encryption implies that of reproducible, homomorphic encryption. (If such an implication is proved then all our results will be subsumed by [19], since LTDFs imply all primitives we build in this paper.) For one thing, a circularly-secure scheme by itself does not necessarily provide the homomorphic property of [19] (or even weaker forms thereof). For example, under widely-believed assumptions one may construct a CCA2-secure, circularly-secure scheme [17,29], but homomorphic properties for such a scheme violate CCA2-security. Moreover, it seems hard to construct a homomorphic encryption scheme starting from a reproducible, circularly-secure scheme that does not provide any homomorphic properties by itself.

Finally, as noted by Rosen and Segev [37], in light of their blackbox impossibility result separating LTDFs from k-wise TDFs, there may be generic assumptions that yield k-wise TDFs, but not LTDFs; we believe that our encryption primitive is an example of those.

Shielding versus non-shielding constructions. We note that almost all blackbox CCA2 constructions are non-shielding, e.g., [31,37,35], except for a few cases which rely on very powerful (and structurally different) primitives, e.g., [11].• Intuitively, the non-shielding property of those constructions is used to do consistency checks on ciphertexts, i.e., it allows a simulator, that typically does not have the entire decryption key, to ensure that a given ciphertext is indeed generated by the encryption algorithm. It would be interesting to explore if there exist weaker encryption primitives (than those we consider) for which the blackbox separation of [24] is the best possible.

Non-bit encryption case. We informally discuss adaptations of Construction $C(\mathcal{E})$ to the case in which the secret-key space of $\mathcal{E}$ is a subset of its plaintext space Msg (which allows the secret key to be encrypted as a whole) and that reproducibility holds with respect to Msg. Let SK be the secret key space of $\mathcal{E}$. For this case we may substantially improve efficiency by having each index key contain only one $\mathcal{E}$-ciphertext, whose underlying randomness will be reused to self-encrypt the secret key $sk \in SK$ 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 $SK \subseteq \mathcal{M}$) circularly secure (again when the entire secret key is encrypted at once). This transformation, however, does not (necessarily) maintain this last, inversion-needed property. Thus, our results suggest that the CPA-to-one-shot-circular transformation may be non-trivial (and interesting) if it is to maintain this last mentioned property.

2 Preliminaries

2.1 Notation and basic definitions

For a finite set S we use $x \leftarrow S$ to denote sampling x uniformly at random from S and denote by $Unif_S$ the uniform distribution on S. If $\mathcal{D}$ is a distribution then $x \leftarrow \mathcal{D}$ denotes choosing x according to $\mathcal{D}$. 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 $\mathcal{D}$ by $Sup(\mathcal{D})$, and write $x \in \mathcal{D}$ to indicate $x \in Sup(\mathcal{D})$. 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 algorithm 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 $x \in \{0,1\}^*$ we use $|x|$ to denote the bit length of x and use x_i, for $1 \le i \le |x|$, to denote the ith bit of x.

2.2 Trapdoor functions and various one-wayness conditions

In this subsection we review the standard notion of injective trapdoor functions, the notion of hardcore functions and various one-wayness conditions.

•The concepts of shielding/non-shielding only apply to encryption or TDF based constructions; see Definition 6.

In the following definitions, let $D = \{D_n\}$ be an ensemble of sets, $\mathcal{D}_n$ be a distribution over D_n and $\mathcal{D} = \{\mathcal{D}_n\}$.

Definition 1. *(one-way injective trapdoor functions) A D-domain* collection of injective trapdoor functions *(TDFs)*[‖] *is given by three algorithms $\mathcal{F} = (G, F, F^{-1})$ as follows. The probabilistic algorithm $G(1^n)$ randomly produces a pair (ik, tk) of* injective/trapdoor keys*; the deterministic algorithm $F(ik, \cdot)$ given $x \in D_n$ produces an* image *$y = F(ik, x)$; and $F^{-1}(tk, \cdot)$ given an image y returns a pre-image x. We require $\mathcal{F}$ satisfy the correctness condition stating*

$$\Pr\left[F^{-1}(tk, F(ik, x)) = x\right] = 1,$$

where the probability is taken over the choices of $(ik, tk) \leftarrow G(1^n)$ and $x \leftarrow D_n$. We stress that the input domain of $F(ik, \cdot)$ only depends on the security parameter 1^n. We use the notation $Domain(F)$ to refer to $D = \{D_n\}$.

We call $\mathcal{F}$ $\mathcal{D}$-one-way if if for any adversary $\mathcal{A}$,

$$\Pr\left[\mathcal{A}(ik, F(ik, x)) = x\right] = negl(n),$$

where the probability is taken over the choices of $(ik, tk) \leftarrow G(1^n)$, $x \leftarrow \mathcal{D}_n$ and $\mathcal{A}$'s coins.

Definition 2. *(k-wise TDF products and k-wise one-wayness [37]) The k-wise product of a D-domain TDF $\mathcal{F} = (G, F, F^{-1})$ is a D-domain TDF $\mathcal{F}^{(k)} = (G^{(k)}, F^{(k)}, F^{-1^{(k)}})$ constructed as follows. The algorithm $G^{(k)}(1^n)$ first samples*

$$(ik_1, tk_1), \ldots, (ik_k, tk_k) \leftarrow G(1^n),$$

and lets $(ik_1, \ldots, ik_k)$ be the index key and $(tk_1, \ldots, tk_k)$ be the trapdoor key. On input $x \in D_n$, $F^{(k)}((ik_1, \ldots, ik_k), \cdot)$ returns $(F(ik_1, x), \ldots, F(ik_k, x))$. Finally, $F^{-1^{(k)}}$ is defined as

$$F^{-1^{(k)}}((tk_1, \ldots, tk_k), y) = F^{-1}(tk_1, y).$$

We say that $\mathcal{F}$ is k-wise $\mathcal{D}$-one-way if $\mathcal{F}^{(k)}$ is $\mathcal{D}$-one-way.

Note that 1-wise $\mathcal{D}$-one-wayness is the standard notion of $\mathcal{D}$-one-wayness defined in Definition 1.

Definition 3. *Let $\mathcal{F} = (G, F, F^{-1})$ be a D-domain TDF and $h = \{h_n\}$ be an ensemble of deterministic functions where $h_n : D_n \to \{0,1\}^{p(n)}$ (for some polynomial p). We say that h is a* $\mathcal{D}$-hardcore function *for $\mathcal{F}$ if for any adversary $\mathcal{A}$,*

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

where $(ik, tk) \leftarrow G(1^n)$ and $x \leftarrow \mathcal{D}_n$.

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

2.3 Definitions related to encryption schemes

All encryption schemes that appear throughout, unless otherwise stated, are *bit-encryption* schemes. In our applications we need to work with a more general notion of encryption schemes involving *public parameters*, as formalized next.

A bit-encryption scheme $\mathcal{E} = (Param, Gen, E, Dec)$ is defined as follows. The *parameter-generation algorithm* $Param$ 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 an input to other algorithms. The *encryption algorithm* E on inputs 1^n, public key pk, bit b and randomness $r \in Rand_n$, outputs a 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

$$\Pr\left[Dec_{sk}(E_{pk}(b)) = b\right] = 1,$$

where $par \leftarrow Param(1^n)$, $(pk, sk) \leftarrow Gen(1^n, par)$ and $b \leftarrow \{0,1\}$. We will typically use $Rand = \{Rand_n\}$ to denote the underlying randomness space of the encryption algorithm of a scheme under consideration.

Assumption 1 *Throughout this paper we make the following two assumptions about any encryption scheme $\mathcal{E} = (Param, Gen, E, Dec)$ under consideration.*

1. *For any n and any $par \in Param(1^n)$, all secret keys output by $Gen(1^n)$ are bitstrings of the same length. Thus, we have an associated secret-key-length function, usually denoted by l, which is a function of the security parameter.*
2. *In all security definitions that involve generating many public keys (e.g., multiple-key based security definitions) we assume all the underlying keys are sampled with respect to a fixed, random par sampled once and for all at the beginning of the underlying game.*

Given the assumptions above, henceforth we typically omit the inclusion of $Param$.

We now review the definition of *chosen-plaintext-attack security* and introduce different flavors of the notion of *circular security* [18,12]. As notation, for $m \in \{0,1\}^*$ we extend E to define $E_{pk}(m) = (E_{pk}(m_1), \ldots, E_{pk}(m_{|m|}))$. Also, for $\mathbf{r} = (r_1, \ldots, r_t)$ and $m \in \{0,1\}^t$ we write

$$E_{pk}(m; \mathbf{r}) = (E_{pk}(m_1; r_1), \ldots, E_{pk}(m_t; r_t)).$$

Definition 4. *For an encryption scheme $\mathcal{E} = (Gen, E, Dec)$ we define the following notions.*

1. *We say $\mathcal{E} = (Gen, E, Dec)$ is chosen-plaintext-attack (CPA) secure if*

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

   where $(pk, sk) \leftarrow Gen(1^n)$.
2. *We say $\mathcal{E} = (Gen, E, Dec)$ is k-rec t-circularly secure if for every adversary $\mathcal{A}$,*

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

   where

$$(pk_1, sk_1), \ldots, (pk_t, sk_t) \leftarrow Gen(1^n)$$

   and for every $1 \le i \le k$

$$\mathbf{c}_i \leftarrow \left(E_{pk_1}(sk_2), \ldots, E_{pk_{t-1}}(sk_t), E_{pk_t}(sk_1)\right).$$

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

$$(pk_1, \ldots, pk_t, \mathbf{c}_1, \ldots, \mathbf{c}_k) \equiv^c (pk_1, \ldots, pk_t, \mathbf{c}'_1, \ldots, \mathbf{c}'_k),$$

where $\mathbf{c}_i$'s are generated as above and for $1 \leq i \leq k$,

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

where $l = |sk_1|$. Note that we add CPA security as a separate condition because otherwise the definition may be satisfied trivially, e.g., consider an encryption scheme under which the secret key is always the all-zero string and the encryption function is the identity function. Also, we stress that in Cases 2 and 3 above all the underlying keys are generated with respect to a fixed, random parameter. (See Case 2 of Assumption 1.)

We adopt the following terminology convention in the paper.

Convention 1 *We use the terminology k-rec circular security (resp., k-ind circular security) as abbreviations for k-rec 1-circularly security (resp., k-ind 1-circularly security).*

We now review the notion of *reproducibility* of an encryption scheme, as defined in [6].

Definition 5. *We call $\mathcal{E} = (Gen, E, Dec)$* reproducible *if there exists a deterministic function R, called the reproduction function, such that for any n, any (pk_1, sk_1), $(pk_2, sk_2) \in Gen(1^n)$, any $r \in Rand_n$ and any $b_1, b_2 \in \{0,1\}$,*

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

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

3 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 tailor our construction to the t-circular security case (i.e., circular security with respect to t pairs of public/secret keys), meaning that we will obtain guarantees of one-wayness for the constructed TDF from t-circular security assumptions.

We first introduce the following pieces of notation. 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 *The following construction, that we call C_1, takes as input a reproducible bit-encryption scheme $\mathcal{E} = (Gen, E, Dec, R)$ and integer $t = t(n)$, and generates 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 key output by $Gen(1^n)$, which is well-defined by Case 1 of Assumption 1. Also, we denote the randomness space of E by $Rand = \{Rand_n\}$.*

– $G(1^n)$: *Sample an injective/trapdoor key* (ik, tk) *as follows. Choose* $(pk, sk) \leftarrow Gen(1^n)$*, and for* $1 \le i \le l$ *and* $0 \le j \le t-1$ *choose* $r_i^j \leftarrow Rand_n$*. Now let*

$$tk = (r_1^0, \ldots, r_l^0, \ldots, r_1^{t-1}, \ldots, r_l^{t-1}), \text{ and}$$
$$ik = (pk, c_1^0, \ldots, c_l^0, \ldots, c_1^{t-1}, \ldots, c_l^{t-1}),$$

where for $1 \le i \le l$ *and* $0 \le j \le t-1$*, we set* $c_i^j = E_{pk}(0; r_i^j)$.

– F: *On an injective key*

$$ik = (pk, c_1^0, \ldots, c_l^0, \ldots, c_1^{t-1}, \ldots, c_l^{t-1})$$

and domain point

$$x = (pk_0, sk_0, \ldots, pk_{t-1}, sk_{t-1})$$

return

$$F(ik, x) = (pk_0, \ldots, pk_{t-1}, c'^0_1, \ldots, c'^0_l, \ldots, c'^{t-1}_1, \ldots, c'^{t-1}_l),$$

where, denoting by b_i^j *the* i*th bit of* $sk_{j+1 \bmod t}$*, we define*

$$c'^j_i = R(c_i^j, b_i^j, sk_j). \tag{1}$$

– F^{-1}: *On a trapdoor key*

$$(r_1^0, \ldots, r_l^0, \ldots, r_1^{t-1}, \ldots, r_l^{t-1})$$

and image point

$$(pk_0, \ldots, pk_{t-1}, c'^0_1, \ldots, c'^0_l, \ldots, c'^{t-1}_1, \ldots, c'^{t-1}_l)$$

retrieve the corresponding pre-image

$$(pk_0, sk_0, \ldots, pk_{t-1}, sk_{t-1})$$

as follows: For $0 \le j \le t-1$ *recover* sk_j *bit-by-bit by encrypting back both* 0 *and* 1 *with the "corresponding" provided randomness (and under the corresponding public key) and finding the matching bit.*

Completeness of the constructed TDF 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 randomness known to the inverter. Second, the construction is entirely blackbox, also accessing (during evaluation) the reproduction function. Third, the construction extends to the non-bit-encryption case (i.e., when the base scheme is a reproducible scheme but not a bit-encryption scheme), by fixing a mapping from bits to two fixed plaintext messages and still continuing to encrypt the input secret key bit-by-bit using the reproduction function during the evaluation algorithm, but actually encrypting the plaintext message each bit is mapped to. For this case, the one-wayness of the constructed TDF reduces to bit-wise circular security of the base scheme (with respect to 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., with respect to a single pair of public/secret keys. The proof for the general case follows similarly. Thus, in the following we have $\mathcal{D} = Gen(1^n)$.

Recall that we use $l = l(n)$ to denote the length of a secret key outout by $Gen(1^n)$. Also, recall the following notation defined earlier. 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 1-circularly secure and $\mathcal{A}$ is an adversary against the k-wise $\mathcal{D}$-one-wayness of $\mathcal{F}$, achieving advantage ϵ, namely

$$\Pr\left[\mathcal{A}\left(\underbrace{pk', E_{pk'}(sk'; \mathbf{r}_1), \ldots, E_{pk'}(sk'; \mathbf{r}_k)}_{\mathbf{image}}, \underbrace{pk_1, E_{pk_1}(0^l; \mathbf{r}_1), \ldots, pk_k, E_{pk_k}(0^l; \mathbf{r}_k)}_{\mathbf{ik}}\right) = (pk', sk')\right] = \epsilon(n), \quad (2)$$

where $(pk', sk'), (pk_1, sk_1), \ldots, (pk_k, sk_k) \leftarrow Gen(1^n)$ and $\mathbf{r}_1, \ldots, \mathbf{r}_k \leftarrow Rand_n^l$. Notice that **ik** contains concatenations of k independent injective keys under $\mathcal{F}$ and **image** contains concatenations of the images of a random domain input, (pk', sk'), under the k injective keys. We first note that if $\mathcal{A}$ were only given **image**, it could perfectly generate **ik** by itself, by sampling

$$(pk_1, sk_1), \ldots, (pk_k, sk_k) \leftarrow Gen(1^n)$$

and appropriately using the reproduction function R to build **ik**. 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.

($\Leftarrow$) This direction 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

$$\Big(pk', E_{pk'}(sk'; \mathbf{r}_1), \ldots, E_{pk'}(sk'; \mathbf{r}_k), pk_1, E_{pk_1}(0^l; \mathbf{r}_1), \ldots, pk_k, E_{pk_k}(0^l; \mathbf{r}_k)\Big),$$

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

We conclude this subsection with the following observation about the structure of the TDF given in Construction 1.

Remark 1. We call a D-domain TDF *certifiable* if membership in D is efficiently decidable. A drawback of Our TDF in Construction 1 (and all those that appear henceforth) is that the TDF is not in general certifiable, since for a given encryption scheme (Gen, E, Dec) checking whether $(pk, sk) \in Gen(1^n)$ is not necessarily efficiently decidable. (Doing the standard test of encrypting many bits under pk and decrypting them back under sk and looking for matches only gives us a necessary condition.) It remains open to determine whether under our assumptions a certifiable TDF can be built.

4 Extracting many hardcore bits

We present two deterministic methods for extracting many hardcore bits from variants of the TDF presented in Construction 1, with tight reductions to the indistinguishability variants of circular security assumptions. The first method applies to t-circular security for $t \geq 3$, allowing us to directly extract $\log((t-1)!)$ bits, by expanding only the domain space (of the TDF of Construction 1) by the same number of bits (but without affecting the sizes of the system's other 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; this results in 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.

4.1 First hardcore extraction method

We begin with some notation. For integer $t > 0$ define $[t] = \{0, \ldots, t-1\}$. Also, for a set X we define

$$f(X) = \{f(x) \colon x \in X\}.$$

Let S contain all permutations f on $[t]$ where f induces only one *cycle*: $X \subseteq [t]$ is called a cycle under f if $X \neq \emptyset$ and $f(X) = X$. Note that $[t]$ is always a cycle under any permutation f over $[t]$, and thus a one-cycle permutation is one that has only the trivial cycle. Formally,

$$S = \big\{f \colon [t] \to [t] \mid f \text{ is injective and } \ \forall X \subsetneq [t], X \text{ is not a cycle under } f\big\}. \tag{3}$$

Note that

$$|S| = (t-1)!.$$

Intuitively, each $f \in S$ defines a possible circular ordering of encrypting a sequence of t pairs of public/secret keys, by having pk_i encrypt $sk_{f(i)}$. The one-cycle property 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

$$\mathcal{O} : [(t-1)!] \to S$$

to be an efficient index function defined using a canonical ordering of the elements of S. We assume the following notation about $\mathcal{O}$.

Notation 1 *We write $\mathcal{O}(i, x)$ to denote $f_i(x)$, where f_i is the ith permutation according to the ordering fixed on S. We also require that, for any $f \in S$, given*

$$sq = \{(0, f(0)), \ldots, (t-1, f(t-1))\},$$

it is possible to efficiently compute the index of f according to the ordering, 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.

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 $(D^t, [(t-1)!])$. Again, let $l = l(n)$ be the secret-key-length function of $\mathcal{E}$ and $Rand = \{Rand_n\}$ be the randomness space of E.*

- *$G(1^n)$: As in Construction 1.*
- *F: On an injective key*
$$ik = (pk, c_1^0, \ldots, c_l^0, \ldots, c_1^{t-1}, \ldots, c_l^{t-1})$$
and domain point
$$x = (pk_0, sk_0, \ldots, pk_{t-1}, sk_{t-1}, u)$$
do the following. First, set the indices
$$(ind_0, \ldots, ind_{t-1}) = (\mathcal{O}(u, 0), \ldots, \mathcal{O}(u, t-1)).$$
Informally, the output will be $pk_0, \ldots, pk_{t-1}$ together with a chain of encryptions, where pk_j encrypts the bits of sk_{ind_j}. Formally, return the tuple
$$(pk_0, \ldots, pk_{t-1}, c'^0_1, \ldots, c'^0_l, \ldots, c'^{t-1}_1, \ldots, c'^{t-1}_l),$$
where, for $0 \le j \le t-1$ and $1 \le i \le l$, denoting by b_i^j the ith bit of sk_{ind_j}, we set
$$c'^j_i = R(c_i^j, b_i^j, sk_j).$$
- *F^{-1} : On a trapdoor key*
$$(r_1^0, \ldots, r_l^0, \ldots, r_1^{t-1}, \ldots, r_l^{t-1})$$
and image point
$$(pk_0, \ldots, pk_{t-1}, c'^0_1, \ldots, c'^0_l, \ldots, c'^{t-1}_1, \ldots, c'^{t-1}_l)$$
do the following steps:
  - *recover $(x_0, \ldots, x_{t-1})$, where $x_j \in \{0,1\}^l$ for all j, as follows: to retrieve the ith bit of x_j for $1 \le i \le l$, encrypt both 0 and 1 under pk_j using randomness r_i^j and check the result against c'^j_i;*
  - *for each $0 \le j \le t-1$ let ind_j be the index for which it holds that pk_{ind_j} is the matching public key of x_j,[¶] and let $sk_{ind_j} = x_j$. Form $sq = \{(0, ind_0), \ldots, (t-1, ind_{t-1})\}$; return*
$$(pk_0, sk_0, \ldots, pk_{t-1}, sk_{t-1}, \mathcal{O}^{-1}(sq)).$$

Hardcore function: *For $\mathcal{F}$ given above we define $h\colon \left(D^t, [(t-1)!]\right) \to [(t-1)!]$ as*
$$h(pk_0, sk_0, \ldots, pk_{t-1}, sk_{t-1}, u) = u.$$

Correctness of the new TDF follows immediately. Note that Construction 1 is a special case of Construction 2, by forming the encrypted cycle with respect to the fixed permutation f defined as
$$f(j) = (j + 1 \mod t).$$

[¶]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.

In contrast, Construction 2 forms the encrypted cycle according to a permutation $f \in S$ provided as input to the evaluation algorithm, where, as we show below, a random choice of f is what is computationally hidden by the output of the evaluation algorithm.

As a main step toward proving that h is a hardcore function for $\mathcal{F}^{(k)}$ we show that the following two distributions are computationally indistinguishable:

$$\mathcal{D} = \left(pk_0, \ldots, pk_{t-1}, \underbrace{E_{pk_0}(sk_{f(0)}), \ldots, E_{pk_{t-1}}(sk_{f(t-1)})}_{\mathbf{1st}}, \ldots, \underbrace{E_{pk_0}(sk_{f(0)}), \ldots, E_{pk_{t-1}}(sk_{f(t-1)})}_{\mathbf{kth}} \right),$$

$$\mathcal{D}' = \left(pk_0, \ldots, pk_{t-1}, \underbrace{E_{pk_0}(0^l), \ldots, E_{pk_{t-1}}(0^l)}_{\mathbf{1st}}, \ldots, \underbrace{E_{pk_0}(0^l), \ldots, E_{pk_{t-1}}(0^l)}_{\mathbf{kth}} \right),$$

where $f \leftarrow S$ and all (pk_i, sk_i) are random pairs of public/secret keys. Here we use $\mathbf{1st}, \ldots, \mathbf{kth}$ to denote copies of the underlying distribution. If $f\colon [t] \mapsto [t]$ is fixed to $f(i) = (i+1 \mod t)$ then $\mathcal{D} \equiv^c \mathcal{D}'$ is exactly the notion of k-ind t-circular security. In what follows we show a tight reduction from distinguishing between $\mathcal{D}$ and $\mathcal{D}'$, for a random $f \in S$, to breaking the notion of k-ind t-circular security. The reduction itself, a more generalized version of that described in Lemma 2, is relatively easy, but its proof of correctness is quite tedious. We first need to establish the following lemma.

Lemma 1. *Let Compose denote a transformation taking a function $f\colon [t] \mapsto [t]$ to another function $g =^{def} Compose(f)\colon [t] \mapsto [t]$, defined as*

$$g(i) = f^{(i)}(0),$$

where we define $f^{(0)}(n) = n$ and

$$f^{(i)}(n) = \underbrace{f(f(\ldots f}_{i}(n)\ldots)).$$

Letting $+$ denote addition modulo t, we then have:

1. *If $f \in S$, then $g = Compose(f)$ is one-to-one. (Recall that the set S was defined in Paeg 14.)*
2. *For distinct $f_1, f_2 \in S$, defining $g_1 = Compose(f_1)$ and $g_2 = Compose(f_2)$, we have $g_1 \neq g_2$.*
3. *Define a transformation $Permute(\cdot)$ that transforms $f \in S$ to $h\colon [t] \mapsto [t]$ as*

   $$h(i) = g^{-1}(g(i) + 1),$$

   where $g = Compose(f)$. For any $f \in S$ we have $Permute(f) \in S$. Moreover, for any distinct $f_1, f_2 \in S$ we have $Permute(f_1) \neq Permute(f_2)$.
4. *The two distributions $Permute(Unif_S)$ and $Unif_S$ are identically distributed.*

Proof. Note that Item 4 follows from Item 3, so we prove Items 1, 2 and 3.

Item 1 Suppose for some $f \in S$, $g = Compose(f)$ is not one-to-one, namely for some $0 \leq i < j \leq t-1$,

$$f^{(i)}(0) = f^{(j)}(0).$$

Define

$$X = \{f^{(i)}(0), f^{(i+1)}(0), \ldots, f^{(j-1)}(0)\}.$$

Note that $|X| = j - i < t$ and so $X \subsetneq [t]$. However, it is easy to see that $f(X) = X$, which is a contradiction to the assumption that $f \in S$.

Item 2 Suppose $f_1, f_2 \in S$, $f_1 \neq f_2$ and $g_1 = \mathit{Compose}(f_1)$ and $g_2 = \mathit{Compose}(f_2)$. Suppose toward a contradiction that $g_1 = g_2$, namely

$$f_1^{(i)}(0) = f_2^{(i)}(0) \text{ for all } i \in [t]. \tag{4}$$

Since $f_1 \neq f_2$ for some $x \in [t]$ we have $f_1(x) \neq f_2(x)$. Since $g_1 \colon [t] \mapsto [t]$ is one-to-one (proved in the previous item), we have for some i that $g_1(i) = x$ or equivalently

$$f_1^{(i)}(0) = x.$$

Now by Equation 4 we have $f_2^{(i)}(0) = x$. Thus,

$$f_1(x) = f_1(f_1^{(i)}(0)) = f_1^{(i+1)}(0) = f_2^{(i+1)}(0) = f_2(f_2^{(i)}(0)) = f_2(x),$$

which is a contradiction to the earlier assumption that $f_1(x) \neq f_2(x)$.

Item 3 Let $f \in S$, $g = \mathit{Compose}(f)$ and h be defined as

$$h(i) = g^{-1}(g(i) + 1).$$

We first show $h \in S$. Suppose toward a contradiction that for $X = \{x_1, \ldots, x_m\}$ it holds that $h(X) = X$, where $m < t$. Again we recall that $+$ below denotes addition modulo t. We have

$$\begin{aligned}\{x_1, \ldots, x_m\} = h(\{x_1, \ldots, x_m\}) &\Leftrightarrow \{x_1, \ldots, x_m\} = \{g^{-1}(g(x_1) + 1), \ldots, g^{-1}(g(x_m) + 1)\} \\ &\Leftrightarrow \{g(x_1), \ldots, g(x_m)\} = \{g(x_1) + 1, \ldots, g(x_m) + 1\} =^{\text{def}} X'.\end{aligned}$$

Assume without loss of generality that $0 \leq g(x_1) < \cdots < g(x_m) \leq t - 1$. Since, $g(x_m) + 1 \in X'$ and $g(x_m)$ is the maximum element of X' we obtain $g(x_m) = t - 1$ and as a result $g(x_1) = 0$. Also, since $g(x_1) + 1 \in X'$ we have $1 \in X'$ and so $g(x_2) = 1$. Continuing using this argument we obtain $g(x_m) = m - 1 < t - 1$. However, this contradicts the previously established fact that $g(x_m) = t - 1$.

We now show that for any two distinct $f_1, f_2 \in S$, $\mathit{Permute}(f_1) \neq \mathit{Permute}(f_2)$. Let $g_1 = \mathit{Compose}(f_1)$, $g_2 = \mathit{Compose}(f_2)$, $h_1 = \mathit{Permute}(f_1)$ and $h_2 = \mathit{Permute}(f_2)$. From the statement we just proved we deduce $h_1, h_2 \in S$. Also, since $f_1 \neq f_2$ by Item 2 we have $g_1 \neq g_2$ and as a result

$$g_1^{-1} \neq g_2^{-1}. \tag{5}$$

Suppose to the contrary that $h_1 = h_2$. Thus,

$$h_1^{(i)}(0) = h_2^{(i)}(0), \text{ for all } i \in [t].$$

On the other hand, we claim

$$\begin{aligned}h_1^{(i)}(0) &= g_1^{-1}(i) \\ h_2^{(i)}(0) &= g_2^{-1}(i),\end{aligned}$$

which imply

$$g_1^{-1}(i) = h_1^{(i)}(0) = h_2^{(i)}(0) = g_2^{-1}(i), \text{ for all } i \in [t]$$

which is a contradiction to Equation 5. We prove our claim for h_1 and the proof for h_2 is exactly the same. As the base case we have

$$h_1^{(0)}(0) \overset{\text{by definition}}{=} 0 = g_1^{-1}(0),$$

as desired. Now assume $h_1^{(i)}(0) = g_1^{-1}(i)$ for some $i < t-1$; we have

$$h_1^{(i+1)}(0) = h_1(h_1^{(i)}(0)) = h_1(g_1^{-1}(i)) = g_1^{-1}(g_1(g_1^{-1}(i)) + 1) = g_1^{-1}(i+1), \tag{6}$$

as claimed. □

Lemma 2. *Let $\mathcal{E} = (Gen, E, Dec)$ be an arbitrary encryption scheme with the secret-key-length function $l = l(n)$, and let $t = t(n)$ and $k = k(n)$ be two arbitrary polynomials. Consider the following distributions:*

$$\mathbf{Dis}_1 = \Big(pk_0, \ldots, pk_{t-1}, E_{pk_0}(sk_1; \mathbf{r}_0^1), \ldots, E_{pk_{t-1}}(sk_0; \mathbf{r}_{t-1}^1), \ldots, E_{pk_0}(sk_1; \mathbf{r}_0^k), \ldots, E_{pk_{t-1}}(sk_0; \mathbf{r}_{t-1}^k)\Big) \tag{7}$$

$$\mathbf{Dis}_2 = \Big(pk_0, \ldots, pk_{t-1}, E_{pk_0}(sk_{f(0)}; \mathbf{r}_0^1), \ldots, E_{pk_{t-1}}(sk_{f(t-1)}; \mathbf{r}_{t-1}^1), \ldots, E_{pk_0}(sk_{f(0)}; \mathbf{r}_0^k), \ldots, E_{pk_{t-1}}(sk_{f(t-1)}; \mathbf{r}_{t-1}^k), f\Big),$$

$$\mathbf{Dis}_3 = \Big(pk_0, \ldots, pk_{t-1}, E_{pk_0}(0^l; \mathbf{r}_0^1), \ldots, E_{pk_{t-1}}(0^l; \mathbf{r}_{t-1}^1), \ldots, E_{pk_0}(0^l; \mathbf{r}_0^k), \ldots, E_{pk_{t-1}}(0^l; \mathbf{r}_{t-1}^k)\Big),$$

$$\mathbf{Dis}_4 = \Big(pk_0, \ldots, pk_{t-1}, E_{pk_0}(0^l; \mathbf{r}_0^1), \ldots, E_{pk_{t-1}}(0^l; \mathbf{r}_{t-1}^1), \ldots, E_{pk_0}(0^l; \mathbf{r}_0^k), \ldots, E_{pk_{t-1}}(0^l; \mathbf{r}_{t-1}^k), f\Big),$$

where

$$\begin{aligned} (pk_0, sk_0), \ldots, (pk_{t-1}, sk_{t-1}) &\leftarrow Gen(1^n) \\ f &\leftarrow S \\ \mathbf{r}_0^1, \ldots, \mathbf{r}_{t-1}^1, \ldots, \mathbf{r}_0^k, \ldots, \mathbf{r}_{t-1}^k &\leftarrow Rand_n^l. \end{aligned} \tag{8}$$

There exists a randomized algorithm Convert satisfying the following two properties:

$$Convert(\mathbf{Dis}_1) \equiv \mathbf{Dis}_2$$
$$Convert(\mathbf{Dis}_3) \equiv \mathbf{Dis}_4$$

Moreover, if $\mathcal{E}$ is k-ind t-circularly secure then

$$\mathbf{Dis}_2 \equiv^c \mathbf{Dis}_4,$$

and the reduction is tight.

Proof. Note that the "moreover" part follows from the existence of $Convert$ with the stated properties, and thus in what follows we show how to construct $Convert$ with the stated properties.

For an arbitrary

$$\mathbf{out} = (pk_0, \ldots, pk_{t-1}, \mathbf{c}_0^1, \ldots, \mathbf{c}_{t-1}^1, \ldots, \mathbf{c}_0^k, \ldots, \mathbf{c}_{t-1}^k)$$

we show how $Convert(\mathbf{out})$ works.

Below we use the operator $+$ to mean addition modulo t.

- Sample $f \leftarrow S$, and define $g = Compose(f)$ and $h = Permute(f)$. (Recall that these two transformations were defined in Lemma 1.) That is,

$$\begin{aligned} g(i) &= f^{(i)}(0) \\ h(i) &= g^{-1}(g(i)+1). \end{aligned}$$

- return

$$(pk_{g(0)}, \ldots, pk_{g(t-1)}, \mathbf{c}_{g(0)}^1, \ldots, \mathbf{c}_{g(t-1)}^1, \ldots, \mathbf{c}_{g(0)}^k, \ldots, \mathbf{c}_{g(t-1)}^k, h).$$

Note that the procedure $Convert$ is PPT since all the functions f, g and h are efficiently computable. In particular, the domain of all these functions is $[t]$, which is poly-sized.

We now prove that $Convert$ provides the desired properties. In the following let

$$\begin{aligned} (pk_0, sk_0), \ldots, (pk_{t-1}, sk_{t-1}) &\leftarrow Gen(1^n) \\ f &\leftarrow S \\ g(i) &= f^{(i)}(0) \\ h(i) &= g^{-1}(g(i)+1). \end{aligned}$$

We have

$$\begin{aligned} Convert(\mathbf{Dis}_1) &\equiv Convert(pk_0, \ldots, pk_{t-1}, \underbrace{E_{pk_0}(sk_1), \ldots, E_{pk_{t-1}}(sk_0)}_{1st}, \ldots, \underbrace{E_{pk_0}(sk_1), \ldots, E_{pk_{t-1}}(sk_0)}_{kth}) \\ &\equiv \Big(pk_{g(0)}, \ldots, pk_{g(t-1)}, \underbrace{E_{pk_{g(0)}}(sk_{g(0)+1}), \ldots, E_{pk_{g(t-1)}}(sk_{g(t-1)+1})}_{1st}, \\ &\qquad \ldots, \underbrace{E_{pk_{g(0)}}(sk_{g(0)+1}), \ldots, E_{pk_{g(t-1)}}(sk_{g(t-1)+1})}_{kth}, h\Big) \end{aligned} \tag{9}$$

Now for $i \in [t]$ defining

$$(pk_i', sk_i') = (pk_{g(i)}, sk_{g(i)})$$

we may rewrite the distribution in Equation 9 as

$$\underbrace{(pk_0', \ldots, pk_{t-1}', \underbrace{E_{pk_0'}(sk_{h(0)}'), \ldots, E_{pk_{t-1}'}(sk_{h(t-1)}')}_{1st}, \ldots, \underbrace{E_{pk_0'}(sk_{h(0)}'), \ldots, E_{pk_{t-1}'}(sk_{h(t-1)}')}_{kth}, h)}_{\mathbf{out1}}. \tag{10}$$

In Equation 10 we used the fact that

$$sk'_{h(i)} = sk_{g(h(i))} = sk_{g(i)+1}.$$

Now since f is chosen uniformly at random from S, by Part 4 of Lemma 1 we have h is also uniformly distributed over S, and so **out1** is a random element according to distribution $\mathbf{Dis}_2$.

To show $Convert(\mathbf{Dis}_3) \equiv \mathbf{Dis}_4$ note that

$$Convert(\mathbf{Dis}_3) \equiv \Big(pk_{g(0)}, \ldots, pk_{g(t-1)}, \underbrace{E_{pk_{g(0)}}(0^l), \ldots, E_{pk_{g(t-1)}}(0^l)}_{1st}, \ldots, \underbrace{E_{pk_{g(0)}}(0^l), \ldots, E_{pk_{g(t-1)}}(0^l)}_{kth}, h\Big),$$

where all the variables are sampled as above. By Lemma 1 g is one-to-one and h is distributed uniformly over S, and so we obtain $Convert(\mathbf{Dis}_3) \equiv \mathbf{Dis}_4$, and the proof is complete. □

The following lemma is standard. We give a sketch of the proof for completeness.

Lemma 3. *Let $\mathcal{F} = (G, F, F^{-1})$ be a D-domain TDF and $h_n : D_n \to \{0,1\}^{p(n)}$ define an ensemble of deterministic functions (for some poly function p). Let $\mathcal{D}_n$ be a distribution over D_n, and let $\mathcal{D} = \{\mathcal{D}_n\}$. For any adversasry $\mathcal{A}$ achieving advantage $\epsilon = \epsilon(n)$ against the $\mathcal{D}$-one-wayness of $\mathcal{F}$, there exists an adversary $\mathcal{B}$ that*

$$\left|\Pr\left[\mathcal{B}(ik, F(ik,x), h(x)) = 1\right] - \Pr\left[\mathcal{B}(ik, F(ik,x), Unif_{\{0,1\}^{p(n)}}) = 1\right]\right| \geq \frac{\epsilon}{2},$$

where $(ik, tk) \leftarrow G(1^n)$, $x \leftarrow \mathcal{D}_n$ and $\mathcal{B}$'s random coins.

Proof. The adversary $\mathcal{B}(ik, y, u)$ works as follows: it runs $\mathcal{A}(ik, y)$ to receive x. If $F(ik, x) \neq y$ then $\mathcal{B}$ returns $b \leftarrow \{0,1\}$. If $F(ik, x) = y$, $\mathcal{B}$ returns 1 if $u = h(x)$, and returns 0 otherwise. The desired bound follows. □

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. By Lemma 3 it suffices to show that h is a hardcore function for $\mathcal{F}^{(k)}$. Proving that h is a hardcore function for $\mathcal{F}^{(k)}$ boils down to showing that $\mathcal{D}'_1 \equiv^c \mathcal{D}'_2$, where

$$\mathcal{D}'_1 = \Big(\underbrace{pk'_1, E_{pk'_1}(0^l; \mathbf{r}^1_0), \ldots, E_{pk'_1}(0^l; \mathbf{r}^1_{t-1})}_{\mathbf{ik_1}}, \ldots, \underbrace{pk'_k, E_{pk'_k}(0^l; \mathbf{r}^k_0), \ldots, E_{pk'_k}(0^l; \mathbf{r}^k_{t-1})}_{\mathbf{ik_k}}$$
$$\underbrace{pk_0, \ldots, pk_{t-1}, E_{pk_0}(sk_{f(0)}; \mathbf{r}^1_0), \ldots, E_{pk_{t-1}}(sk_{f(t-1)}; \mathbf{r}^1_{t-1})}_{\mathbf{im_1}}, \ldots,$$
$$\underbrace{pk_0, \ldots, pk_{t-1}, E_{pk_0}(sk_{f(0)}; \mathbf{r}^k_0), \ldots, E_{pk_{t-1}}(sk_{f(t-1)}; \mathbf{r}^k_{t-1})}_{\mathbf{im_k}}, f\Big),$$

$$\mathcal{D}'_2 = \Bigg(\underbrace{E_{pk'_1}(0^l;\mathbf{r}^1_0),\ldots,E_{pk'_1}(0^l;\mathbf{r}^1_{t-1})}_{\mathbf{ik_1}},\ldots,\underbrace{E_{pk'_k}(0^l;\mathbf{r}^k_0),\ldots,E_{pk'_k}(0^l;\mathbf{r}^k_{t-1})}_{\mathbf{ik_k}}$$
$$\underbrace{pk_0,\ldots,pk_{t-1},E_{pk_0}(sk_{f(0)};\mathbf{r}^1_0),\ldots,E_{pk_{t-1}}(sk_{f(t-1)};\mathbf{r}^1_{t-1})}_{\mathbf{im_1}},\ldots,$$
$$\underbrace{pk_0,\ldots,pk_{t-1},E_{pk_0}(sk_{f(0)};\mathbf{r}^k_0),\ldots,E_{pk_{t-1}}(sk_{f(t-1)};\mathbf{r}^k_{t-1})}_{\mathbf{im_k}},f'\Bigg),$$

where

$$\begin{aligned}(pk'_1,sk'_1),\ldots,(pk'_k,sk'_k),(pk_0,sk_0),\ldots,(pk_{t-1},sk_{t-1}) &\leftarrow Gen(1^n)\\ f,f' &\leftarrow S\\ \mathbf{r}^1_0,\ldots,\mathbf{r}^1_{t-1},\ldots,\mathbf{r}^k_0,\ldots,\mathbf{r}^k_{t-1} &\leftarrow Rand^l_n.\end{aligned} \tag{11}$$

Fix the above way of sampling variables in the following. Note that since $\mathcal{E}$ is reproducible, given only $(\mathbf{im_1},\ldots,\mathbf{im_k})$ one can perfectly simulate the rest, namely $(\mathbf{ik_1},\ldots,\mathbf{ik_k})$ is obtained by sampling $(pk'_1,sk'_1),\ldots,(pk'_k,sk'_k)$ and using the reproduction function. Thus, to prove $\mathcal{D}'_1 \equiv^c \mathcal{D}'_2$, it suffices to show

$$\mathcal{D}_1 \equiv^c \mathcal{D}_2, \tag{12}$$

where

$$\mathcal{D}_1 = \Big(pk_0,\ldots,pk_{t-1},E_{pk_0}(sk_{f(0)};\mathbf{r}^1_0),\ldots,E_{pk_{t-1}}(sk_{f(t-1)};\mathbf{r}^1_{t-1}),\ldots,$$
$$E_{pk_0}(sk_{f(0)};\mathbf{r}^k_0),\ldots,E_{pk_{t-1}}(sk_{f(t-1)};\mathbf{r}^k_{t-1}),f\Big),$$

$$\mathcal{D}_2 = \Big(pk_0,\ldots,pk_{t-1},E_{pk_0}(sk_{f(0)};\mathbf{r}^1_0),\ldots,E_{pk_{t-1}}(sk_{f(t-1)};\mathbf{r}^1_{t-1}),\ldots,$$
$$E_{pk_0}(sk_{f(0)};\mathbf{r}^k_0),\ldots,E_{pk_{t-1}}(sk_{f(t-1)};\mathbf{r}^k_{t-1}),f'\Big),$$

From Lemma 2 we have

$$\mathcal{D}_1 \equiv^c \mathcal{D}_3, \tag{13}$$

$$\mathcal{D}_3 = \Big(pk_0,\ldots,pk_{t-1},E_{pk_0}(0^l;\mathbf{r}^1_0),\ldots,E_{pk_{t-1}}(0^l;\mathbf{r}^1_{t-1}),\ldots,E_{pk_0}(0^l;\mathbf{r}^k_0),\ldots,E_{pk_{t-1}}(0^l;\mathbf{r}^k_{t-1}),f\Big).$$

Applying Lemma 2 again we obtain $\mathcal{D}_2 \equiv^c \mathcal{D}_3$. Thus, we have

$$\mathcal{D}_1 \equiv^c \mathcal{D}_3 \equiv^c \mathcal{D}_2,$$

as desired. □

4.2 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 $\mathcal{F} = (G, F, F^{-1})$ we build is $(D^t, \{0,1\}^m)$. Let $l = l(n)$ be the secret-key-length function of $\mathcal{E}$ and $Rand = \{Rand_n\}$ be the randomness space of E.*

- *$G(1^n)$: Sample an injective/trapdoor key (ik, tk) as follows. Choose $(pk, sk) \leftarrow Gen(1^n)$, and for $1 \le i \le l$ and $0 \le j \le t-1$ choose $r_i^j \leftarrow Rand_n$. Also, for every $1 \le i \le m$ choose $r_i \leftarrow Rand_n$. Now let*

$$tk = (r_1^0, \ldots, r_l^0, \ldots, r_1^{t-1}, \ldots, r_l^{t-1}, r_1, \ldots, r_m), \text{ and}$$
$$ik = (pk, c_1^0, \ldots, c_l^0, \ldots, c_1^{t-1}, \ldots, c_l^{t-1}, c_1, \ldots, c_m),$$

  where for $1 \le i \le l$ and $0 \le j \le t-1$, we set $c_i^j = E_{pk}(0; r_i^j)$, and for $1 \le i \le m$ we set $c_i = E_{pk}(0; r_i)$.
- *On an injective key*

$$ik = (pk, c_1^0, \ldots, c_l^0, \ldots, c_1^{t-1}, \ldots, c_l^{t-1}, c_1, \ldots, c_m)$$

  and domain point

$$x = (pk_0, sk_0, \ldots, pk_{t-1}, sk_{t-1}, u)$$

  return

$$F(ik, x) = (pk_0, \ldots, pk_{t-1}, {c'}_1^0, \ldots, {c'}_l^0, \ldots, {c'}_1^{t-1}, \ldots, {c'}_l^{t-1}, c'_1, \ldots, c'_m),$$

  where, denoting by b_i^j the ith bit of $sk_{(j+1 \bmod t)}$, we define

$${c'}_i^j = R(c_i^j, b_i^j, sk_j). \tag{14}$$

  Also, for $1 \le h \le m$ we define

$$c'_h = R(c_h, u_h, sk_0).$$

- *F^{-1}: as in prior constructions.*

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

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

Correctness of inversion is again clear 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. By Lemma 3 it suffices to show that h is a hardcore function for $\mathcal{F}^{(k)}$. To prove this we need to show $\mathcal{D}'_1 \equiv^c \mathcal{D}'_2$, where

$$\mathcal{D}'_1 = \Big(\underbrace{pk'_1, E_{pk'_1}(0^l;\mathbf{r}_0^1), \ldots, E_{pk'_1}(0^l;\mathbf{r}_{t-1}^1), E_{pk'_1}(0^m;\mathbf{r}_t^1)}_{\mathbf{ik_1}}, \ldots,$$
$$\underbrace{pk'_k, E_{pk'_k}(0^l;\mathbf{r}_0^k), \ldots, E_{pk'_k}(0^l;\mathbf{r}_{t-1}^k), E_{pk'_k}(0^m;\mathbf{r}_t^k)}_{\mathbf{ik_k}},$$
$$\underbrace{pk_0, \ldots, pk_{t-1}, E_{pk_0}(sk_1;\mathbf{r}_0^1), \ldots, E_{pk_{t-1}}(sk_0;\mathbf{r}_{t-1}^1), E_{pk_0}(u;\mathbf{r}_t^1)}_{\mathbf{im_1}}, \ldots,$$
$$\underbrace{pk_0, \ldots, pk_{t-1}, E_{pk_0}(sk_1;\mathbf{r}_0^k), \ldots, E_{pk_{t-1}}(sk_0;\mathbf{r}_{t-1}^k), E_{pk_0}(u;\mathbf{r}_t^k)}_{\mathbf{im_k}}, u \Big),$$

$$\mathcal{D}'_2 = \Big(\underbrace{pk'_1, E_{pk'_1}(0^l;\mathbf{r}_0^1), \ldots, E_{pk'_1}(0^l;\mathbf{r}_{t-1}^1), E_{pk'_1}(0^m;\mathbf{r}_t^1)}_{\mathbf{ik_1}}, \ldots,$$
$$\underbrace{pk'_k, E_{pk'_k}(0^l;\mathbf{r}_0^k), \ldots, E_{pk'_k}(0^l;\mathbf{r}_{t-1}^k), E_{pk'_k}(0^m;\mathbf{r}_t^k)}_{\mathbf{ik_k}},$$
$$\underbrace{pk_0, \ldots, pk_{t-1}, E_{pk_0}(sk_1;\mathbf{r}_0^1), \ldots, E_{pk_{t-1}}(sk_0;\mathbf{r}_{t-1}^1), E_{pk_0}(u;\mathbf{r}_t^1)}_{\mathbf{im_1}}, \ldots,$$
$$\underbrace{pk_0, \ldots, pk_{t-1}, E_{pk_0}(sk_1;\mathbf{r}_0^k), \ldots, E_{pk_{t-1}}(sk_0;\mathbf{r}_{t-1}^k), E_{pk_0}(u;\mathbf{r}_t^k)}_{\mathbf{im_k}}, u' \Big),$$

where

$$\begin{aligned} (pk'_1, sk'_1), \ldots, (pk'_k, sk'_k), (pk_0, sk_0), \ldots, (pk_{t-1}, sk_{t-1}) &\leftarrow Gen(1^n) \\ u, u' &\leftarrow \{0,1\}^m \\ \mathbf{r}_0^1, \ldots, \mathbf{r}_{t-1}^1, \ldots, \mathbf{r}_0^k, \ldots, \mathbf{r}_{t-1}^k &\leftarrow Rand_n^l \text{ and } \mathbf{r}_t^1, \ldots, \mathbf{r}_t^k \leftarrow Rand_n^m. \end{aligned} \tag{15}$$

Fix the above way of sampling variables in the following. Note that since $\mathcal{E}$ is reproducible, given only $(\mathbf{im_1}, \ldots, \mathbf{im_k})$ one can perfectly simulate the rest, namely $(\mathbf{ik_1}, \ldots, \mathbf{ik_k})$ is obtained by sampling $(pk'_1, sk'_1), \ldots, (pk'_k, sk'_k)$ and using the reproduction function. Thus, to prove $\mathcal{D}'_1 \equiv^c \mathcal{D}'_2$, it suffices to show $\mathcal{D}_1 \equiv^c \mathcal{D}_4$, where

$$\mathcal{D}_1 = \Big(pk_0, \ldots, pk_{t-1}, \underbrace{E_{pk_0}(sk_1), \ldots, E_{pk_{t-1}}(sk_0), E_{pk_0}(u)}_{1st}, \ldots, \underbrace{E_{pk_0}(sk_1), \ldots, E_{pk_{t-1}}(sk_0), E_{pk_0}(u)}_{kth}, u \Big),$$

$$\mathcal{D}_4 = \Big(pk_0, \ldots, pk_{t-1}, \underbrace{E_{pk_0}(sk_1), \ldots, E_{pk_{t-1}}(sk_0), E_{pk_0}(u)}_{1st}, \ldots, \underbrace{E_{pk_0}(sk_1), \ldots, E_{pk_{t-1}}(sk_0), E_{pk_0}(u)}_{kth}, u' \Big).$$

We now introduce $\mathcal{D}_2$ and $\mathcal{D}_3$ and show

$$\mathcal{D}_1 \equiv^c \mathcal{D}_2 \equiv^c \mathcal{D}_3 \equiv^c \mathcal{D}_4,$$

which will conclude the proof:

$$\mathcal{D}_2 = \Big(pk_0, \ldots, pk_{t-1}, \underbrace{E_{pk_0}(0^l), \ldots, E_{pk_{t-1}}(0^l), E_{pk_0}(u)}_{1st}, \ldots, \underbrace{E_{pk_0}(0^l), \ldots, E_{pk_{t-1}}(0^l), E_{pk_0}(u)}_{kth}, u\Big),$$

$$\mathcal{D}_3 = \Big(pk_0, \ldots, pk_{t-1}, \underbrace{E_{pk_0}(0^l), \ldots, E_{pk_{t-1}}(0^l), E_{pk_0}(u)}_{1st}, \ldots, \underbrace{E_{pk_0}(0^l), \ldots, E_{pk_{t-1}}(0^l), E_{pk_0}(u)}_{kth}, u'\Big).$$

Now, $\mathcal{D}_1 \equiv^c \mathcal{D}_2$ follows by k-ind t-circular security of $\mathcal{E}$; $\mathcal{D}_2 \equiv^c \mathcal{D}_3$ follows by CPA-security of $\mathcal{E}$; and $\mathcal{D}_3 \equiv^c \mathcal{D}_4$ follows by k-ind t-circular security of $\mathcal{E}$. □

Remark 2. In many concrete settings, for a public-key encryption scheme $\mathcal{E} = (Param, Gen, E, Dec)$ with public parameters, we have that

$$(par, pk, sk) \equiv (par, Pub(sk_u, par), sk_u),$$

where $par \leftarrow Param(1^n)$, $(pk, sk) \leftarrow G(1^n, par)$, $sk_u \leftarrow \{0,1\}^l$ and Pub is a deterministic function. That is, the secret key is chosen uniformly at random and the public key is obtained deterministically from the secret key and the public parameters. For such schemes we may easily modify Construction 3 so that, following the notation used in Construction 3, the index key ik is augmented with $par \leftarrow Param(1^n)$ and that the evaluation function F no longer takes pk_i's as input (so its entire input is a bitstring), by computing $pk_i = Pub(sk_i, par)$ on its own for $0 \leq i \leq t-1$. Now by taking $m \in \omega(t \cdot l)$ we obtain a TDF (from the assumptions stated in Theorem 3) that hides a $(1 - o(1))$-fraction of its input bits. Note that if one-wayness (as opposed to k-wise one-wayness) is desired we can have TDFs without public parameters as described above, i.e., by incorporating the parameter-generation algorithm of $\mathcal{E}$ into the key-generation algorithm of the constructed TDF. However, in order to have k-wise one-wayness the constructed TDF should also have a separate public-parameter generation algorithm.

5 CCA-secure encryption

5.1 Constructions of CCA secure encryption from our assumptions

Rosen and Segev [37, Theorem 1] give a construction of a CCA1-secure encryption scheme from any $\omega(\log n)$-wise one-way TDF and of a CCA2-secure encryption from any $\Omega(n)$-wise one-way TDFs. Their constructions are fully-blackbox and *non-shielding*, in the sense described below.

Recall the notion of *fully-blackbox reductions* [36]; we use (C, R) to denote a fully-blackbox reduction, where C denotes the construction and R denotes the reduction algorithm. In the definition below we review what it means for an encryption-based or TDF-based construction C to be *shielding/non-shielding* [24].

Definition 6. *An encryption-based blackbox construction $C = (G, E, D)$ is called* shielding *if D never calls the encryption function of the oracle scheme. Formally, for any $\mu = (g, e, d)$ that implements an encryption scheme, D^{μ} never calls e. Similarly, a TDF-based construction $C = (G, E, D)$ is called* non-shielding *if D never calls the evaluation algorithm of the oracle scheme. A construction is called non-shielding if it is not shielding.*

We call (C, R) a shielding (resp., non-shielding) fully-blackbox reduction if (1) (C, R) is a fully-blackbox reduction and (2) C is a shielding (resp., non-shielding) construction. We simply use the terms non-shielding/shielding blackbox constructions to refer to non-shielding/shielding fully-blackbox reductions.

The following result is from [37].

Theorem 4. *[37] There exists a non-shielding blackbox construction of CCA1-secure (resp., CCA2-secure) encryption schemes from $\omega(\log n)$-wise one-way (resp., $\Omega(n)$-wise one-way) TDFs. In particular, the constructed decryption algorithm (for both CCA1 and CCA2 cases) D calls F and F^{-1}, the evaluation and inversion algorithms, of the base TDF.*

We may now use Theorem 4 and our results from the previous section to obtain CCA1 and CCA2 secure encryption schemes from our assumptions. Note that all TDF constructions we have presented have the property that the constructed inversion algorithm, F^{-1}, calls the encryption algorithm of the base reproducible encryption scheme. Thus, we have the following corollary.

Corollary 1. *There exists a non-shielding blackbox construction of CCA1-secure (resp., CCA2-secure) encryption schemes from reproducible, $\omega(\log n)$-wise (resp., $\Omega(n)$-wise) t-circularly-secure encryption schemes, for any t.*

5.2 Shielding versus non-shielding CCA-secure constructions

Gertner, Malkin and Myers [24] show that there are no shielding blackbox construction of CCA1-secure encryption from CPA-secure encryption. In Corollary 1 we showed our assumptions (for appropriately chosen parameters) result in a non-shielding CCA1-secure encryption construction. Since our base assumptions are strictly stronger than CPA security (at least in a blackbox sense), a natural question is whether or not it is possible to give a shielding construction based on our assumptions. We do not currently know the answer to this question, but as we show below, there exists an encryption primitive, which is implied by our assumptions, based on which a non-shielding blackbox CCA1-construction is possible, but from which no shielding blackbox CCA1-construction is possible.

Our new encryption primitive is an extension of CPA-secure encryption, requiring that security holds even when encrypting certain *randomness-dependent messages*. The following definition is basically an adaptation of variants of those of [27,8] to the bit-encryption case.

Definition 7. *A bit-encryption scheme $\mathcal{E} = (Gen, E, Dec)$ with randomness space $\{0,1\}^{\rho}$ is q-randomness-dependent-message (RDM) secure if*

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

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 bit-by-bit, we should form each encryption under a fresh and independent public key. Otherwise, an adversary can easily distinguish between the two distributions. The reason is that an adversary given c_1 and c_2 for $c_1 = E_{pk}(b_1, r)$ and $c_2 = E_{pk}(b_2; r)$ can check whether $b_1 = b_2$.

q-RDM-secure encryption from our assumptions. We first show below that the notion defined above is implied by our assumptions. For simplicity, we show the implication from 1-circular security assumptions (i.e., circular security with respect to one pair of public/secret keys), although this generalizes to get implications from t-circular security assumptions.

Lemma 4. *Assume $\mathcal{E} = (Gen, E, Dec, R)$ is a reproducible, q-ind 1-circularly-secure bit encryption scheme with public key space $\{0,1\}^{l_1(n)}$, secret-key space $\{0,1\}^{l_2(n)}$ and randomness space $Rand_n$, for any security parameter n. There exists a q-RDM secure encryption scheme based on $\mathcal{E}$.*

Proof. Let $l_1 = l_1(n)$, $l_2 = l_2(n)$ and $Rand = Rand_n$. Given $\mathcal{E} = (Gen, E, Dec, R)$ we define below a bit encryption scheme $\mathcal{E}' = (Gen', E', Dec')$ whose randomness space, $Rand'$, is the public/secret key space of $\mathcal{E}$, i.e., $Rand' = \{0,1\}^{l_1+l_2}$.

- $Gen'(1^n)$: sample $(pk, sk) \leftarrow Gen(1^n)$ and $r \leftarrow Rand$; form the public key as $pk_{const} = E_{pk}(0; r)$ and the secret key as $sk_{const} = r$.
- E': given public key $pk_{const} = c$, bit b and randomness (pk', sk') return

$$c_{const} = (pk', R(c, b, sk')).$$

- D': given secret key $sk_{const} = r$ and ciphertext (pk', c') return the bit b such that $E_{pk'}(b; r) = c'$.

To prove q-RDM security of $\mathcal{E}'$ it suffices to show (by reproducibility of $\mathcal{E}$) that

$$\begin{aligned}\left(pk', E_{pk'}(pk'; \mathbf{r_1^1}), E_{pk'}(sk'; \mathbf{r_2^1}), \ldots, E_{pk'}(pk'; \mathbf{r_1^q}), E_{pk'}(sk'; \mathbf{r_2^q})\right)& \\ \equiv^c \left(pk', E_{pk'}(0^{l_1}; \mathbf{r_1^1}), E_{pk'}(0^{l_2}; \mathbf{r_2^1}), \ldots, E_{pk'}(0^{l_1}; \mathbf{r_1^q}), E_{pk'}(0^{l_2}; \mathbf{r_2^q})\right)&,\end{aligned}$$

where $(pk', sk') \leftarrow Gen(1^n)$, $\mathbf{r}_1^1, \ldots, \mathbf{r}_1^q \leftarrow \{0,1\}^{l_1}$ and $\mathbf{r}_2^1, \ldots, \mathbf{r}_2^q \leftarrow \{0,1\}^{l_2}$. The above indistinguishability follows easily from q-ind 1-circular security of $\mathcal{E}$. □

Non-shielding CCA1 construction from q-RDM-secure encryption. Next, we show q-RDM-secure encryption easily implies q-wise one-way TDFs, which we will use to show the existence of a non-shielding CCA1 construction. We sketch the construction based on a **q**-RDM-secure encryption scheme $\mathcal{E}$. Let $\mathcal{E}$'s randomness space be $\{0,1\}^\rho$, and define TDF $\mathcal{F} = (G, F, F^{-1})$ as follows. The algorithm $G(1^n)$ runs $Gen(1^n)$ ρ times to obtain $(pk_1, sk_1), \ldots, (pk_\rho, sk_\rho)$, and returns $ik = (pk_1, \ldots, pk_\rho)$ and $tk = (sk_1, \ldots, sk_\rho)$. The algorithm F has domain space $\{0,1\}^\rho$ and $F_{pk_1,\ldots,pk_\rho}(r)$ returns $(E_{pk_1}(r_1; r), \ldots, E_{pk_\rho}(r_\rho, r))$. The inversion algorithm F^{-1} works in the obvious way.

Now it is not hard to show if $\mathcal{E}$ is q-RDM secure, then $\mathcal{F}$ is q-wise one-way. Specifically, note that the view of an adversary against q-wise one-wayness of $\mathcal{F}$ is as

$$\left(E_{pk_1^1}(r_1; r), \ldots, E_{pk_\rho^1}(r_\rho, r), \ldots, E_{pk_1^q}(r_1; r), \ldots, E_{pk_\rho^q}(r_\rho, r)\right),$$

and also recall the definition of q-RDM security. (For better readability, we have removed the inclusion of pk_i^j's in the above equation.) Thus, by applying Theorem 4 we obtain the following.

Corollary 2. *For any $q \in \omega(\log n)$ there exists a non-shielding blackbox construction of CCA1-secure encryption from q-RDM-secure bit-encryption.*

Impossibility of shielding CCA1-construction from q-RDM-secure encryption. We now show the blackbox separation of [24], stating that there are no shielding blackbox constructions of CCA1-secure encryption from CPA-secure encryption, extends even if the base scheme is q-RDM-secure, for any poly-bounded q. Combined with Corollary 2 this gives us an encryption primitive which permits a non-shielding blackbox CCA1-secure construction, but from which no shielding blackbox CCA1-secure construction is possible. We first start with an informal description of the separation model of [24] and then give the formal definitions.

Specifically, [24] 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, which are defined based on $\mathcal{O}_1$. For any candidate construction $\mathcal{E} = (Gen^{\mathcal{O}_1}, Enc^{\mathcal{O}_1}, Dec^{\mathbf{g},\mathbf{d}})$ Gertner et al. prove that

1. 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 $\mathcal{E}$ with very high probability, where the probability is taken over a random choice of $(\mathbf{g}, \mathbf{e}, \mathbf{d}, \mathbf{w}, \mathbf{u})$ and all internal random coins of the CCA1 game (Formalized in Theorem 5 below);
2. no adversary $\mathcal{A}^{\mathcal{O}}$ that makes at most a polynomial number of queries can win against the CPA-security of $(\mathbf{g}, \mathbf{e}, \mathbf{d})$ with better $1/2 + poly/2^n$ probability, where the probability is taken over the random choice of $\mathcal{O} = (\mathbf{g}, \mathbf{e}, \mathbf{d}, \mathbf{w}, \mathbf{u})$ and those of the adversary and the CPA-game [24, Theorem 1]. That is, a random $(\mathbf{g}, \mathbf{e}, \mathbf{d})$ is CPA-secure in a very strong sense against any query-bounded oracle-adversary $\mathcal{A}^{\mathcal{O}}$.

Therefore, to rule out shielding fully-blackbox constructions of CCA1-secure encryption from a new encryption primitive, it suffices to prove Item 2 above with respect to the new primitive. This is what we do below with respect to RDM secure encryption (Theorem 6). We first give the formal description of the oracles as used in [24].

Definition 8. *([24]) 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* $(\mathbf{g}, \mathbf{e})$*, for which it holds* $\mathbf{d}(sk, c) = b$ *if there exists* $r \in \{0,1\}^n$ *s.t.* $\mathbf{e}(\mathbf{g}(sk), b, r) = c$*; 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 it holds that* $\mathbf{g}^{-1}(pk) = \emptyset$ *then* $\mathbf{w}(pk) = \perp$*; otherwise, sample the strings* $r_1, \ldots, r_n \leftarrow \{0,1\}^n$ *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 [24], 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 CCA1 security of the constructed scheme.

Theorem 5. *([24]) Fix a shielding bit-encryption construction (Gen, E, Dec). There exists a CCA1 adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, where $\mathcal{A}$ is poly-bounded in the number of queries but unbounded otherwise, for which it holds that*

$$\Pr\left[(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1^{Dec^{\mathbf{g},\mathbf{d}}(SK),\mathcal{O}}(PK)\ ;\ \mathcal{A}_2^{\mathcal{O}}(c, \sigma) = b\right] \geq 1 - \frac{1}{n}, \tag{16}$$

where $\mathcal{O} = (\mathbf{g}, \mathbf{e}, \mathbf{d}, \mathbf{w}, \mathbf{u}) \leftarrow \psi$, $b \leftarrow \{0,1\}$, $(PK, SK) \leftarrow Gen^{\mathbf{g},\mathbf{e},\mathbf{d}}(1^n)$ and $c \leftarrow E^{\mathbf{g},\mathbf{e},\mathbf{d}}(m_b)$. Note that σ is the state information that A_1 passes on to A_2. Also, $Dec^{\mathbf{g},\mathbf{d}}(SK)$ denotes the decryption oracle to which $\mathcal{A}_1$ has access. Note that since A is a CCA1 adversary only A_1 has access to $Dec^{\mathbf{g},\mathbf{d}}(SK)$.

We give and prove the following theorem, a CPA version of which was proved in [24].

Theorem 6. *For any (possibly) inefficient adversary $\mathcal{A}$ that makes at most $p = p(n)$ queries (for some p), it holds that*

$$\Pr_{\mathcal{O}=(\mathbf{g},\mathbf{e},\mathbf{d},\mathbf{w},\mathbf{u})\leftarrow\psi,\ \ b\leftarrow\{0,1\},\ ds_b\leftarrow\mathcal{DS}_b}\left[\mathcal{A}^{\mathcal{O}}(ds_b) = b\right] \leq \frac{1}{2} + \frac{poly(p)}{2^n} \tag{17}$$

where

$$\mathcal{DS}_0 \equiv \left((pk_1^1, \mathbf{e}_{pk_1^1}(r_1; r)), \ldots, (pk_n^1, \mathbf{e}_{pk_n^1}(r_n; r)), \ldots, (pk_1^q, \mathbf{e}_{pk_1^q}(r_1; r)), \ldots, (pk_n^q, \mathbf{e}_{pk_n^q}(r_n; r))\right) \tag{18}$$

$$\mathcal{DS}_1 \equiv \left((pk_1^1, \mathbf{e}_{pk_1^1}(0; r)), \ldots, (pk_n^1, \mathbf{e}_{pk_n^1}(0; r)), \ldots, (pk_1^q, \mathbf{e}_{pk_1^q}(0; r)), \ldots, (pk_n^q, \mathbf{e}_{pk_n^q}(0; r))\right), \tag{19}$$

in which $r \leftarrow \{0,1\}^n$ and pk_i^j, for $1 \leq i \leq n$ and $1 \leq j \leq q$, is formed by sampling $sk_i^j \leftarrow \{0,1\}^n$ and setting $pk_i^j = \mathbf{g}(sk_i^j)$.

Proof. We first fix some notation. Let Pub_{chal} be the set of public keys given to $\mathcal{A}$ as part of its input ds_b. (*chal* stands for challenge.) To be consistent with the above notation assume

$$Pub_{chal} = \{pk_1^1, \ldots, pk_n^1, \ldots, pk_1^q, \ldots, pk_n^q\}.$$

Let Pub be the set of public keys that $\mathcal{A}$ obtains by querying $\mathbf{g}$. Let $PubCiph_{chal}$ be the set of pairs of public keys/ciphertexts which $\mathcal{A}$ can retrieve as part of its input ds_b. Finally, let $PubCiph$ contains all elements of $PubCiph_{chal}$ plus those pairs of public keys/ciphertexts that $\mathcal{A}$ obtains by querying $\mathbf{e}$ and by querying $\mathbf{w}$.

First, we may assume that (1) $\mathcal{A}$ only calls its oracles on the security parameter n, (2) $\mathcal{A}$ never queries $\mathbf{u}$, (3) $\mathcal{A}$ only queries $\mathbf{w}$ on inputs $pk \in Pub_{chal}$ and (4) $\mathcal{A}$ never queries $\mathbf{d}$. We explain below why we can make these assumptions.

For (1), note that by Definition 8 the outputs of functions $(\mathbf{g}, \mathbf{e}, \mathbf{d}, \mathbf{u}, \mathbf{u})$ on different security parameters are independent of each other. Thus, calling these functions on security parameters other than n gives no knowledge to the adversary as the adversary can sample those answers by itself.

For (2), for any $(pk, c) \notin PubCiph$, 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 $(pk, c) \in PubCiph$, however, then $\mathcal{A}$ already knows the answer and there is no point in calling $\mathbf{u}$.

For (3), first note that for any $pk \notin Pub_{chal} \cup Pub$, the query $\mathbf{w}(pk)$ is answered with $\perp$ except with an inverse-exponential probability (using the same reasoning as above). Also, if $pk \in Pub$, then $\mathcal{A}$ itself can sample the answer by querying $\mathbf{e}$, since $\mathcal{A}$ knows $\mathbf{g}^{-1}(pk)$.

Using similar reasoning, we can show that any query $\mathbf{d}(sk, c)$ that $\mathcal{A}$ trivially does not know the answer to is replied to with $\perp$ except with an inverse-exponential probability.

Now assuming $\mathcal{A}$ makes at most $p = poly(n)$ queries, by observation (3) we may assume that $\mathcal{A}$ never queries $\mathbf{w}$, but instead $\mathcal{A}$'s input includes $p \times n^2 \times q = poly(p)$ more ciphertexts, where for each public key $pk_i^j \in Pub_{chal}$, we include p bit-by-bit encryptions of $sk_i^j = \mathbf{g}^{-1}(pk_i^j)$ (so $p \times n$ encryptions for each public key and since we have nq public keys this gives us the above number).

Now since $\mathcal{A}$ only queries $\mathbf{g}$ and $\mathbf{e}$, we consider the following events.

- (a) $PubHit$: at least one of $\mathbf{g}$ queries results in some $pk \in Pub_{chal}$;
- (b) $CiphHit$: $\mathcal{A}$ makes a query $\mathbf{e}(pk, b, r)$, to get c, and it holds that $(pk, c) \in C_{chal}$.

If neither $PubHit$ nor $CiphHit$ holds, then we can show that the probability that $\mathcal{A}$ can determine b is at most $1/2 + poly(n)/2^n$. Moreover, both $PubHit$ and $CiphHit$ can easily be shown that occur with at most $poly(n)/2^n$ probability. □

Using standard techniques in blackbox separations (especially applying the Borel-Cantelli lemma) Theorems 5 and 6 can be combined to obtain the following corollary.

Corollary 3. *For any poly-bounded q, there exists no shielding blackbox construction of CCA1-secure encryption from q-RDM-secure encryption.*

We note that it seems that one can generalize Corollary 3 to rule out the existence of shielding blackbox 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 blackbox 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 3 still holds true if RDM security is replaced with circular security.

6 Constructions for Deterministic encryption

6.1 Preliminaries

We start by reviewing a few basic facts relating to entropy. The *min-entropy* of a distribution (or a random variable) $\mathcal{D}$ is defined as

$$H_\infty(D) = \min_{d \in \mathcal{D}} \log(1/\Pr[\mathcal{D} = d]).$$

If $l \leq H_\infty(\mathcal{D})$ we call $\mathcal{D}$ an l-source. We also recall the notion of *average min entropy*, formalized by Dodis et al. [21], defined as

$$\tilde{H}_\infty(X|Y) = -\log\left(E_{y \leftarrow Y}(2^{-H_\infty(X|Y=y)})\right),$$

where (X, Y) are two random variables.

The following is a well-known fact about average min entropy.

Lemma 5. *([21]) For any random variables (X, Y) it holds that $\tilde{H}_\infty(X|Y) \geq \tilde{H}_\infty(X, Y) - \log|Sup(Y)|$.*

Let S be a set of function indices. Recall that a family of functions $\mathcal{H} = \{hash_i \colon \{0,1\}^k \mapsto R \mid i \in S\}$ is called universal if for all $x_1, x_2 \in \{0,1\}^k$ with $x_1 \neq x_2$ it holds that

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

The following lemma, from [21], shows that universal hash functions are good *average-case extractors*.

Lemma 6. *([21]) Let $\mathcal{H} = \{hash_i \colon \{0,1\}^k \mapsto R \mid i \in S\}$ be a family of universal hash functions. For any random variables (X, D), where D takes values in $\{0,1\}^k$, it holds that*

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

where $hash \leftarrow \mathcal{H}$.

6.2 Deterministic encryption: syntax and security

Since a deterministic encryption scheme is syntactically the same as a TDF, we denote a deterministic-encryption scheme as $\mathcal{DE} = (G, F, F^{-1})$. For a function l we call $\mathcal{DE}$ an l-bit scheme if the plaintext space of $\mathcal{DE}$ on any security parameter n is $\{0,1\}^{l(n)}$. We start by giving a notion of security for deterministic encryption schemes, which is essentially the single-message, indistinguishability-based notion of [10]. See [10] for definitional equivalences.

Definition 9. *We say that a deterministic l-bit encryption scheme $\mathcal{DE} = (G, F, F^{-1})$ is secure with respect to indistinguishability of λ-source inputs (shortly, (λ, l)-IND secure) if for any λ-sources $\mathcal{M}_0$ and $\mathcal{M}_1$ over $\{0,1\}^l$, it holds that*

$$(ik, F_{ik}(\mathcal{M}_0)) \equiv^c (ik, F_{ik}(\mathcal{M}_1)),$$

where $(ik, tk) \leftarrow G(1^n)$.

6.3 Tools for obtaining deterministic encryption

In definitions below, we explicitly include the parameter-generation algorithm, since the definitions delicately depend on the presence of public parameters. Throughout this section we work with randomized encryption schemes (as base schemes for obtaining DE schemes) whose key generation algorithms admit a special form stated in Remark 2 and reviewed below.

Definition 10. *We call a randomized encryption scheme $\mathcal{E} = (Param, Gen, E, Dec)$ with secret-key-length function $l = l(n)$ canonical if there exists a deterministic function Pub such that*

$$(par, pk, sk) \equiv (par, Pub(sk_u, par), sk_u),$$

where $par \leftarrow Param(1^n)$, $(pk, sk) \leftarrow G(1^n, par)$, $sk_u \leftarrow \{0,1\}^l$. That is, the secret key of a canonical scheme is chosen uniformly at random and the public key is obtained deterministically from the secret key and the public parameters. Henceforth, we will reserve Pub to denote the stated function of a canonical form scheme under consideration.

We start by defining an extended notion of circular security, requiring that 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 [14] to the average case. We note it is possible to prove our results with respect to the weaker definition of [14], but the proofs become more complex.

Definition 11. *We say that a canonical bit-encryption scheme $\mathcal{E} = (Param, Gen, E, Dec)$ with secret-key-length function l is (λ, l)-entropy circularly secure if for any joint distribution $(\mathcal{SK}, \mathcal{X})$, where $\mathcal{SK}$ is a distribution over $\{0,1\}^l$, satisfying the condition $\tilde{H}_\infty(\mathcal{SK}|\mathcal{X}) \geq \lambda$, we have*

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

where $(sk, x) \leftarrow (\mathcal{SK}, \mathcal{X})$, $par \leftarrow Param(1^n)$ and $pk = Pub(sk, par)$. We stress that par is chosen independently of (sk, x).

We should clarify that Definition 11 is different from simply the combination of circular security and *leakage resilience* notions [1,32]. Under the leakage-resilience model, the public/secret keys are chosen as spelled out by the scheme, but the leakage function f (to be evaluated on the secret key) is chosen by the adversary (after seeing the public key). Under our model, in contrast, the secret key may be chosen from a non-full-entropy distribution, but the leaked information (x above) is chosen independently of the random par.

Next we define another strengthening of the notion of [14], 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* [33], as is the case with known circularly-secure schemes [12,13].

Definition 12. *Let $\mathcal{E} = (Param, Gen, E, Dec)$ be a canonical bit-encryption scheme with secret-key-length function l. We say $\mathcal{E}$ is strongly-(λ, l)-entropy circularly secure if*

- *(a) for any λ-source $\mathcal{SK}$ on $\{0,1\}^l$,*

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

  where $sk \leftarrow \mathcal{SK}$, $par \leftarrow Param(1^n)$ and $pk = Pub(sk, par)$; and
- *(b) for any λ-sources $\mathcal{SK}_1$ and $\mathcal{SK}_2$ on $\{0,1\}^l$, it holds that*

$$(par, Pub(sk_1, par)) \equiv^c (par, Pub(sk_2, par)),$$

  where $sk_1 \leftarrow \mathcal{SK}_1$, $sk_2 \leftarrow \mathcal{SK}_2$ and $par \leftarrow Param(1^n)$. Note that par is chosen independently from both sk_1 and sk_2.

As mentioned before, Condition (b) above in some sense states that Pub should act closely like a seeded randomness extractor.

6.4 Constructions

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

Theorem 7. *Let $\mathcal{E} = (Param, Gen, E, Dec, R)$ be a canonical reproducible bit-encryption scheme with secret-key-length function l and $\mathcal{DE} = C_1(\mathcal{E}, 1)$ be the DE scheme built in Construction 1 based on $\mathcal{E}$ and $t = 1$.*[8] *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 $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$, where

$$\mathcal{DS}_1 \equiv (\underbrace{par, pk, E_{pk}(0^l; \mathbf{r})}_{\textbf{ik}}, \underbrace{pk_1, E_{pk_1}(sk_1; \mathbf{r})}_{\textbf{image}}), \text{ and}$$

$$\mathcal{DS}_2 \equiv (\underbrace{par, pk, E_{pk}(0^l; \mathbf{r})}_{\textbf{ik}}, \underbrace{pk_2, E_{pk_2}(sk_2; \mathbf{r})}_{\textbf{image}}),$$

are computationally indistinguishable, where $(pk, sk) \leftarrow Gen(1^n)$, $sk_1 \leftarrow \mathcal{D}_1$, $sk_2 \leftarrow \mathcal{D}_2$, $par \leftarrow Param(1^n)$, $pk_1 = Pub(sk_1, par)$, $pk_2 = Pub(sk_2, par)$ and $\mathbf{r} \leftarrow Rand_n^l$. Fix the described way of sampling variables in the following. From strong-(λ, l)-entropy circularly security we obtain

$$\begin{aligned} &(par, pk_1, E_{pk_1}(sk_1; \mathbf{r})) \equiv^c (par, pk_1, E_{pk_1}(0^l; \mathbf{r})), \text{ and} \\ &(par, pk_2, E_{pk_2}(sk_2; \mathbf{r})) \equiv^c (par, pk_2, E_{pk_2}(0^l; \mathbf{r})). \end{aligned} \tag{20}$$

Now from Equation 20 and reproducibility of $\mathcal{E}$, we obtain

$$\begin{aligned} &\mathcal{DS}_1 \equiv^c (par, pk, E_{pk}(0^l; \mathbf{r}), pk_1, E_{pk_1}(0^l; \mathbf{r})), \text{ and} \\ &\mathcal{DS}_2 \equiv^c (par, pk, E_{pk}(0^l; \mathbf{r}), pk_2, E_{pk_2}(0^l; \mathbf{r})). \end{aligned} \tag{21}$$

Now since

$$(par, pk_1) \equiv^c (par, pk_2),$$

which is again implied by strong (λ, l)-entropy circularly security of $\mathcal{E}$, we have

$$(par, pk, E_{pk}(0^l; \mathbf{r}), pk_1, E_{pk_1}(0^l; \mathbf{r})) \equiv^c (par, pk, E_{pk}(0^l; \mathbf{r}), pk_2, E_{pk_2}(0^l; \mathbf{r}))$$

and hence $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$, as desired. □

Next we show that the "weaker" entropy circular security assumption also gives rise to DE schemes, but with looser security bounds and under more inefficient constructions. Our construction employs the encrypt-with-hardcore (EWH) technique, described in the introduction.

As terminology, we say that a bit encryption scheme $\mathcal{E} = (Param, Gen, E, Dec, R)$ has a *bit-string ciphertext space* if there exists a polynomial p_c such that the ciphertext space of $\mathcal{E}$ is a subset of $\{0,1\}^{p_c}$: formally, for all n, all $par \in Param(1^n)$, all $(pk, sk) \in Gen(1^n, par)$ and all b it holds that all $Sup(E_{pk}(b)) \subseteq \{0,1\}^{p_c(n)}$. Similarly, we may define an encryption scheme with a bitstring ciphertext space or a TDF with a bitstring image space, etc.

[8] Here we are working with a modified version of Construction 1 stated in Remark 2. Note that the constructed deterministic encryption scheme does not have public parameters.

Theorem 8. *Let $\mathcal{E} = (Param, Gen, E, Dec, R)$ be a canonical, reproducible, (λ, l)-entropy circularly secure encryption scheme, with randomness space $Rand_n = \{0,1\}^{p_r}$, secret-key-length function $l = l(n)$ and with bitstring public-key and ciphertext spaces. 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 first, and then proceed with the formal proof. The first step of the proof is to show that we can use reproducibility of $\mathcal{E}$ to encrypt any arbitrarily-long bitstring, say of length $p = p(n)$, using a p_r-bit-long randomness string. (Recall that $Rand_n = \{0,1\}^{p_r}$.) This can be done (see Lemma 7) by defining a new PKE scheme whose public keys are vectors of p base-public-keys, $(pk_1, \ldots, pk_p)$, and in which the encryption function reuses randomness $r \in \{0,1\}^{p_r}$ to encrypt $m = m_1 \ldots m_p \in \{0,1\}^p$ as

$$(E_{pk_1}(m_1; r), \ldots, E_{pk_p}(m_p; r)).$$

Now consider the TDF given by Construction 3, based on $t = 1$ and $m = l + p_r - \lambda$. Define

$$hc(sk, x) = (hash, hash(x)),$$

where $hash\colon \{0,1\}^m \mapsto \{0,1\}^{p_r}$ is chosen from a *family of universal hash functions*. As the next step we show that hc is a hardcore function for the TDF. Having proved this, to be able to apply the EWH method, we need to show that $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$, for

$$\mathcal{DS}_1 \equiv (hash(x), hash, par, pk, E_{pk}(0^l; \mathbf{r}_1), E_{pk}(0^{|x|}; \mathbf{r}_1), E_{pk}(sk; \mathbf{r}_1), E_{pk}(x; \mathbf{r}_2)), \text{ and}$$
$$\mathcal{DS}_2 \equiv (y, hash, par, pk, E_{pk}(0^l; \mathbf{r}_1), E_{pk}(0^{|x|}; \mathbf{r}_1), E_{pk}(sk; \mathbf{r}_1), E_{pk}(x; \mathbf{r}_2)),$$

where $y \leftarrow \{0,1\}^{p_r}$, $par \leftarrow Param(1^n)$, $(sk, x) \leftarrow (\mathcal{SK}, \mathcal{X})$, $pk = Pub(sk, par)$ 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 by Lemma 5

$$\tilde{H}_\infty(\mathcal{SK}|\mathcal{X}) \geq \tilde{H}_\infty(\mathcal{SK}, \mathcal{X}) - \log|Sup(\mathcal{X})| \geq (l + p_r + u) - (l + p_r - \lambda) = \lambda + u$$

we may appeal to the (λ, l)-entropy circular security of $\mathcal{E}$ to replace $E_{pk}(sk; \mathbf{r}_1)$, in both $\mathcal{DS}_1$ and $\mathcal{DS}_2$, with an all-zero encryption. (That is, we deduce that, say, $\mathcal{DS}_1$ is indistinguishable from a distribution that is exactly the same as $\mathcal{DS}_1$ but in which $E_{pk}(sk; \mathbf{r}_1)$ is replaced with $E_{pk}(0^l; \mathbf{r}_1)$.) 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). Finally, using the facts that

$$\tilde{H}_\infty(\mathcal{X}|\mathcal{SK}) \geq \tilde{H}_\infty(\mathcal{X}, \mathcal{SK}) - |Sup(\mathcal{SK})| \geq (l + p_r + u) - l = p_r + u,$$

and that $u \in \omega(\log n)$, we apply Lemma 6 to replace $hash(x)$ with a random string y.

We now give the formal proof of the theorem above. Before giving the proof, we need to establish some lemmas.

In the following we will introduce variants of the TDFs discussed earlier, with associated hardcore functions hc, where hc also takes as input the underlying index key ik, besides the domain input x to produce $hc(ik, x)$, i.e., hc also depends on the underlying index key.

We give the following simple lemma, showing that using reproducibility, one can obtain schemes with arbitrarily-large plaintexts using relatively short randomness.

Lemma 7. *Assuming the existence of a reproducible, CPA-secure bit-encryption scheme $\mathcal{E} = (Param, G, E, D)$ with randomness space $\{0,1\}^{p_r}$, for any poly function p there exists a CPA-secure p-bit encryption scheme $\mathcal{E}' = (Param', G', E', D')$ with the same randomness space $\{0,1\}^{p_r}$.*

Proof. Define $\mathcal{E}'$ as follows:

- $Param' = Param$;
- $G'(1^n; par)$: run $G(1^n; par)$ p times to produce $(pk_1, sk_1), \ldots, (pk_p, sk_p)$ and form the public key as $(pk_1, \ldots, pk_p)$ and the secret key as $(sk_1, \ldots, sk_p)$;
- E': on public key $pk_{ext} = (pk_1, \ldots, pk_p)$, message $m \in \{0,1\}^p$ and randomness $r \in \{0,1\}^{p_r}$ return
$$(E_{pk_1}(m_1; r), \ldots, E_{pk_p}(m_p; r)).$$
- D': clear.

Using a simple hybrid argument one can prove the CPA-security of $\mathcal{E}'$ based on the CPA-security and reproducibility of $\mathcal{E}$. To do this, for $1 \leq i \leq p+1$ define hybrid $\mathcal{D}_i$ under which an encryption of m under $(pk_1, \ldots, pk_p)$ and randomness r is produced as

$$\begin{array}{ll} (E_{pk_1}(0; r), \ldots, E_{pk_{i-1}}(0; r), E_{pk_i}(m_i; r), \ldots, E_{pk_p}(m_p; r)) & 1 \leq i \leq p \\ (E_{pk_1}(0; r), \ldots, E_{pk_{i-1}}(0; r), E_{pk_i}(0; r), \ldots, E_{pk_p}(0; r)) & i = p+1. \end{array}$$

Now note that $\mathcal{D}_1$ and $\mathcal{D}_{p+1}$ are identically distributed to, respectively, $E_{pk}(m)$ and $E_{pk}(0^p)$, and that $\mathcal{D}_1 \equiv^c \cdots \equiv^c \mathcal{D}_{p+1}$. □

We note that a similar version of Lemma 7 may be proved without assuming reproducibility, but by applying a pseudorandom generator (PRG) to stretch the randomness for the encryption algorithm. However, since we need reproducibility for other purposes anyway, we work with the version of the lemma given above.

As terminology, we say that function f is k-bit-valued if f's output is always in $\{0,1\}^k$.

Theorem 9. *Let $\mathcal{E}_1$ be a reproducible, CPA-secure bit-encryption scheme with randomness space $\{0,1\}^{p_r}$. Let $\mathcal{F} = (G, F, F^{-1})$ be a TDF family with an associated p_r-bit-valued hardcore function hc, with a bitstring image space and suppose $Domain(F) = \{0,1\}^l$. Assume that for any λ-source distribution $\mathcal{M}$ on $\{0,1\}^l$ it holds that*

$$(ik, F_{ik}(x), hc(ik, x)) \equiv^c (ik, F_{ik}(x), s), \tag{22}$$

where $(ik, tk) \leftarrow G(1^n)$, $x \leftarrow \mathcal{M}$ and $s \leftarrow \{0,1\}^{p_r}$. Then, there exists a (λ, l)-IND-secure deterministic encryption scheme $\widetilde{\mathcal{DE}} = (\widetilde{G}, \widetilde{F}, \widetilde{F}^{-1})$.

Proof. Since $\mathcal{F}$ has a bitstring image space, let the image space of $\mathcal{F}$ be a subset of $\{0,1\}^{p_o}$. Given $\mathcal{E}_1$, by Lemma 7, we may assume the existence of a CPA-secure encryption scheme $\mathcal{E} = (Param, Gen, E, Dec)$, whose randomness space is $\{0,1\}^{p_r}$ and whose plaintext space is $\{0,1\}^{p_o}$. We define $\widetilde{\mathcal{DE}} = (\widetilde{G}, \widetilde{F}, \widetilde{F}^{-1})$ as follows.

- $\widetilde{G}(1^n)$: return (ik, pk) as the injective key and (tk, sk) as the trapdoor key, by sampling
$$par \leftarrow Param(1^n);\ (pk, sk) \leftarrow Gen(1^n, par) \text{ and } (ik, tk) \leftarrow G(1^n);$$
- $\widetilde{F}$: define
$$\hat{F}_{(ik,pk)}(m) = E_{pk}\left(F_{ik}(m); h_{ik}(m)\right);$$

– $\widetilde{F}^{-1}$: define

$$\hat{F}^{-1}_{(tk,sk)}(c) = F^{-1}_{tk}(Dec_{sk}(c)).$$

The completeness of $\widetilde{\mathcal{DE}}$ is clear. Now toward (λ, l)-IND-security of $\widetilde{\mathcal{DE}}$, we need to prove that for arbitrary λ-source distributions $\mathcal{M}_1$ and $\mathcal{M}_2$, it holds that $\mathcal{DS}_1 \equiv^c \mathcal{DS}_4$, where

$$\begin{aligned}\mathcal{DS}_1 &= (ik, pk, E_{pk}(F_{ik}(m); hc(ik, m)))\\ \mathcal{DS}_4 &= \left(ik, pk, E_{pk}(F_{ik}(m'); hc(ik, m'))\right).\end{aligned}$$

where $(ik, tk) \leftarrow G(1^n)$, $par \leftarrow Param(1^n)$, $(pk, sk) \leftarrow Gen(1^n, par)$, $m \leftarrow \mathcal{M}_1$ and $m' \leftarrow \mathcal{M}_2$. Fix this way of sampling variables in what comes below.

To prove $\mathcal{DS}_1 \equiv^c \mathcal{DS}_4$ we introduce $\mathcal{DS}_2$ and $\mathcal{DS}_3$ and show

$$\mathcal{DS}_1 \equiv^c \mathcal{DS}_2 \equiv^c \mathcal{DS}_3 \equiv^c \mathcal{DS}_4.$$

Define

$$\begin{aligned}\mathcal{DS}_2 &= (ik, pk, E_{pk}(F_{ik}(m); s))\\ \mathcal{DS}_3 &= \left(ik, pk, E_{pk}(F_{ik}(m'); s)\right),\end{aligned}$$

where $s \leftarrow \{0,1\}^{p_r}$.

Recall that by the assumption on $\mathcal{F}$ we have

$$(ik, F_{ik}(m), hc(ik, m)) \equiv^c (ik, F_{ik}(m), s)\,; \tag{23}$$

$$\left(ik, F_{ik}(m'), hc(ik, m')\right) \equiv^c \left(ik, F_{ik}(m'), s\right). \tag{24}$$

Now $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$ follows by Equation 23; $\mathcal{DS}_2 \equiv^c \mathcal{DS}_3$ follows by CPA security of $\mathcal{E}$; $\mathcal{DS}_3 \equiv^c \mathcal{DS}_4$ follows by Equation 23. □

We are now ready to give the proof of Theorem 8.

Theorem 8. (restated) *Let $\mathcal{E} = (Param, Gen, E, Dec, R)$ be a canonical, reproducible, (λ, l)-entropy circularly secure encryption scheme, with randomness space $Rand_n = \{0,1\}^{p_r}$, secret-key-length function $l = l(n)$ and with bitstring public-key and ciphertext spaces. 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.*

Proof. Fix $u \in \omega(\log n)$ and let $l_i = l + p_r + u$ and $l_o = 2l + p_r - \lambda$. Our goal is to build an (l_i, l_o)-IND-secure deterministic encryption scheme. To do so, since we already have $\mathcal{E}$, which is reproducible with randomness space $\{0,1\}^{p_r}$, by Theorem 9 it suffices to construct a TDF $\mathcal{F} = (G, F, F^{-1})$ with a bitstring image space and with an associated p_r-bit-valued hardcore function hc, which satisfies the following properties:

1. $Domain(F) = \{0,1\}^{l_o}$; and
2. For any l_i-source $\mathcal{M}$ over $\{0,1\}^{l_o}$,

$$(ik, F_{ik_{ext}}(x_{ext}), hc(ik_{ext}, x_{ext}) \equiv^c (ik_{ext}, F_{ik_{ext}}(x_{ext}), s),$$

where $x_{ext} \leftarrow \mathcal{M}$, $(ik_{ext}, tk_{ext}) \leftarrow G(1^n)$ and $s \leftarrow \{0,1\}^{p_r}$.

Thus, we focus on building $(\mathcal{F}, hc)$ with the properties above. To this end, we need a universal family $\mathcal{H}$ of hash functions from $\{0,1\}^{l+p_r-\lambda}$ to $\{0,1\}^{p_r}$.

We build $\mathcal{F} = (G, F, F^{-1})$ by instantiating Construction 3 with $\mathcal{E}$, integer $t = 1$ and integer $m = l + p_r - \lambda$, with the only difference that we augment the injective key with $hash \leftarrow \mathcal{H}$.[9] (Recall that for Construction 3 t denotes the number of public/secret key pairs and m is the number of bits added to the input of the TDF.)

Note that

$$Domain(F) = \{0,1\}^{l+l+p_r-\lambda} = \{0,1\}^{l_o}$$

so Property 1 above is satisfied. To define the associated hardcore function hc, for an injective key $ik_{ext} = (ik, hash)$ and domain point $x_{ext} = (sk, x) \in \{0,1\}^l \times \{0,1\}^{l+p_r-\lambda}$, we simply define

$$hc(ik_{ext}, x_{ext}) = hash(x).$$

For property 2 we need to show that for any arbitrary l_i-source $\mathcal{M}$, it holds that $\mathcal{DS} \equiv^c \mathcal{DS}'$, where

$$\mathcal{DS} \equiv (\underbrace{par, pk, E_{pk}(0^l; \mathbf{r}_1), E_{pk}(0^m; \mathbf{r}_2), hash}_{\mathbf{ik}}, \underbrace{pk', E_{pk'}(sk'; \mathbf{r}_1), E_{pk'}(x; \mathbf{r}_2)}_{\mathbf{image}}, \underbrace{hash(x)}_{\mathbf{hc}}), \text{ and} \quad (25)$$

$$\mathcal{DS}' \equiv (\underbrace{par, pk, E_{pk}(0^l; \mathbf{r}_1), E_{pk}(0^m; \mathbf{r}_2), hash}_{\mathbf{ik}}, \underbrace{pk', E_{pk'}(sk'; \mathbf{r}_1), E_{pk'}(x; \mathbf{r}_2)}_{\mathbf{image}}, \underbrace{s}_{\mathbf{hc}}),$$

in which $par \leftarrow Param(1^n)$, $(pk, sk) \leftarrow Gen(1^n, par)$, $\mathbf{r}_1 \leftarrow Rand_n^l$, $\mathbf{r}_2 \leftarrow Rand_n^m$, $hash \leftarrow \mathcal{H}$, $(sk', x) \leftarrow \mathcal{M}$, $pk' = Pub(sk', par)$ and $s \leftarrow \{0,1\}^{p_r}$. Since $\mathcal{E}$ is reproducible, for each of the above two distributions, given

$$(par, hash, \mathbf{image}, \mathbf{hc})$$

one can perfectly simulate the rest. Thus, to show $\mathcal{DS} \equiv^c \mathcal{DS}'$ it suffices to prove $\mathcal{DS}_1 \equiv^c \mathcal{DS}_4$, where

$$\mathcal{DS}_1 \equiv (par, hash, \underbrace{pk', E_{pk'}(sk'; \mathbf{r}_1), E_{pk'}(x; \mathbf{r}_2)}_{\mathbf{image}}, \underbrace{hash(x)}_{\mathbf{hc}}), \text{ and} \quad (26)$$

$$\mathcal{DS}_4 \equiv (par, hash, \underbrace{pk', E_{pk'}(sk'; \mathbf{r}_1), E_{pk'}(x; \mathbf{r}_2)}_{\mathbf{image}}, \underbrace{s}_{\mathbf{hc}}),$$

We introduce two more distributions, $\mathcal{DS}_2$ and $\mathcal{DS}_3$, and will show that $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2 \equiv^c \mathcal{DS}_3 \equiv^c \mathcal{DS}_4$, which will conclude the proof.

Fix the above way of sampling variables. Define

$$\mathcal{DS}_2 \equiv (par, hash, pk', E_{pk'}(0^l; \mathbf{r}_1), E_{pk'}(0^m; \mathbf{r}_2), hash(x)), \quad (27)$$

$$\mathcal{DS}_3 \equiv (par, hash, pk', E_{pk'}(0^l; \mathbf{r}_1), E_{pk'}(0^m; \mathbf{r}_2), s).$$

Before proving the desired indistinguishability relations we give the following two facts, obtained from Lemma 5.

[9] We are again working with the modified version of Construction 1 stated in Remark 2.

$$\tilde{H}_\infty(sk'|x) \geq H_\infty(sk', x) - (l + p_r - \lambda) = l_i - (l + p_r - \lambda) = \lambda + u \tag{28}$$

$$\tilde{H}_\infty(x|sk') \geq H_\infty(x, sk') - l = p_r + u. \tag{29}$$

We now proceed with the rest of the proof.
To prove $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$, note that by Equation 28 and the (λ, l)-entropy circular security (Definition 11) we have

$$\left(par, pk', E_{pk'}(sk'; \mathbf{r}_1), E_{pk'}(x; \mathbf{r}_2), x\right) \equiv^c \left(par, pk', E_{pk'}(0^l; \mathbf{r}_1), E_{pk'}(0^m; \mathbf{r}_2), x\right), \tag{30}$$

which imply $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$.

To prove $\mathcal{DS}_2 \equiv^c \mathcal{DS}_3$ it suffices to show

$$\mathcal{DS}'_2 = (hash, hash(x), sk') \text{ and } \mathcal{DS}'_3 = (hash, s, sk')$$

are (statistically) indistinguishable: this is because we can define a randomized algorithm A such that $A(\mathcal{DS}'_2) \equiv \mathcal{DS}_2$ and $A(\mathcal{DS}'_3) = \mathcal{DS}_3$: A samples par at random and lets $pk' = Pub(sk', par)$ and also samples the rest of the variables appropriately. By Lemma 6 we have

$$\Delta\left(\mathcal{DS}'_2, \mathcal{DS}'_3\right) \leq \frac{1}{2}\sqrt{\frac{2^{p_r}}{2^{\tilde{H}_\infty(x|sk')}}} \leq \frac{1}{2}\sqrt{\frac{2^{p_r}}{2^{(p_r+u)}}} \leq \frac{1}{2^{u/2}} = negl(n), \tag{31}$$

where the second inequality follows from Equation 29.

To prove $\mathcal{DS}_3 \equiv^c \mathcal{DS}_4$ note that by Equation 30 we have

$$\left(par, pk', E_{pk'}(sk'; \mathbf{r}_1), E_{pk'}(x; \mathbf{r}_2)\right) \equiv^c \left(par, pk', E_{pk'}(0^l; \mathbf{r}_1), E_{pk'}(0^m; \mathbf{r}_2)\right), \tag{32}$$

which implies $\mathcal{DS}_3 \equiv^c \mathcal{DS}_4$. □

7 Realizations

In this section we show how to build, based on concrete assumption, encryption schemes that provide reproducibility and also strong forms of circular security, i.e., (λ, l)-strong circular security for an appropriate setting of parameters.

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 integer $x > 1$, $g^x = g \cdot g^{x-1}$. For an integer x we define $g^{-x} = (g^x)^{-1}$. 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 $(b_1, \ldots, b_l) \odot (g_1, \ldots, g_l) = \prod_{1 \leq i \leq l} g_i^{b_i}$.

7.1 From the decisional Diffie -Hellman (DDH) assumption

Let $\mathcal{G}$ be a *group scheme*, 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

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

where $\mathbb{G}$ is chosen by running $\mathcal{G}(1^n)$, $g_1, \ldots, g_4 \leftarrow \mathbb{G}$ and $d \leftarrow \mathbb{Z}_{|\mathbb{G}|}$.

We present the encryption scheme of [12], which we refer to as the BHHO scheme, below.

Definition 13. *(From [12]) Define $\mathcal{E} = (Param, Gen, E, Dec)$, which is parameterized over an integer $l = l(n)$ (which we instantiate later), as follows.*

- *$Param(1^n)$: Produce $(\mathbb{G}, g, o) \leftarrow \mathcal{G}(1^n)$ and return $par = (\mathbb{G}, g, \mathbf{g})$, where $\mathbf{g} \leftarrow \mathbb{G}^l$;*
- *$Gen(1^n)$: Sample the secret key as $sk \leftarrow \{0,1\}^l$ and set the public key $pk = sk \odot \mathbf{g}$;*
- *$E_{pk}(g_1; r))$: Sample $r \leftarrow \mathbb{Z}_q$ and return $(\mathbf{g}^r, pk^r \cdot g_1)$; and*
- *$D_{sk}((\mathbf{g}', g'))$: Clear from the encryption algorithm.*

Reproducibility. We now verify the reproducibility property with respect to every fixed choice of par. To do this, we need to show that from

$$\left(\underbrace{\mathbb{G}, g, \mathbf{g}}_{par}, \underbrace{sk_1 \odot \mathbf{g}}_{pk_1}, \underbrace{(\mathbf{g}^r, pk_1^r \cdot g_1)}_{E_{pk_1}(g_1;r)}, \underbrace{g_2}_{\text{target message}}, \underbrace{sk_2}_{\text{target secret key}} \right),$$

one can compute $(\mathbf{g}^r, (sk_2 \odot \mathbf{g})^r \cdot g_2)$; this is easy to see considering that the last quantity is indeed $(\mathbf{g}^r, (sk_2 \odot \mathbf{g}^r) \cdot g_2)$, and that all of $\mathbf{g}^r$, sk_2 and g_2 are provided in the input tuple.

We show below the optimized version of the instantiation of our general TDF construction using the BHHO scheme. By optimized we mean we have removed all redundancies created under the "raw" instantiation.

Construction 4 *The TDF is parameterized over $l = l(n)$. See Theorem 10 on how to instantiate l.*

- *G: sample $(\mathbb{G}, g, o) \leftarrow \mathcal{G}(1^n)$ and sample the trapdoor key as*

$$tk = (r_1, \ldots, r_l) \leftarrow \mathbb{Z}_{|\mathbb{G}|}^l$$

  and the injective key as

$$ik = \begin{pmatrix} \mathbf{g} \\ \mathbf{g}^{r_1} \\ \vdots \\ \mathbf{g}^{r_l} \end{pmatrix} \tag{33}$$

  where $\mathbf{g} \leftarrow \mathbb{G}^l$.
- *F: on injective key*

$$ik = \begin{pmatrix} \mathbf{g}' \\ \mathbf{g}'_1 \\ \vdots \\ \mathbf{g}'_l \end{pmatrix} \tag{34}$$

  and domain point $x \in \{0,1\}^l$ return

$$F_{ik}(x) = \begin{pmatrix} x \odot \mathbf{g}' \\ (x \odot \mathbf{g}'_1) \cdot g^{x_1} \\ \vdots \\ (x \odot \mathbf{g}'_1) \cdot g^{x_l} \end{pmatrix} \tag{35}$$

– F^{-1}*: on trapdoor key* $(r_1, \ldots, r_l) \in \mathbb{Z}_{|G|}^l$ *and image*

$$ik = \begin{pmatrix} g' \\ g'_1 \\ \vdots \\ g'_l \end{pmatrix} \tag{36}$$

return $x = x_1 x_2 \ldots x_l \in \{0,1\}^l$ *where* x_i *is the bit such that*

$$g'_i = (g')^{r_i} \cdot g^{x_i}$$

Next, we show that the proof of circular security of [12] easily extends to yield strong-(λ, l)-entropy circular security, where the ratio $\frac{\lambda}{l}$ can get as inverse-polynomially-small as one desires. (Formally, for any *a priori* fixed polynomial $p = p(n)$ we can have an instantiation of the BHHO scheme which is $(\frac{l}{p}, l)$-entropy circularly secure.) The proof of entropy circular security (Theorem 10 below) is, however, implicit in [12] and we include it here only for self-containment purposes. See also [15, Lemma 5.1, Corollary 5.2] for similar statements. We first recall the following proposition, a more general version of which was proved in [12] and then give the main theorem.

Proposition 1. *(From [12]) Let* $\mathcal{G}$ *be a DDH-hard group scheme. 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*

$$\mathcal{DS}_1 = \begin{pmatrix} g_1 & g_2 & \cdots & g_l \\ g_1^{r_1} \cdot g_{1,1} & g_2^{r_1} \cdot g_{1,2} & \cdots & g_l^{r_1} \cdot g_{1,l} \\ \vdots & \vdots & \ddots & \vdots \\ g_1^{r_v} \cdot g_{v,1} & g_2^{r_v} \cdot g_{v,2} & \cdots & g_l^{r_v} \cdot g_{v,l} \end{pmatrix} \tag{37}$$

$$\mathcal{DS}_2 = \begin{pmatrix} g_1 & g_2 & \cdots & g_l \\ g_1^{r_1} & g_2^{r_1} & \cdots & g_l^{r_1} \\ \vdots & \vdots & \ddots & \vdots \\ g_1^{r_v} & g_2^{r_v} & \cdots & g_l^{r_v} \end{pmatrix} \tag{38}$$

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 10. *(Implicit in [12]) Let* $v = v(n)$ *be an upper-bound on the size of any group output by* $\mathcal{G}(1^n)$*. Letting* $\lambda = \log v + h$*, where* $h \in \omega(\log n)$ *is an arbitrary function, and* $l > \lambda$ *be an arbitrary value, the scheme of Definition 13, when parameterized with* l*, is strongly-*(λ, l)*-entropy circularly secure.*

Proof. We first show Condition (a) of Definition 12 for the BHHO scheme. To encrypt the bits of the secret key, we encrypt $b \in \{0,1\}$ by encrypting g^b. Let $(\mathcal{SK}, \mathcal{X})$ be an arbitrary joint distribution where $\mathcal{SK}$ is a distribution over $\{0,1\}^l$ and $\tilde{H}_\infty(\mathcal{SK}|\mathcal{X}) \geq \lambda$. Below we show a more general statement than Condition (a) of Definition 12, showing

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

where $(sk, x) \leftarrow (\mathcal{SK}, \mathcal{X})$, $par \leftarrow Param(1^n)$ and $pk = Pub(sk, par)$. Note that this also shows the (λ, l)-entropy circular security condition of the scheme (Definition 11) and that it implies Condition (a) of Definition 12, since for Condition (a) of Definition 12 we may simply set $\mathcal{X}$ to be independent of $\mathcal{SK}$, so we have

$$H_\infty(\mathcal{SK}) = \tilde{H}_\infty(\mathcal{SK}|\mathcal{X}).$$

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

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

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

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

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

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

We now briefly show that each two adjacent distributions are indistinguishable. The facts that $\mathcal{DS}_1 \equiv^c \mathcal{DS}_2$ and $\mathcal{DS}_3 \equiv^c \mathcal{DS}_4$ follow by Proposition 1, considering that each two respective distributions have the same "pattern." The facts that $\mathcal{DS}_2 \equiv^c \mathcal{DS}_3$ and $\mathcal{DS}_4 \equiv^c \mathcal{DS}_5$ follow by considering that each two respective distributions have the same pattern, that the inner product is a universal hash function and that $\tilde{H}_\infty(sk|x) \geq \log v + h$. (See Lemma 6.)

Finally, it is easy to verify the second condition of strong-(λ, l)-circular security (i.e., Condition (b), Definition 12), by considering the fact that the inner product, used in the key-generation algorithm, acts as a universal hash function.

□

7.2 From the quadratic residuosity (QR) and related assumptions

Brakerski and Goldwasser [13] 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) [34] 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 $RSAGen(1^n)$ is a PPT algorithm that on input 1^n generates a *Blum integer* N, i.e., $N = pq$ with p and q being distinct primes satisfying the condition $p, q \equiv 3 \pmod 4$. We say that the *quadratic residuosity* (QR) problem is hard under RSAGen if $\{N, U(\mathcal{QR}_N)\}_{n\in\mathbb{N}}$ is computationally indistinguishable from $\{N, U(\mathcal{QNR}_N)\}_{n\in\mathbb{N}}$, where N is generated according to $RSAGen(1^n)$.

We now describe the BG scheme.

Definition 14. *(From [13])*

- $Param(1^n)$*: 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* $pk = (sk \odot \mathbf{g})^{-1}$*;*
- $E_{pk}(b))$*: 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 also note that a similar statement to that of Theorem 10 may be given for the BG scheme, showing strong-entropy-circular-security properties of the BG scheme. We omit the details.

8 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 blackbox construction from CPA-secure encryption alone is either impossible, or very difficult to find. We mention a few open problems that arise from our work. First, it would be interesting to see if the blackbox result of [25] 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 Applebaum et al. [3] can be used to instantiate our base assumptions. Finally, as mentioned earlier, our techniques allow us to understand better the relations between certain circularly-secure schemes and DE-secure schemes. It would be interesting to see if similar connections could be proved in other settings. For example, DDH-based constructions of DE schemes satisfying *auxiliary-input security* [16] share certain design principles with those of randomized schemes satisfying *auxiliary-input leakage resilience* [20]; however, a generic connection is still not known.

Acknowledgements. We would like to thank Venkatesh Srinivasan for comments on an earlier version of this paper. We are also grateful to the anonymous reviewers for their comments that improved the presentation of this paper.

References

1. Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *Theory of Cryptography*, pages 474–495. Springer, 2009.
2. Benny Applebaum. Key-dependent message security: Generic amplification and completeness. *Journal of Cryptology*, 27(3):429–451, 2014.
3. 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*, pages 595–618. Springer, 2009.
4. Boaz Barak, Iftach Haitner, Dennis Hofheinz, and Yuval Ishai. Bounded key-dependent message security. In *Advances in Cryptology–EUROCRYPT 2010*, pages 423–444. Springer, 2010.
5. Mihir Bellare, Alexandra Boldyreva, and Adam ONeill. Deterministic and efficiently searchable encryption. In *Advances in Cryptology-CRYPTO 2007*, pages 535–552. Springer, 2007.
6. Mihir Bellare, Alexandra Boldyreva, and Jessica Staddon. Randomness re-use in multi-recipient encryption schemeas. In *Public Key CryptographyPKC 2003*, pages 85–99. Springer, 2003.
7. Mihir Bellare, Marc Fischlin, Adam ONeill, and Thomas Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In *Advances in Cryptology–CRYPTO 2008*, pages 360–378. Springer, 2008.
8. Eleanor Birrell, Kai-Min Chung, Rafael Pass, and Sidharth Telang. Randomness-dependent message security. In *Theory of Cryptography*, pages 700–720. Springer, 2013.
9. 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.
10. Alexandra Boldyreva, Serge Fehr, and Adam ONeill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *Advances in Cryptology–CRYPTO 2008*, pages 335–359. Springer, 2008.
11. Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2006.
12. Dan Boneh, Shai Halevi, Mike Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In *Advances in Cryptology–CRYPTO 2008*, pages 108–125. Springer, 2008.
13. Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability. In *Advances in Cryptology–CRYPTO 2010*, pages 1–20. Springer, 2010.
14. Zvika Brakerski, Shafi Goldwasser, and Yael Tauman Kalai. Black-box circular-secure encryption beyond affine functions. In *Theory of Cryptography*, pages 201–218. Springer, 2011.
15. Zvika Brakerski, Shafi Goldwasser, and Yael Tauman Kalai. Black-box circular-secure encryption beyond affine functions. In *Theory of Cryptography*, pages 201–218. Springer, 2011.
16. Zvika Brakerski and Gil Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. *Journal of cryptology*, 27(2):210–247, 2014.
17. 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.
18. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Advances in Cryptology (EUROCRYPT 2001)*, pages 93–118. Springer, 2001.
19. Seung Geol Choi and Hoeteck Wee. Lossy trapdoor functions from homomorphic reproducible encryption. *Information Processing Letters*, 112(20):794–798, 2012.
20. Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *Theory of Cryptography*, pages 361–381. Springer, 2010.
21. Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM journal on computing*, 38(1):97–139, 2008.
22. 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.
23. Benjamin Fuller, Adam ONeill, and Leonid Reyzin. A unified approach to deterministic encryption: New constructions and a connection to computational entropy. *Journal of Cryptology*, 28(3):671–717, 2015.
24. Yael Gertner, Tal Malkin, and Steven Myers. Towards a separation of semantic and CCA security for public key encryption. In *Theory of Cryptography*, pages 434–455. Springer, 2007.
25. Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *FOCS*, page 126. IEEE, 2001.

26. Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32. ACM, 1989.
27. Brett Hemenway and Rafail Ostrovsky. Building injective trapdoor functions from oblivious transfer. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:127, 2010.
28. Brett Hemenway and Rafail Ostrovsky. Building lossy trapdoor functions from lossy encryption. In *Advances in Cryptology-ASIACRYPT 2013*, pages 241–260. Springer, 2013.
29. Dennis Hofheinz. Circular chosen-ciphertext security with compact ciphertexts. In *Advances in Cryptology–EUROCRYPT 2013*, pages 520–536. Springer, 2013.
30. 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.
31. Steven Myers and Abhi Shelat. Bit encryption is complete. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 607–616. IEEE, 2009.
32. Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. *SIAM Journal on Computing*, 41(4):772–814, 2012.
33. Noam Nisan and David Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996.
34. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology (EUROCRYPT99)*, pages 223–238. Springer, 1999.
35. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. *SIAM Journal on Computing*, 40(6):1803–1844, 2011.
36. Omer Reingold, Luca Trevisan, and Salil Vadhan. Notions of reducibility between cryptographic primitives. In *Theory of Cryptography*, pages 1–20. Springer, 2004.
37. Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. *SIAM Journal on Computing*, 39(7):3058–3088, 2010.
38. Ron D Rothblum. On the circular security of bit-encryption. In *Theory of Cryptography*, pages 579–598. Springer, 2013.
39. Yevgeniy Vahlis. Two is a crowd? a black-box separation of one-wayness and security under correlated inputs. In *Theory of Cryptography*, pages 165–182. Springer, 2010.
40. Hoeteck Wee. Dual projective hashing and its applicationslossy trapdoor functions and more. In *Advances in Cryptology–EUROCRYPT 2012*, pages 246–262. Springer, 2012.