Trapdoors Beyond Inversion: Recoverable Information in Public-Key Cryptography

Kanad Chakravarti, Shreyas Gupta, Bhavana Kanukurthi, Girisha Shankar

Indian Institute of Science, Bengaluru, India.
{kanadc,shreyas17521,bhavana,girishabs}@iisc.ac.in

Abstract. Classical trapdoor functions recover complete preimages, whereas public-key decryption needs only designated information about an encryption input. We study this distinction through trapdoor hardcore functions (TDHFs), whose trapdoor recovers a prescribed value $h(x)$ from a public evaluation $f(x)$, while $h(x)$ remains pseudorandom given the public key and evaluation. For perfectly correct primitives, we show that adaptive TDHFs characterize IND-CCA2-secure public-key encryption. The constructive direction requires only a single recoverable bit and allows noninjective public evaluations. Extending the rejection-sampling approach, we show that adaptive pseudorandomness itself controls queries to unselected candidate evaluations, without requiring full inversion. We also prove that lossy TDHFs characterize lossy public-key encryption, identifying statistical hiding of the designated recovered value as the relevant lossiness requirement. Fiber samplability together with fiber hiding yields a parameter-preserving correspondence with lossy encryption and, through the known long-message transformation, almost-always lossy trapdoor functions when the recovered-value length exceeds the fiber-sampling randomness length. Ordinary TDHFs provide the corresponding characterization of IND-CPA encryption. Together, these results make explicit the roles of adaptive recovery, conditional sampling, and randomness costs in transformations between public-key encryption and full-inversion primitives.

Keywords: Trapdoor hardcore functions · public-key encryption · chosen-ciphertext security · lossy encryption · lossy trapdoor functions · IND-CCA

1 Introduction

Trapdoor functions are a foundational abstraction of public-key cryptography. A public key specifies an efficiently computable function, while the trapdoor enables recovery of a preimage from its public evaluation. In the injective setting, the trapdoor recovers the unique input x from $f(x)$. This full-inversion interface underlies the classical constructions of RSA and Rabin [22,21] and remains central to the design of public-key primitives.

Public-key decryption implements a more limited recovery interface. An encryption algorithm evaluates a randomized map on an input $x = (m, r)$, where m is the message and r is the encryption randomness. Decryption recovers m, but need not recover r or the complete input (m, r). Full inversion therefore supplies information that decryption does not inherently require.

This distinction already appears in the work of Goldwasser and Micali [6,7]. Their trapdoor predicates associate a public instance with a hidden bit that is recoverable using the trapdoor but computationally hidden without it. Trapdoor predicates are equivalent to semantically secure public-key encryption, showing that recovery of the complete preimage is unnecessary for basic encryption.

Trapdoor predicates and injective trapdoor functions represent two extremes of recovery: the trapdoor reveals either a single hidden bit or the complete input. Between these extremes lies a more general question that has remained largely implicit:

given $f(x)$ what information $h(x)$ should the trapdoor recover?

This recovery question is distinct from the adversarial access under which security must hold and from whether the hidden information is protected computationally or statistically.

We formalize this intermediate recovery interface through a *trapdoor hardcore function* (TDHF). Given a public evaluation of an input x, the trapdoor recovers a prescribed value $h(x)$ without recovering x, while $h(x)$ remains pseudorandom without the trapdoor. The trapdoor need not recover x or any complete preimage of the public evaluation.

This abstraction separates three aspects of trapdoor functionality:

1. *Recovery:* what information does the trapdoor recover?
2. *Adaptivity:* under what oracle access must the recovered value remain hidden?
3. *Hiding:* is the recovered value computationally pseudorandom or statistically hidden?

Recovery distinguishes TDHFs from trapdoor functions. Adaptivity strengthens the adversary's oracle access and yields the trapdoor functionality corresponding to chosen-ciphertext security. Hiding strengthens computational pseudorandomness to statistical hiding under an indistinguishable key mode.

Separating these properties reveals structure that is obscured when recovery, adaptivity, and hiding are bundled into full-inversion primitives. The framework identifies what lossy encryption loses, characterizes the recovery interface corresponding to IND-CCA2 security, and compares both interfaces formally with complete preimage recovery. We present these stronger consequences first and return afterward to the base characterization of IND-CPA-secure encryption.

1.1 Lossy Trapdoor Hardcore Functions

Our first characterization concerns the hiding dimension. We define a *lossy trapdoor hardcore function* (LTDHF) with computationally indistinguishable recov-

erable and lossy key modes. Under a recoverable key, the trapdoor recovers the designated value $h(x)$. Under a lossy key, the value $h(X)$ is statistically indistinguishable from uniform even given the public key and the public evaluation of X.

Lossy encryption is already known to yield lossy trapdoor functions under an additional parameter condition. Hemenway and Ostrovsky [9] give such a transformation when the message length exceeds the encryption-randomness length. Their result obtains full inversion from lossy encryption, but only in this long-message regime.

An LTDHF imposes a different recovery requirement: it statistically hides the designated value recovered by its trapdoor, but need not support recovery of a complete preimage. We prove the unconditional black-box equivalence

$$\boxed{\mathsf{LTDHF} \quad \Longleftrightarrow_{\mathsf{BB}} \quad \text{Lossy Public-Key Encryption.}}$$

This characterization imposes no relation between message length and encryption-randomness length. It identifies statistical hiding of the information recovered by decryption, rather than loss of the complete encryption input, as the property captured by lossy encryption.

The canonical LTDHF-to-lossy-encryption construction does not preserve this regime because it uses a complete LTDHF input as encryption randomness. We identify fiber samplability and fiber hiding as the strengthening that restores parameter preservation and therefore a route to LTDF. Fiber samplability permits efficient sampling of a correctly distributed input conditioned on a prescribed value of $h(x)$. The prescribed value determines the fiber, while fresh randomness selects a point within it. Together, fiber samplability and fiber hiding preserve the message and randomness parameters across the equivalence. We call such LTDHFs as Fiber-hiding LTDHFs.

If the hardcore output has length $\ell(\lambda)$ and fiber sampling uses $s(\lambda)$ random bits, we obtain

$$\boxed{\text{Fiber-hiding } \mathsf{LTDHF}(\ell, s) \quad \Longleftrightarrow_{\mathsf{BB}} \quad \text{Lossy Encryption}(\ell, s).}$$

$$\boxed{\text{Fiber-hiding } \mathsf{LTDHF}(\ell, s) \quad \Longrightarrow_{\mathsf{BB}} \quad \mathsf{LTDF} \quad \text{whenever } \ell(\lambda) \geq s(\lambda) + 1.}$$

Fiber samplability exposes the residual randomness within a designated-value fiber, while fiber hiding ensures that the chosen fiber remains statistically hidden. Together, they provide the parameter-preserving route to full inversion.

1.2 Minimal Trapdoor Functionality for CCA2 Security

We next consider the adaptivity dimension and the trapdoor functionality underlying IND-CCA2 security. Known constructions obtain CCA security through several forms of additional structure, including tag-based, lossy, correlated-input, and full-inversion approaches; we discuss these constructions in Section 2.

Full inversion provides a particularly direct route: it recovers the encryption randomness and thereby supports ciphertext-consistency checks. The lossy-TDF construction of Peikert and Waters [20] follows this paradigm: decryption inverts the trapdoor-function input used as the core encryption randomness and checks consistency across the ciphertext components. Hohenberger, Koppula, and Waters [10], for example, construct CPA-secure encryption with full randomness recovery from injective trapdoor functions and use this intermediate functionality to obtain IND-CCA2-secure encryption. Full randomness recovery is sufficient for IND-CCA2 security, but recovers strictly more information than ordinary decryption outputs. It therefore does not identify the recovery interface intrinsic to IND-CCA2 security. Known black-box separation results already distinguish full inversion from the functionality needed for CPA-secure encryption [5,4]. This motivates the corresponding question for chosen-ciphertext security:

> *What trapdoor functionality exactly characterizes CCA security in a black-box sense, without requiring recovery of the complete input or all of the encryption randomness?*

We present *adaptive trapdoor hardcore functions* (ATDHFs) as a characterization of this functionality. An adaptive TDHF adversary may make partial-inversion queries both before and after receiving the challenge, except on the challenge evaluation itself.
Adaptive TDHFs exactly capture fully adaptive chosen-ciphertext security:

$$\mathsf{ATDHF} \iff_{\mathsf{BB}} \text{IND-CCA2-secure Public-Key Encryption.}$$

Our construction follows the rejection-sampling framework of Kiltz, Mohassel, and O'Neill [12], but replaces their injective evaluation with a potentially noninjective ATDHF evaluation. Consequently, the reduction must protect several sampled evaluations although only the selected challenge ciphertext is excluded from decryption queries. We prove that adaptive pseudorandomness itself prevents an adversary from querying an independently sampled fresh evaluation; a standard hybrid extends this guarantee to all sampled evaluations. Thus, adaptive pseudorandomness, rather than injectivity, is sufficient for the CCA2 proof.

The known full-recovery route [20,10] and our characterization can therefore be summarized as

$$\begin{aligned} \mathsf{LTDF} &\Longrightarrow_{\mathsf{BB}} \text{IND-CCA2 PKE (via randomness recovery),} \\ \mathsf{iTDF} &\Longrightarrow_{\mathsf{BB}} \text{CPA-RR} \Longrightarrow_{\mathsf{BB}} \text{IND-CCA2 PKE,} \\ \mathsf{ATDHF} &\iff_{\mathsf{BB}} \text{IND-CCA2 PKE.} \end{aligned}$$

The first two routes obtain CCA2 security from primitives that recover a complete randomness-bearing input. The characterization shows that CCA2 security itself corresponds to the weaker interface of adaptive recovery of a designated hidden value. This distinction is formal in the black-box setting. Myers and Shelat observe that the oracle underlying the Gertner–Malkin–Reingold [5] separation already provides CCA2-secure one-bit encryption while ruling out poly-to-one

trapdoor functions [19]. Combining their black-box one-bit-to-many-bit transformation with our characterization yields

$$\boxed{\mathsf{iTDF} \implies \mathsf{ATDHF}, \quad \mathsf{ATDHF} \not\Longrightarrow \mathsf{iTDF}.}$$

Thus, ATDHF lies strictly below injective trapdoor functions in the stated black-box construction order while still characterizing IND-CCA2-secure encryption.

Connection to Hohenberger, Koppula, and Waters [10]. The breakthrough of Hohenberger, Koppula, and Waters showed that injective trapdoor functions suffice for IND-CCA2-secure encryption. Viewed through our framework, their construction highlights the power of full trapdoor inversion: by first obtaining encryption with randomness recovery, the trapdoor recovers the complete encryption input, including all encryption randomness. This stronger recovery capability is then exploited to obtain ciphertext validation and, ultimately, CCA2 security. Their result therefore establishes a powerful sufficient route to CCA2 security from full inversion. Our characterization addresses a different question: what recovery functionality corresponds directly to CCA2 security itself? ATDHFs show that adaptive recovery of a designated value is sufficient to characterize CCA2 encryption, without requiring recovery of the entire encryption input. Consequently, existing constructions demonstrate that full inversion suffices for CCA2 security, but they do not establish that full inversion is necessary. We now summarise our key contribution in the context of CCA2 security.

Our framework revisits full inversion as the organizing abstraction for this search. By studying adaptive recovery of designated information—even a single bit—and explicitly analyzing the associated sampling and hiding structure, we formulate a more fine-grained constructive agenda for the trapdoor foundations of chosen-ciphertext security.

The preceding characterizations concern the two stronger dimensions of the framework. Adaptivity leads from basic designated-value recovery to the functionality corresponding to chosen-ciphertext security, while lossiness strengthens computational pseudorandomness to statistical hiding. Both characterizations build on a more fundamental recovery primitive. We now return to this base notion and show that ordinary trapdoor hardcore functions characterize IND-CPA-secure public-key encryption.

1.3 The Base Characterization

The base characterization completes the framework. We prove the black-box equivalence

$$\boxed{\mathsf{TDHF} \quad \Longleftrightarrow_{\mathsf{BB}} \quad \text{IND-CPA-secure Public-Key Encryption.}}$$

The correspondence is direct. Given a TDHF, the pseudorandom value $h(x)$ serves as a one-time pad, while the trapdoor enables decryption. Conversely,

a public-key ciphertext can be viewed as the public evaluation of $x = (m, r)$, where the trapdoor recovers the designated value $h(m, r) = m$. For single-bit outputs, this recovers the classical equivalence between trapdoor predicates and semantically secure public-key encryption.

This equivalence serves as the base case of the framework. Adaptivity strengthens the adversary's oracle access, while lossiness strengthens the hiding guarantee; neither changes the information recovered by the trapdoor.

1.4 Discussion

The characterizations above separate three properties that are often combined in trapdoor primitives. Recovery determines what information the trapdoor reveals. Adaptivity determines the oracle access under which that information remains hidden. Hiding determines whether the recovered value is protected computationally or statistically. Full inversion, adaptive security, and statistical hiding are therefore distinct strengthening directions rather than successive formulations of a single notion.

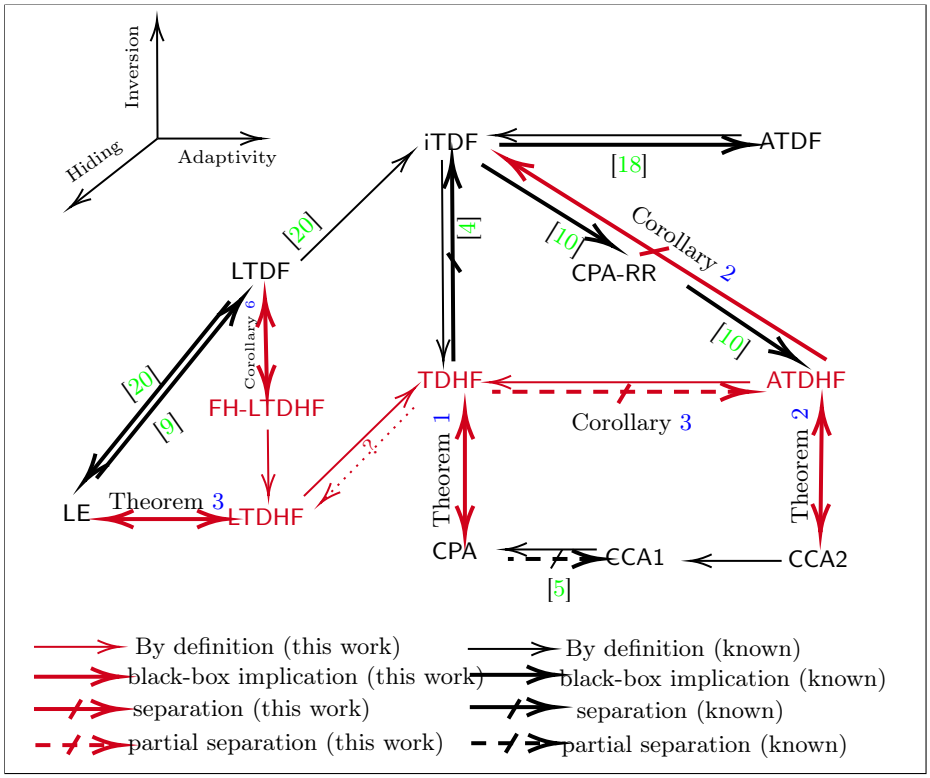

Fig. 1: Relationships among TDHFs and other Cryptomania primitives

One-way functions and pseudorandom generators isolate basic computational resources underlying Minicrypt. Analogously, TDHFs isolate the trapdoor functionality underlying public-key encryption: a public evaluation hides a designated value that the trapdoor recovers. The adaptive and lossy characterizations show that this recovery interface persists under chosen-ciphertext security and statistical hiding. Together, these results identify public evaluation together with trapdoor recovery of a hidden designated value, rather than full inversion itself, as a basic trapdoor resource of cryptomania.

TDHF and KEM/PKE. TDHFs admit a direct Key Encapsulation Mechanism (KEM) interpretation: under matching correctness and sampling conventions, making encapsulation randomness explicit yields the TDHF interface, and adaptive TDHF security corresponds to the standard CCA key-indistinguishability experiment. We therefore do not claim a fundamentally different encapsulation functionality; rather, we use this formulation to study recoverable information and its sampling structure. In particular, our CCA2 characterization applies even when the trapdoor recovers only a single pseudorandom bit, for which standard KEM/DEM composition does not directly provide a suitable symmetric encryption and authentication key. Building on the rejection-sampling approach of Kiltz, Mohassel, and O'Neill, we handle potentially noninjective public evaluations by showing that adaptive pseudorandomness suffices to control queries to unselected candidate evaluations, without requiring complete input recovery. In the lossy setting, we distinguish efficient sampling within a prescribed designated-value fiber from statistical hiding of the selected fiber. Together, these properties yield a parameter-preserving correspondence with lossy encryption and, in the long-message regime, a route to almost-always lossy trapdoor functions. The intended value of the TDHF viewpoint thus lies in these proof extensions and parameter-sensitive structural comparisons. Likewise, while the basic TDHF–PKE equivalence is a direct reformulation, viewing the framework merely as a recasting of PKE overlooks its intended contribution: understanding how adaptive recovery, fiber sampling, and statistical hiding support stronger security and parameter-preserving transformations.

2 Related Work

Our work builds on three lines of research: trapdoor primitives for chosen-ciphertext security, lossy cryptography, and relaxations of full trapdoor inversion. We compare these works according to their public evaluation interface, trapdoor recovery interface, and security guarantee.

Trapdoor primitives for chosen-ciphertext security.

Several approaches obtain chosen-ciphertext security by adding structure that enables decryption to reject malformed or inconsistent ciphertexts. Tag-based and identity-based approaches bind ciphertext generation to an auxiliary tag or identity and support generic CCA transformations [11,2]. Correlated-product-secure trapdoor functions preserve one-wayness under related inputs [23].

Other generic CPA-to-CCA transformations use hinting pseudorandom generators or suitable key-dependent-message security [16,13].

Full randomness recovery provides a particularly direct consistency mechanism. The lossy-TDF construction of Peikert and Waters [20] recovers the randomness-bearing trapdoor-function input and uses all-but-one structure to enforce consistency across ciphertext components. Hohenberger, Koppula, and Waters [10] construct CCA-secure encryption from injective trapdoor functions through the intermediate functionality of CPA-secure encryption with full randomness recovery. At the primitive level, Mao and Yi [18] study transformations from injective trapdoor functions to adaptive trapdoor functions.

Adaptive TDHFs isolate a weaker recovery interface. The trapdoor recovers only the value corresponding to the plaintext, while partial-inversion access models decryption queries. The equivalence with IND-CCA2 identifies adaptive designated-value recovery as an equivalent trapdoor interface for fully adaptive chosen-ciphertext security. Full inversion and full randomness recovery remain sufficient construction tools, but are strictly stronger than the recovery interface required by the characterization in the stated black-box construction order.

Lossy primitives.

Peikert and Waters introduced lossy and all-but-one trapdoor functions [20]. A lossy trapdoor function has computationally indistinguishable injective and lossy modes, and the lossy evaluation has bounded image size. This framework gives constructions of CCA-secure encryption, oblivious transfer, and collision-resistant hashing.

Lossy encryption instead requires ciphertexts under a lossy key to statistically hide the plaintext. Hemenway and Ostrovsky [9] show that lossy encryption with message length $\ell(\lambda)$ and randomness length $s(\lambda)$ yields a lossy trapdoor function when $\ell(\lambda) \geq s(\lambda) + 1$.

Our LTDHFs separate loss of the trapdoor-recovered value from loss of the complete input. An LTDHF statistically hides only the designated value recovered by the trapdoor; its public evaluation need not lose all information about the complete input. This gives an unconditional equivalence between LTDHFs and lossy encryption. The canonical equivalence does not preserve the message-to-randomness ratio required by the Hemenway–Ostrovsky transformation. Fiber samplability provides efficient access to prescribed designated-value fibers, while fiber hiding requires their lossy evaluations to be statistically indistinguishable. Together, these properties preserve the message and randomness parameters.

Trapdoor predicates and partial trapdoor recovery.

Goldwasser and Micali introduced trapdoor predicates and established their connection with semantically secure public-key encryption [6,7]. Gertner, Malkin, and Reingold showed that trapdoor functions cannot be obtained from trapdoor predicates through a broad class of black-box constructions [5]. These results separate recovery of a hidden predicate from recovery of a complete preimage.

TDHFs extend this recovery interface from a hidden bit to a prescribed polynomial-length value $h(x)$. This extension makes the recovery interface compatible with modern encryption notions: adaptive TDHFs model decryption-

Primitive	Public interface	Trapdoor output	Security or loss
Adaptive TDF	$\mathsf{Eval}(\mathsf{pk}, x)$	x	Adaptive one-wayness
Adaptive trapdoor relation	Sample (x, y)	x	Tag-based adaptive one-wayness
Lossy trapdoor relation	$\mathsf{Eval}(\mathsf{pk}, x)$	Injective $H(x)$	Bounded evaluation image
ATDHF	$\mathsf{Eval}(\mathsf{pk}, x)$	$h(x)$	Adaptive pseudorandomness
LTDHF	$\mathsf{Eval}(\mathsf{pk}, x)$	$h(x)$	Statistical hiding of $h(X)$

Table 1: Comparison with trapdoor-relation primitives.

oracle access, while lossy TDHFs model statistical hiding of the value recovered by decryption.

Trapdoor relations.

Wee introduced adaptive trapdoor relations as a relaxation of adaptive trapdoor functions [24]. The public algorithm need not evaluate on arbitrary inputs; it is sufficient to sample a random witness together with its public image. The trapdoor still recovers the complete sampled witness, and security is defined using tag-restricted inversion queries.

Xue, Lu, Li, and Liu introduced lossy trapdoor relations [25]. Their trapdoor recovers a publicly computable injective map $H(x)$ of the input, while lossiness is measured through the image size of the public evaluation.

Trapdoor relations and TDHFs relax different interfaces. Trapdoor relations weaken public evaluation while retaining recovery of a complete witness or an injective encoding of it. TDHFs retain public evaluation on arbitrary inputs but require recovery only of a designated value $h(x)$, which may be many-to-one. Their lossy variants also impose different requirements. Lossy trapdoor relations bound the image of the public evaluation, whereas LTDHFs require $h(X)$ itself to be statistically hidden given the lossy key and evaluation.

3 Preliminaries

3.1 Notation

1. The security parameter is denoted by $\lambda \in \mathbb{N}$ and 1^λ as its unary representation.
2. For a finite set $\mathcal{S}$, we write $x \xleftarrow{\$} \mathcal{S}$ to denote that x is sampled uniformly from $\mathcal{S}$.
3. For a probabilistic algorithm $\mathcal{A}$, the notation $y \leftarrow \mathcal{A}(x)$ denotes that $\mathcal{A}$ is executed on input x using fresh random coins and outputs y.

4. For an oracle $\mathcal{O}$, we write $A^{\mathcal{O}}(x)$ for an execution of $\mathcal{A}$ on input x with oracle access to $\mathcal{O}$; Unless stated otherwise, all algorithms run in probabilistic polynomial time and make at most polynomially many oracle queries.
5. A function $\mathsf{negl} : \mathbb{N} \to [0,1]$ is negligible if for every positive polynomial p, there exists $n_0 \in \mathbb{N}$ such that $\mathsf{negl}(n) < \frac{1}{p(n)}$ for every $n > n_0$.
6. We use PPT as an abbreviation for probabilistic polynomial time.
7. For two distribution ensembles $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$, we write $X \approx_c Y$ if they are computationally indistinguishable.
8. We write $X \approx_s Y$ if their statistical distance $\frac{1}{2}\sum_\omega |\Pr[X = \omega] - \Pr[Y = \omega]|$ denoted by $\Delta(X, Y)$ is negligible in λ. When there is no ambiguity, we may write $\approx$ for statistical indistinguishability.
9. We use $X \equiv Y$ to denote that distributions X, Y are identical.

3.2 Public-Key Encryption

A public-key encryption scheme $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}_\lambda$, randomness space $\mathcal{R}_\lambda$, and ciphertext space $\mathcal{C}_\lambda$ consists of the following polynomial-time algorithms.

- **Key generation:** $\mathsf{KGen}(1^\lambda) \to (\mathsf{pke.pk}, \mathsf{pke.sk})$. The randomized key-generation algorithm takes the security parameter as input and outputs a public key $\mathsf{pke.pk}$ and a secret key $\mathsf{pke.sk}$.
- **Encryption:** $\mathsf{Enc}(\mathsf{pke.pk}, m; r) \to \mathsf{ct}$. The deterministic encryption algorithm takes a public key $\mathsf{pke.pk}$, a message $m \in \mathcal{M}_\lambda$, and randomness $r \in \mathcal{R}_\lambda$, and outputs a ciphertext $\mathsf{ct} \in \mathcal{C}_\lambda$. When the random coins are sampled internally, we write $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pke.pk}, m)$.
- **Decryption:** $\mathsf{Dec}(\mathsf{pke.sk}, \mathsf{ct}) \to m$ or $\perp$. The deterministic decryption algorithm takes the secret key and a ciphertext and outputs a message in $\mathcal{M}_\lambda$ or the failure symbol $\perp$.

Definition 1 (Perfect correctness of PKE). *A public-key encryption scheme is perfectly correct if, for every $\lambda \in \mathbb{N}$, every key pair $(\mathsf{pke.pk}, \mathsf{pke.sk}) \in \mathrm{Supp}\left(\mathsf{KGen}(1^\lambda)\right)$, every message $m \in \mathcal{M}_\lambda$, and every randomness value $r \in \mathcal{R}_\lambda$, it holds that*

$$\mathsf{Dec}\left(\mathsf{pke.sk}, \mathsf{Enc}(\mathsf{pke.pk}, m; r)\right) = m.$$

3.2.1 Security of PKE Depending on the power of adversary we have different notions of security for the PKE. For a CPA secure scheme the adversary has access only to the encryption oracle whereas for CCA secure scheme, the adversary can additionally have access to decryption oracle.

For $\mathsf{atk} \in \{\mathsf{CPA}, \mathsf{CCA\text{-}1}, \mathsf{CCA\text{-}2}\}$, an adversary is a pair of PPT algorithms $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. For a secret key $\mathsf{pke.sk}$, define the decryption oracle

$$\mathcal{O}_{\mathsf{pke.sk}}(\mathsf{ct}) := \mathsf{Dec}(\mathsf{pke.sk}, \mathsf{ct}).$$

For a challenge ciphertext $\mathsf{ct}^\star$, define the restricted decryption oracle

$$\mathcal{O}^{\mathsf{ct}^\star}_{\mathsf{pke.sk}}(\mathsf{ct}) := \begin{cases} \bot, & \text{if } \mathsf{ct} = \mathsf{ct}^\star, \\ \mathsf{Dec}(\mathsf{pke.sk}, \mathsf{ct}), & \text{otherwise.} \end{cases}$$

For $b \in \{0,1\}$, define the experiment $\mathsf{Expt}^{\mathsf{ind\text{-}atk}\text{-}b}_{\mathsf{PKE},\mathcal{A}}(\lambda)$ as follows.

function $\mathsf{Expt}^{\mathsf{ind\text{-}atk}\text{-}b}_{\mathsf{PKE},\mathcal{A}}\ (1^\lambda)$
  $(\mathsf{pke.pk}, \mathsf{pke.sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
  if $\mathsf{atk} = \mathsf{CPA}$ **then**
    $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\lambda, \mathsf{pke.pk})$
  else
    $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{pke.sk}}}(1^\lambda, \mathsf{pke.pk})$
  end if
  if $m_0, m_1 \notin \mathcal{M}_\lambda$ or $|m_0| \neq |m_1|$ **then**
    return 0
  end if
  $\mathsf{ct}^\star \leftarrow \mathsf{Enc}(\mathsf{PKE.pk}, m_b)$
  if $\mathsf{atk} = \mathsf{CCA2}$ **then**
    $b' \leftarrow \mathcal{A}_2^{\mathcal{O}^{\mathsf{ct}^\star}_{\mathsf{pke.sk}}}(\mathsf{ct}^\star, \mathsf{st})$
  else
    $b' \leftarrow \mathcal{A}_2(\mathsf{ct}^\star, \mathsf{st})$
  end if
  return b'
end function

Definition 2 (IND-CPA, IND-CCA1, and IND-CCA2 security). *For* $\mathsf{atk} \in \{\mathsf{CPA}, \mathsf{CCA\text{-}1}, \mathsf{CCA\text{-}2}\}$, *define the advantage of an adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *by*

$$\mathsf{Adv}^{\mathsf{ind\text{-}atk}}_{\mathsf{PKE},\mathcal{A}}(\lambda) := \left| \Pr\left[\mathsf{Expt}^{\mathsf{ind\text{-}atk}\text{-}0}_{\mathsf{PKE},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{ind\text{-}atk}\text{-}1}_{\mathsf{PKE},\mathcal{A}}(\lambda) = 1\right] \right|$$

A public-key encryption scheme is IND-atk *secure if, for every* PPT *adversary* $\mathcal{A}$*, the function* $\mathsf{Adv}^{\mathsf{ind\text{-}atk}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$ *is negligible in* λ.

In particular:

- IND-CPA provides no decryption-oracle access;
- IND-CCA1 provides decryption-oracle access only before challenge generation; and
- IND-CCA2 provides decryption-oracle access both before and after challenge generation, except that the exact challenge ciphertext may not be submitted after challenge generation.

3.3 Trapdoor Functions

We first define a general one-way trapdoor function, where inversion is required to recover some valid preimage. We then define injective trapdoor functions, where the preimage is unique.

Let $\mathcal{D}_\lambda$ and $\mathcal{R}_\lambda$ denote efficiently recognizable input and output spaces, respectively, and assume that the uniform distribution over $\mathcal{D}_\lambda$ can be sampled efficiently.

Definition 3 (Trapdoor function). *A trapdoor function* $\mathsf{TDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$ *consists of the following polynomial-time algorithms.*

- ***Key generation:*** $\mathsf{Gen}(1^\lambda) \to (\mathsf{tdf.pk}, \mathsf{tdf.sk})$. *The randomized key-generation algorithm outputs a public evaluation key* $\mathsf{tdf.pk}$ *and a secret trapdoor* $\mathsf{tdf.sk}$.
- ***Evaluation:*** $\mathsf{Eval}(\mathsf{tdf.pk}, x) \to y$. *The deterministic evaluation algorithm maps* $x \in \mathcal{D}_\lambda$ *to* $y \in \mathcal{R}_\lambda$.
- ***Inversion:*** $\mathsf{Inv}(\mathsf{tdf.sk}, y) \to x'$ *or* $\perp$. *The deterministic inversion algorithm outputs a candidate preimage* $x' \in \mathcal{D}_\lambda$ *or* $\perp$.

Correctness. *For every* $\lambda \in \mathbb{N}$, *every key pair* $(\mathsf{tdf.pk}, \mathsf{tdf.sk}) \in \mathrm{Supp}\left(\mathsf{Gen}(1^\lambda)\right)$, *and every input* $x \in \mathcal{D}_\lambda$, *it holds that*

$$\mathsf{Eval}\left(\mathsf{tdf.pk}, \mathsf{Inv}\left(\mathsf{tdf.sk}, \mathsf{Eval}(\mathsf{tdf.pk}, x)\right)\right) = \mathsf{Eval}(\mathsf{tdf.pk}, x).$$

One-wayness. *For every* PPT *adversary* $\mathcal{A}$,

$$\Pr\left[\mathsf{Eval}(\mathsf{tdf.pk}, x') = \mathsf{Eval}(\mathsf{tdf.pk}, x) \;\middle|\; \begin{array}{r}(\mathsf{tdf.pk}, \mathsf{tdf.sk}) \leftarrow \mathsf{Gen}(1^\lambda); \quad x \xleftarrow{\$} \mathcal{D}_\lambda; \\ x' \leftarrow \mathcal{A}\left(1^\lambda, \mathsf{tdf.pk}, \mathsf{Eval}(\mathsf{tdf.pk}, x)\right)\end{array}\right]$$

is negligible in λ.

Polynomial preimage bound (Poly-to-one)[1]*: There exists a polynomial* $p(\cdot)$ *such that for every* $\lambda \in \mathbb{N}$, *every key pair* $(\mathsf{tdf.pk}, \mathsf{tdf.sk}) \in \mathsf{Supp}(\mathsf{KGen}(1^\lambda))$, *and every* $y \in \mathcal{R}_\lambda$, *the number of valid preimages is bounded by:*

$$|\{x \in \mathcal{D}_\lambda \mid \mathsf{Eval}(\mathsf{tdf.pk}, x) = y\}| \leq p(\lambda)$$

Definition 4 (Injective trapdoor function). *An injective trapdoor function* $\mathsf{iTDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$ *is a trapdoor function according to Definition 3 such that, for every* $\lambda \in \mathbb{N}$ *and every key pair* $(\mathsf{tdf.pk}, \mathsf{tdf.sk}) \in \mathrm{Supp}\left(\mathsf{Gen}(1^\lambda)\right)$, *the function* $\mathsf{Eval}(\mathsf{tdf.pk}, \cdot) : \mathcal{D}_\lambda \to \mathcal{R}_\lambda$ *is injective.*

Equivalently, for every $x_0, x_1 \in \mathcal{D}_\lambda$,

$$\mathsf{Eval}(\mathsf{tdf.pk}, x_0) = \mathsf{Eval}(\mathsf{tdf.pk}, x_1) \implies x_0 = x_1.$$

[1] Following standard conventions, we restrict the preimage size to be polynomial, as unrestricted many-to-one trapdoor functions with super-polynomial preimages can be constructed from general one-way functions [1]. This distinction is also necessary while extending cryptographic primitives to CCA security settings, where bit-encryption completeness results [19] demonstrate that underlying primitive bounds must be carefully constrained.

3.3.1 Adaptive Trapdoor Functions An adaptive trapdoor function strengthens the one-wayness of an injective trapdoor function by allowing the adversary to invert arbitrary outputs other than the challenge output. The trapdoor function must remain one-way even in the presence of this restricted inversion oracle [12].

Let $\mathsf{ATDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$ be an injective trapdoor function with input space $\mathcal{D}_\lambda$ and output space $\mathcal{R}_\lambda$. For a secret key sk and a challenge output $y^\star$, define the restricted inversion oracle

$$\mathcal{O}_{\mathsf{sk}}^{y^\star}(y) := \begin{cases} \bot, & \text{if } y = y^\star, \\ \mathsf{Inv}(\mathsf{sk}, y), & \text{otherwise.} \end{cases}$$

For a PPT adversary $\mathcal{A}$, define the adaptive one-wayness experiment as follows.

```
function Expt^aow_{ATDF,A}(1^λ)
    (pk, sk) ← Gen(1^λ)
    x* ←$ D_λ
    y* ← Eval(pk, x*)
    x' ← A^{O^{y*}_{sk}}(1^λ, pk, y*)
    if Eval(pk, x') = y* then
        return 1
    else
        return 0
    end if
end function
```

Definition 5 (Adaptive trapdoor function). *An injective trapdoor function* $\mathsf{ATDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$ *is an* adaptive trapdoor function *if, for every* PPT *adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\mathsf{aow}}_{\mathsf{ATDF},\mathcal{A}}(\lambda) := \Pr\left[\mathsf{Expt}^{\mathsf{aow}}_{\mathsf{ATDF},\mathcal{A}}(1^\lambda) = 1\right]$$

is negligible in λ*.*

3.4 Lossy Primitives

Definition 6 (Lossy trapdoor function). *An* (n, k)*-lossy trapdoor function* $\mathsf{LTDF} = (\mathsf{Gen}_{\mathsf{inj}}, \mathsf{Gen}_{\mathsf{lossy}}, \mathsf{Eval}, \mathsf{Inv})$ *has input space* $\mathcal{D}_\lambda = \{0,1\}^{n(\lambda)}$*, where* n *and* k *are polynomially bounded and* $0 \le k(\lambda) \le n(\lambda)$*.*

It consists of the following polynomial-time algorithms.

- ***Injective-mode key generation:*** $\mathsf{Gen}_{\mathsf{inj}}(1^\lambda) \to (\mathsf{pk}_i, \mathsf{sk}_i)$.
- ***Lossy-mode key generation:*** $\mathsf{Gen}_{\mathsf{lossy}}(1^\lambda) \to \mathsf{pk}_\ell$.
- ***Evaluation:*** $\mathsf{Eval}(\mathsf{pk}, x) \to y$*. The same deterministic evaluation algorithm is used with keys generated in either mode.*

– ***Inversion:*** $\mathsf{Inv}(\mathsf{sk}_i, y) \to x$ *or* $\perp$.

The following properties are required.

Injective-mode correctness. *For every key pair* $(\mathsf{pk}_i, \mathsf{sk}_i) \in \mathrm{Supp}\left(\mathsf{Gen}_{\mathsf{inj}}(1^\lambda)\right)$, *the map* $\mathsf{Eval}(\mathsf{pk}_i, \cdot)$ *is injective, and for every* $x \in \{0,1\}^{n(\lambda)}$,

$$\mathsf{Inv}\left(\mathsf{sk}_i, \mathsf{Eval}(\mathsf{pk}_i, x)\right) = x.$$

Mode indistinguishability.

$$\left\{\mathsf{pk}_i \;\middle|\; (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}_{\mathsf{inj}}(1^\lambda)\right\}_{\lambda\in\mathbb{N}} \approx_c \left\{\mathsf{pk}_\ell \;\middle|\; \mathsf{pk}_\ell \leftarrow \mathsf{Gen}_{\mathsf{lossy}}(1^\lambda)\right\}_{\lambda\in\mathbb{N}}$$

Lossiness. *For every lossy public key* $\mathsf{pk}_\ell \in \mathrm{Supp}\left(\mathsf{Gen}_{\mathsf{lossy}}(1^\lambda)\right)$, *the image of the function* $\mathsf{Eval}(\mathsf{pk}_\ell, \cdot)$ *has size at most* $2^{n(\lambda)-k(\lambda)}$. *That is,*

$$\left|\left\{\mathsf{Eval}(\mathsf{pk}_\ell, x) : x \in \{0,1\}^{n(\lambda)}\right\}\right| \leq 2^{n(\lambda)-k(\lambda)}.$$

Remark 1. Let $X \xleftarrow{\$} \{0,1\}^{n(\lambda)}$ and $Y = \mathsf{Eval}(\mathsf{pk}_\ell, X)$. The bounded-image condition implies $\widetilde{H}_\infty(X \mid Y) \geq k(\lambda)$, where $\widetilde{H}_\infty$ denotes average conditional min-entropy.

Definition 7 (Lossy public-key encryption). *A lossy public-key encryption scheme* $\mathsf{LPKE} = (\mathsf{Gen}_{\mathsf{inj}}, \mathsf{Gen}_{\mathsf{lossy}}, \mathsf{Enc}, \mathsf{Dec})$ *with message space* $\mathcal{M}_\lambda$, *randomness space* $\mathcal{R}_\lambda$, *and ciphertext space* $\mathcal{C}_\lambda$ *consists of the following polynomial-time algorithms.*

– ***Recoverable-mode key generation:*** $\mathsf{Gen}_{\mathsf{inj}}(1^\lambda) \to (\mathsf{pk}_i, \mathsf{sk}_i)$.

– ***Lossy-mode key generation:*** $\mathsf{Gen}_{\mathsf{lossy}}(1^\lambda) \to \mathsf{pk}_\ell$.

– ***Encryption:*** $\mathsf{Enc}(\mathsf{pk}, m; r) \to \mathsf{ct}$. *The same encryption algorithm is used with public keys generated in either mode.*

– ***Decryption:*** $\mathsf{Dec}(\mathsf{sk}_i, \mathsf{ct}) \to m$ *or* $\perp$. *Decryption is defined only for recoverable-mode secret keys.*

The following properties are required.

Perfect correctness. *For every key pair* $(\mathsf{pk}_i, \mathsf{sk}_i) \in \mathrm{Supp}\left(\mathsf{Gen}_{\mathsf{inj}}(1^\lambda)\right)$, *every message* $m \in \mathcal{M}_\lambda$, *and every randomness value* $r \in \mathcal{R}_\lambda$,

$$\mathsf{Dec}\left(\mathsf{sk}_i, \mathsf{Enc}(\mathsf{pk}_i, m; r)\right) = m.$$

Mode indistinguishability.

$$\left\{\mathsf{pk}_i \;\middle|\; (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}_{\mathsf{inj}}(1^\lambda)\right\}_{\lambda\in\mathbb{N}} \approx_c \left\{\mathsf{pk}_\ell \;\middle|\; \mathsf{pk}_\ell \leftarrow \mathsf{Gen}_{\mathsf{lossy}}(1^\lambda)\right\}_{\lambda\in\mathbb{N}}$$

Ciphertext lossiness. *There exists a negligible function* ε *such that, for every security parameter* λ, *every lossy public key* $\mathsf{pk}_\ell \in \mathrm{Supp}\left(\mathsf{Gen}_{\mathsf{lossy}}(1^\lambda)\right)$, *and every pair of equal-length messages* $m_0, m_1 \in \mathcal{M}_\lambda$, *we have*

$$\Delta\left(\mathsf{Enc}(\mathsf{pk}_\ell, m_0; R_0), \mathsf{Enc}(\mathsf{pk}_\ell, m_1; R_1)\right) \leq \varepsilon(\lambda),$$

where $R_0, R_1 \xleftarrow{\$} \mathcal{R}_\lambda$ *are sampled independently.*

4 Trapdoor Hardcore Functions

We introduce trapdoor hardcore functions (TDHFs) as a multi-bit extension of trapdoor predicates. Informally, a TDHF consists of a public evaluation function f, a designated function h, and a trapdoor. Given $f(x)$ and the trapdoor, one can recover $h(x)$, but the trapdoor need not enable recovery of x itself. Without the trapdoor, $h(x)$ is computationally indistinguishable from a uniformly random string, even given $f(x)$.

Let λ be the security parameter. For every λ, let $\mathcal{D}_\lambda$ and $\mathcal{R}_\lambda$ denote the input space and the evaluation range, respectively. Let $\mathcal{R}'_\lambda = \{0,1\}^{\ell(\lambda)}$ denote the hardcore range, where $\ell\colon \mathbb{N} \to \mathbb{N}$ is polynomially bounded. We require that the spaces $\mathcal{D}_\lambda$ and $\mathcal{R}_\lambda$ be efficiently recognizable and that the uniform distribution over $\mathcal{D}_\lambda$ be efficiently sampleable.

Definition 8 (Trapdoor hardcore function). *A trapdoor hardcore function is a tuple of probabilistic polynomial-time algorithms* $\mathsf{TDHF} = \big(\mathsf{KGen}, \mathsf{Eval}, \mathsf{hEval}, \mathsf{pInv}\big)$ *with the following syntax.*

- $\mathsf{KGen}(1^\lambda) \to (\mathsf{tdhf.pk}, \mathsf{tdhf.sk})$*: The randomized key-generation algorithm takes as input the unary representation of the security parameter and outputs a public evaluation key* $\mathsf{tdhf.pk}$ *and a secret trapdoor* $\mathsf{tdhf.sk}$*.*
- $\mathsf{Eval}(\mathsf{tdhf.pk}, x) \to y$*: The deterministic evaluation algorithm takes as input the public key* $\mathsf{tdhf.pk}$ *and an input* $x \in \mathcal{D}_\lambda$*, and outputs an evaluation* $y \in \mathcal{R}_\lambda$*.*
- $\mathsf{hEval}(\mathsf{tdhf.pk}, x) \to z$*: The deterministic hardcore-evaluation algorithm takes as input the public key* $\mathsf{tdhf.pk}$ *and an input* $x \in \mathcal{D}_\lambda$*, and outputs a hardcore value* $z \in \mathcal{R}'_\lambda$*.*
- $\mathsf{pInv}(\mathsf{tdhf.sk}, y) \to z'$ *or* $\perp$*: The deterministic algorithm takes as input the secret trapdoor* $\mathsf{tdhf.sk}$ *and an evaluation* $y \in \mathcal{R}_\lambda$*, and outputs a value* $z' \in \mathcal{R}'_\lambda$ *or a distinguished failure symbol* $\perp$*.*

Definition 9 (Perfect correctness). *A* TDHF *is perfectly correct if, for every security parameter* $\lambda \in \mathbb{N}$*, every key pair* $(\mathsf{tdhf.pk}, \mathsf{tdhf.sk}) \in \mathrm{Supp}\left(\mathsf{KGen}(1^\lambda)\right)$*, and every input* $x \in \mathcal{D}_\lambda$*, it holds that*

$$\mathsf{pInv}\left(\mathsf{tdhf.sk}, \mathsf{Eval}(\mathsf{tdhf.pk}, x)\right) = \mathsf{hEval}(\mathsf{tdhf.pk}, x).$$

4.1 TDHF Security

We require the hardcore value of a uniformly sampled input to be computationally indistinguishable from a uniformly random string, even given the public key and the public evaluation of the input.

The requirement that the hardcore value be pseudorandom, rather than merely difficult to compute, is important for polynomial-length outputs. For a single-bit output, unpredictability and pseudorandomness are equivalent. For a multi-bit output, however, the complete output may be difficult to predict while still being efficiently distinguishable from uniform.

Definition 10 (TDHF security). *A perfectly correct* TDHF *is secure if for every PPT adversary $\mathcal{A}$, the advantage*

$$\mathsf{Adv}^{\mathsf{tdhf}}_{\mathsf{TDHF},\mathcal{A}}(\lambda) := \left| \begin{array}{l} \Pr\left[\mathcal{A}\begin{pmatrix} 1^\lambda, \mathsf{tdhf.pk}, \\ \mathsf{Eval}(\mathsf{tdhf.pk}, x), \\ \mathsf{hEval}(\mathsf{tdhf.pk}, x) \end{pmatrix} = 1 \;\middle|\; \begin{array}{l} (\mathsf{tdhf.pk}, \mathsf{tdhf.sk}) \xleftarrow{\$} \mathsf{KGen}(1^\lambda) \\ x \xleftarrow{\$} \mathcal{D}_\lambda \end{array}\right] \\ - \Pr\left[\mathcal{A}\begin{pmatrix} 1^\lambda, \mathsf{tdhf.pk}, \\ \mathsf{Eval}(\mathsf{tdhf.pk}, x), \\ u \end{pmatrix} = 1 \;\middle|\; \begin{array}{l} (\mathsf{tdhf.pk}, \mathsf{tdhf.sk}) \xleftarrow{\$} \mathsf{KGen}(1^\lambda) \\ x \xleftarrow{\$} \mathcal{D}_\lambda, \\ u \xleftarrow{\$} \{0,1\}^{\ell(\lambda)} \end{array}\right] \end{array} \right|$$

is negligible in λ.

4.2 Equivalence with CPA-Secure Public-Key Encryption

We first establish that TDHFs are equivalent to CPA-secure public-key encryption. The correspondence is direct: a TDHF provides exactly the pseudorandom value needed to mask a plaintext, while the trapdoor enables recovery of that value. We prove both directions for messages in $\{0,1\}^{\ell(\lambda)}$, where $\ell(\lambda)$ is the output length of the hardcore-evaluation algorithm.

Theorem 1. *Perfectly correct, secure trapdoor hardcore functions exist if and only if perfectly correct,* IND-CPA*-secure public-key encryption schemes exist.*

We defer the proof of theorem 1 to appendix A.

Corollary 1 ([5]). *There is no black-box construction of poly-to-one* TDF *from* TDHF*.*

Proof (Proof sketch). Suppose that there were a black-box construction of poly-to-one trapdoor functions from TDHFs. Composing this construction with the public-key-encryption-to-TDHF direction of Theorem 1 would give a black-box construction of poly-to-one trapdoor functions from public-key encryption. This contradicts the separation of Gertner, Malkin, and Reingold [5], which rules out such a construction from trapdoor predicates, equivalently public-key encryption.

5 Strengthenings of Trapdoor Hardcore Functions

There are two natural and conceptually distinct ways to strengthen the primitive.

The first strengthens its *security*. In the basic security experiment, the adversary receives a single public evaluation and must distinguish its hardcore value from uniform. We strengthen this experiment by allowing the adversary to query the partial-inversion algorithm on other evaluations, both before and after receiving the challenge, except that the challenge evaluation itself cannot be queried after challenge generation.

The second strengthens its *recovery capability*. Instead of requiring the trapdoor to recover only the designated value $h(x)$, one may require it to recover the whole preimage x, and hence any $h(x)$. This gives the familiar notion of a trapdoor function and, when evaluation is injective, an injective trapdoor function. From corollary 1, we note that this is a strict strengthening.

These two strengthenings are conceptually orthogonal. Adaptivity increases the information available to the adversary, whereas full inversion increases the information recovered by the trapdoor holder. In particular, adaptive security does not require the trapdoor to recover the complete input. The TDHF framework separates these dimensions and makes their interaction explicit.

5.1 Adaptive Trapdoor Hardcore Functions

We first strengthen TDHF in the adaptive direction. The trapdoor continues to recover only $h(x)$; what changes is the adversary's access to the trapdoor functionality. In addition to the public key and challenge evaluation, the adversary may query the partial-inversion algorithm on other evaluations: $y \longmapsto \mathsf{pInv}(\mathsf{tdhf.sk}, y)$.

We say that a TDHF is adaptively secure if this guarantee holds even when the adversary makes partial-inversion queries both before and after receiving the challenge, except that the challenge evaluation itself is excluded from post-challenge queries.

5.1.1 Adaptive Security Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a two-stage PPT adversary. Define the partial-inversion oracle

$$\mathcal{O}_{\mathsf{TDHF.sk}}(y) := \mathsf{pInv}(\mathsf{TDHF.sk}, y).$$

For a challenge evaluation $y^\star$, define the restricted partial-inversion oracle

$$\mathcal{O}^{y^\star}_{\mathsf{tdhf.sk}}(y) := \begin{cases} \bot, & \text{if } y = y^\star, \\ \mathsf{pInv}(\mathsf{tdhf.sk}, y), & \text{otherwise.} \end{cases}$$

For $b \in \{0,1\}$, define the experiment $\mathsf{Expt}^{\mathsf{atdhf}\text{-}b}_{\mathsf{TDHF},\mathcal{A}}(\lambda)$ as follows.

function $\mathsf{Expt}^{\mathsf{atdhf}\text{-}b}_{\mathsf{TDHF},\mathcal{A}}(1^\lambda)$
  $(\mathsf{tdhf.pk}, \mathsf{tdhf.sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
  $\sigma \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{tdhf.sk}}}(1^\lambda, \mathsf{tdhf.pk})$
  $x^\star \xleftarrow{\$} \mathcal{D}_\lambda$
  $y^\star \leftarrow \mathsf{Eval}(\mathsf{tdhf.pk}, x^\star)$
  if $b = 0$ **then**
    $z^\star \leftarrow \mathsf{hEval}(\mathsf{tdhf.pk}, x^\star)$
  else
    $z^\star \xleftarrow{\$} \{0,1\}^{\ell(\lambda)}$
  end if

$b' \leftarrow \mathcal{A}_2^{\mathcal{O}^{y^\star}_{\mathsf{tdhf.sk}}}\left(1^\lambda, \mathsf{tdhf.pk}, \sigma, y^\star, z^\star\right)$
return b'
end function

Definition 11 (ATDHF security). *A perfectly correct* TDHF *is an* ATDHF *if, for every two-stage* PPT *adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, the advantage*

$$\mathsf{Adv}^{\mathsf{atdhf}}_{\mathsf{TDHF},\mathcal{A}}(\lambda) := \left| \Pr\left[\mathsf{Expt}^{\mathsf{atdhf\text{-}0}}_{\mathsf{TDHF},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{atdhf\text{-}1}}_{\mathsf{TDHF},\mathcal{A}}(\lambda) = 1\right] \right|$$

is negligible in λ*.*

Theorem 2 (Characterization of chosen-ciphertext security). *Perfectly correct and secure* ATDHF*s are black-box equivalent to perfectly correct,* IND-CCA2*-secure public-key encryption schemes:*

ATDHF $\iff_{\mathsf{BB}}$ IND-CCA2-secure public-key encryption.

We defer the proof of Theorem 2 to Appendix B.

Corollary 2 (Black-box separation from full inversion). *There is no black-box construction of* iTDF *from* ATDHF*.*

Proof. Suppose, toward a contradiction, that there is a black-box construction of injective trapdoor functions from ATDHFs. By Theorem 2, every IND-CCA2-secure public-key encryption scheme yields an ATDHF through a black-box construction. Composing these constructions would therefore give a black-box construction of injective trapdoor functions from IND-CCA2-secure public-key encryption.

Gertner, Malkin, and Reingold [5] construct an oracle relative to which trapdoor predicates exist but poly-to-one trapdoor functions do not admit such a black-box construction. Myers and shelat [19] observe that the trapdoor predicate in this oracle world already constitutes CCA2-secure one-bit encryption. They also give a black-box transformation from one-bit CCA2-secure encryption to CCA2-secure encryption for polynomial-length messages. Hence, in the same oracle world, IND-CCA2-secure public-key encryption exists but poly-to-one trapdoor functions cannot be constructed through the stated class of black-box constructions.

Since every injective trapdoor function is, in particular, a poly-to-one trapdoor function, the composed construction above would contradict this separation.

The same interpretation appears explicitly in subsequent work. Hemenway and Ostrovsky describe injective trapdoor functions as black-box separated from IND-CCA encryption, both in their work on constructing injective trapdoor functions from oblivious transfer [8] and in their work on constructing lossy trapdoor functions from lossy encryption [9].

Thus, the CCA2 characterization is not merely a reformulation of full inversion. An ATDHF captures IND-CCA2-secure encryption while remaining strictly below injective trapdoor functions in the stated black-box construction order.

Theorem 2 and Corollary 2 separate adaptive security from full inversion. Ordinary TDHF security protects the designated value without partial-inversion access. An ATDHF preserves this guarantee in the presence of admissible pre- and post-challenge queries. In particular,

$$\mathsf{ATDHF} \implies \mathsf{TDHF},$$

matching the implication

$$\mathsf{IND\text{-}CCA2} \implies \mathsf{IND\text{-}CPA}.$$

The converse direction is the security-upgrade question addressed by generic CPA-to-CCA2 transformations.

Corollary 3 ([4]). *There is no shielding*[2] *black-box construction of* ATDHF *from* TDHF*s in the model of [4].*

Proof Sketch.
Following the separation of semantic and chosen-ciphertext security by Gertner, Malkin, and Myers [4], shielding black-box reductions from CCA-secure primitives to CPA-secure primitives are impossible. For single-bit hardcore outputs, our one-bit ATDHF-to-CCA2 construction (Theorem 7) yields a shielding black-box reduction where decryption only invokes the partial-inversion oracle (pInv) without black-box access to forward evaluation. Consequently, any shielding black-box construction of ATDHF from TDHF would violate the GMM separation [4]. (Multi-bit extensions, by contrast, require non-shielding techniques [19], which bypass this barrier).

Corollary 4 ([3,17,13,10,20]). *There is a black-box construction of* ATDHF *from* TDHF *and* X*, where* $X \in \{\mathsf{RO}, \mathsf{HPRG}, \mathsf{KDM}\text{-}\textit{security}, \mathsf{iTDF}, \mathsf{LTDF}\}$

Full inversion is also relevant to the study of adaptive security: known constructions use complete preimage recovery to enforce ciphertext consistency and thereby obtain chosen-ciphertext security. Later, we will see that lossiness provides a different route to full inversion under an additional parameter condition.

[2] We use the term shielding in the sense of Gertner, Malkin, and Myers [4]: a black-box construction (or reduction) is shielding if it does not use the underlying primitive's forward evaluation subroutine (Eval) inside the security reduction's (partial) inversion/decryption simulation. As shown by Myers and Shelat [19], general multi-bit transformations are inherently non-shielding, but shielding black-box separations (such as [4]) apply directly to the foundational one-bit regime.

5.2 Full Inversion

A different strengthening increases the recovery power of the trapdoor. Given a public evaluation $f(x)$, a TDHF requires recovery only of the prescribed value $h(x)$. A trapdoor function instead recovers a complete preimage x' satisfying $f(x') = f(x)$. If f is injective, then this preimage is unique and hence $x' = x$.

The three recovery interfaces can therefore be written as

$$\begin{aligned} \mathsf{TDHF} &: f(x) \xrightarrow{\text{trapdoor}} h(x), \\ \mathsf{TDF} &: f(x) \xrightarrow{\text{trapdoor}} x' \text{ such that } f(x') = f(x), \\ \mathsf{iTDF} &: f(x) \xrightarrow{\text{trapdoor}} x. \end{aligned}$$

A trapdoor function is not obtained simply by setting $h(x) = x$ in the TDHF definition. Indeed, given $f(x)$ and a candidate value x', an adversary can efficiently test whether $f(x') = f(x)$. The real input always satisfies this relation, whereas an independent uniform input generally does not. Thus, x cannot be pseudorandom given $f(x)$. Full inversion therefore strengthens the trapdoor functionality, rather than arising as a special choice of the hardcore function.

Corollary 2 shows that this difference in recovery capability is substantive: adaptive recovery of the designated value does not suffice to construct injective full inversion.

The next subsection explains how this additional functionality can be used to obtain adaptive security.

5.3 Full Inversion as a Route to Adaptivity

Although adaptivity and full inversion strengthen different aspects of a TDHF, known results show that full inversion provides one route to adaptive security.

Hohenberger, Koppula, and Waters [10] construct CCA2-secure public-key encryption from injective trapdoor functions through a black-box transformation. Their result does not require the underlying trapdoor function to satisfy additional properties such as lossiness or correlated-product security. Thus, at the level of public-key encryption, full inversion suffices to obtain adaptive chosen-ciphertext security.

Mao and Yi [18] establish a corresponding transformation directly at the level of trapdoor primitives. Under their stated restrictions, they construct adaptive trapdoor functions and tag-based adaptive trapdoor functions from injective trapdoor functions. Schematically,

$$\mathsf{iTDF} \Longrightarrow \mathsf{ATDF}.$$

These results identify full inversion as one route to adaptive security.

Combining these positive constructions with Corollary 2 gives a strict hierarchy in the black-box construction order:

$$\boxed{\mathsf{iTDF} \Longrightarrow_{\mathsf{BB}} \mathsf{ATDHF}, \qquad \mathsf{ATDHF} \not\Longrightarrow_{\mathsf{BB}} \mathsf{iTDF}.}$$

The forward implication shows that injective full inversion provides sufficient structure for adaptive TDHF security. The reverse separation shows that this additional recovery capability is not inherent to the primitive that characterizes IND-CCA2-secure encryption.

The strict hierarchy formalizes the distinction between adaptivity and full inversion. Adaptivity strengthens the adversary's oracle access while requiring the designated value to remain pseudorandom, whereas full inversion strengthens the information recovered by the trapdoor.

The characterization theorems identify the direct upgrade

$$\mathsf{TDHF} \overset{?}{\Longrightarrow} \mathsf{ATDHF}$$

with the problem of generically upgrading IND-CPA-secure encryption to IND-CCA2-secure encryption. This formulation separates the adaptivity question from the stronger requirement of complete preimage recovery.

The separation therefore sharpens the constructive question. Full inversion gives one route to adaptive security, but no black-box construction can recover injective full inversion from ATDHFs in the stated separation model. It remains to determine whether ATDHFs can be constructed directly from standard assumptions without first obtaining injective trapdoor functions, full randomness recovery, or IND-CCA2-secure encryption.

We next turn to the hiding dimension. Lossy TDHFs strengthen computational pseudorandomness to statistical hiding under an indistinguishable key mode.

6 Lossy Trapdoor Hardcore Functions

We now consider a third strengthening of a TDHF. Adaptive security strengthens the adversarial power, while full inversion strengthens the trapdoor's recovery capability. Lossiness instead strengthens the hiding guarantee under an alternate public-key mode.

A lossy trapdoor hardcore function has two computationally indistinguishable key modes. In the recoverable mode, the trapdoor recovers the designated hardcore value. In the lossy mode, that value is statistically indistinguishable from uniform, even given the public key and the public evaluation.

This notion differs from a lossy trapdoor function. A lossy trapdoor function loses information about the complete input. A lossy TDHF requires statistical hiding only of the designated hardcore value; the public evaluation may retain information about the rest of the input.

We first prove that perfectly correct and secure LTDHFs are black-box equivalent to perfectly correct and secure lossy public-key encryption. We then examine whether this equivalence can yield full inversion. The canonical construction does not preserve the message-to-randomness ratio required by the known lossy-encryption-to-LTDF transformation. Fiber samplability provides efficient access to prescribed designated-value fibers, fiber hiding hides information about the

selected fiber. Together, they form the additional condition needed for parameter preservation.

6.1 Syntax and Correctness

Let $\lambda \in \mathbb{N}$ be the security parameter. For every λ, let $\mathcal{D}_\lambda$ and $\mathcal{R}_\lambda$ denote the input space and the evaluation range, respectively. Let $\mathcal{R}'_\lambda = \{0,1\}^{\ell(\lambda)}$ denote the hardcore range, where $\ell\colon \mathbb{N} \to \mathbb{N}$ is polynomially bounded. As in the definition of a TDHF, we require that these spaces be efficiently recognizable and that the uniform distribution over $\mathcal{D}_\lambda$ be efficiently sampleable.

Definition 12 (Lossy trapdoor hardcore function). *A lossy trapdoor hardcore function is a tuple of* PPT *algorithms* $\mathsf{LTDHF} = (\mathsf{KGen}_{\mathsf{rec}}, \mathsf{KGen}_{\mathsf{los}}, \mathsf{Eval}, \mathsf{hEval}, \mathsf{pInv})$ *with the following syntax.*

- $\mathsf{KGen}_{\mathsf{rec}}(1^\lambda) \to (\mathsf{tdhf.pk}_{\mathsf{rec}}, \mathsf{tdhf.sk})$*: The randomized recoverable-mode key-generation algorithm outputs a public evaluation key* $\mathsf{tdhf.pk}_{\mathsf{rec}}$ *and a secret trapdoor* $\mathsf{tdhf.sk}$*.*
- $\mathsf{KGen}_{\mathsf{los}}(1^\lambda) \to \mathsf{tdhf.pk}_{\mathsf{los}}$*: The randomized lossy-mode key-generation algorithm outputs a public evaluation key* $\mathsf{tdhf.pk}_{\mathsf{los}}$*. No trapdoor is associated with this mode.*
- $\mathsf{Eval}(\mathsf{tdhf.pk}, x) \to y$*: The deterministic evaluation algorithm takes as input a public key generated in either mode and an input* $x \in \mathcal{D}_\lambda$*, and outputs an evaluation* $y \in \mathcal{R}_\lambda$*.*
- $\mathsf{hEval}(\mathsf{tdhf.pk}, x) \to z$*: The deterministic hardcore-evaluation algorithm takes as input a public key generated in either mode and an input* $x \in \mathcal{D}_\lambda$*, and outputs a hardcore value* $z \in \mathcal{R}'_\lambda$*.*
- $\mathsf{pInv}(\mathsf{tdhf.sk}, y) \to z'$ *or* $\perp$*: The deterministic partial-inversion algorithm takes as input a recoverable-mode trapdoor and an evaluation* $y \in \mathcal{R}_\lambda$*, and outputs a value* $z' \in \mathcal{R}'_\lambda$ *or a distinguished failure symbol* $\perp$*.*

Definition 13 (Perfect correctness). *An* LTDHF *is perfectly correct if, for every security parameter* $\lambda \in \mathbb{N}$*, every key pair* $(\mathsf{tdhf.pk}_{\mathsf{rec}}, \mathsf{tdhf.sk}) \in \mathrm{Supp}\left(\mathsf{KGen}_{\mathsf{rec}}(1^\lambda)\right)$*, and every input* $x \in \mathcal{D}_\lambda$*, it holds that*

$$\mathsf{pInv}\left(\mathsf{tdhf.sk}, \mathsf{Eval}\left(\mathsf{tdhf.pk}_{\mathsf{rec}}, x\right)\right) = \mathsf{hEval}\left(\mathsf{tdhf.pk}_{\mathsf{rec}}, x\right).$$

6.2 LTDHF Security

The security of an LTDHF consists of two requirements. First, a public key generated in the recoverable mode must be computationally indistinguishable from a public key generated in the lossy mode. Second, for every lossy public key, the designated value of a uniformly sampled input must be statistically close to uniform given its public evaluation.

Definition 14 (Mode indistinguishability). *An* LTDHF *has indistinguishable modes if: for every* PPT *adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\mathsf{mode}}_{\mathsf{LTDHF},\mathcal{A}}(\lambda) := \left| \begin{array}{l} \Pr\left[\mathcal{A}\left(1^\lambda, \mathsf{tdhf.pk_{rec}}\right) = 1 \,\middle|\, (\mathsf{tdhf.pk_{rec}}, \mathsf{tdhf.sk}) \leftarrow \mathsf{KGen_{rec}}(1^\lambda)\right] \\ - \Pr\left[\mathcal{A}\left(1^\lambda, \mathsf{tdhf.pk_{los}}\right) = 1 \,\middle|\, \mathsf{tdhf.pk_{los}} \leftarrow \mathsf{KGen_{los}}(1^\lambda)\right] \end{array} \right|$$

is negligible in λ*.*

Definition 15 (Statistical lossiness). *An* LTDHF *satisfies statistical lossiness if there exists a negligible function* ε *such that, for every security parameter* λ *and every lossy public key* $\mathsf{TDHF.pk_{los}} \in \mathrm{Supp}\left(\mathsf{KGen_{los}}(1^\lambda)\right)$*, we have*

$$\Delta\left(\left(\mathsf{Eval}(\mathsf{TDHF.pk_{los}}, X), \mathsf{hEval}(\mathsf{TDHF.pk_{los}}, X)\right), \left(\mathsf{Eval}(\mathsf{TDHF.pk_{los}}, X), U\right)\right) \leq \varepsilon(\lambda),$$

where $X \xleftarrow{\$} \mathcal{D}_\lambda$ *and* $U \xleftarrow{\$} \{0,1\}^{\ell(\lambda)}$ *are independent.*

Definition 16 (Secure lossy trapdoor hardcore function). *A perfectly correct* LTDHF *is secure if it satisfies indistinguishability of modes according to Definition 14 and statistical lossiness according to Definition 15.*

The recoverable mode alone already defines an ordinary TDHF. The next proposition shows that its computational security follows from mode indistinguishability and statistical lossiness.

Proposition 1. *Every perfectly correct and secure* LTDHF *yields a perfectly correct and secure* TDHF *by defining* $\mathsf{KGen} := \mathsf{KGen_{rec}}$ *and retaining the same algorithms* Eval*,* hEval*, and* pInv*.*

Proof. Perfect correctness follows immediately from recoverable-mode correctness.

It remains to prove security. Let $X \xleftarrow{\$} \mathcal{D}_\lambda$ and $U \xleftarrow{\$} \{0,1\}^{\ell(\lambda)}$ be independent. Consider the following distributions:

$$\begin{aligned} H_0 &:= \left(\mathsf{TDHF.pk_{rec}}, \mathsf{Eval}\left(\mathsf{TDHF.pk_{rec}}, X\right), \mathsf{hEval}\left(\mathsf{TDHF.pk_{rec}}, X\right)\right), \\ H_1 &:= \left(\mathsf{TDHF.pk_{los}}, \mathsf{Eval}\left(\mathsf{TDHF.pk_{los}}, X\right), \mathsf{hEval}\left(\mathsf{TDHF.pk_{los}}, X\right)\right), \\ H_2 &:= \left(\mathsf{TDHF.pk_{los}}, \mathsf{Eval}\left(\mathsf{TDHF.pk_{los}}, X\right), U\right), \\ H_3 &:= \left(\mathsf{TDHF.pk_{rec}}, \mathsf{Eval}\left(\mathsf{TDHF.pk_{rec}}, X\right), U\right), \end{aligned}$$

where $(\mathsf{TDHF.pk_{rec}}, \mathsf{TDHF.sk}) \leftarrow \mathsf{KGen_{rec}}(1^\lambda)$ in H_0 and H_3, and $\mathsf{TDHF.pk_{los}} \leftarrow \mathsf{KGen_{los}}(1^\lambda)$ in H_1 and H_2. Mode indistinguishability and closure under efficient post-processing give

$$H_0 \approx_c H_1 \qquad \text{and} \qquad H_2 \approx_c H_3.$$

It remains to compare H_1 and H_2. For every lossy public key $\mathsf{pk} \in \mathrm{Supp}\left(\mathsf{KGen_{los}}(1^\lambda)\right)$, Definition 15 gives

$$\Delta\Big(\left(\mathsf{Eval}(\mathsf{pk}, X), \mathsf{hEval}(\mathsf{pk}, X)\right), \left(\mathsf{Eval}(\mathsf{pk}, X), U\right)\Big) \leq \varepsilon(\lambda).$$

Therefore,

$$\begin{aligned}\Delta(H_1, H_2) &= \mathbb{E}_{\mathsf{pk}\leftarrow\mathsf{KGen}_{\mathsf{los}}(1^\lambda)}\left[\Delta\left((\mathsf{Eval}(\mathsf{pk}, X), \mathsf{hEval}(\mathsf{pk}, X)\right), (\mathsf{Eval}(\mathsf{pk}, X), U))\right] \\ &\leq \varepsilon(\lambda).\end{aligned}$$

Hence, $H_1 \approx_s H_2$. Combining the three transitions gives $H_0 \approx_c H_3$. The distributions H_0 and H_3 are exactly the real and random distributions in the TDHF security experiment. Therefore, the recoverable mode defines a secure TDHF.

Theorem 3 (Characterization of lossy encryption). *Perfectly correct and secure lossy trapdoor hardcore functions are black-box equivalent to perfectly correct and secure lossy public-key encryption schemes.*

Prior work shows that lossiness (under certain constraints) supplies the additional structure needed for full inversion. We discuss this notion in the next section.

6.3 Relationship between Lossiness and Full Inversion

Hemenway and Ostrovsky [9] show that lossy encryption can be derandomized into a lossy trapdoor function when the message length is at least one bit larger than the randomness length – pairwise-independent hashing can be used to derive the encryption randomness from the prospective input. The resulting deterministic map is injective in its recoverable mode and lossy under indistinguishable lossy parameters.

The canonical construction in Theorem 3, however, does not preserve this regime. If an LTDHF has input length $d(\lambda)$ and hardcore-output length $\ell(\lambda)$, then the resulting encryption scheme has message length $\ell(\lambda)$ but uses a complete LTDHF input as randomness, and therefore has randomness length $d(\lambda)$.

Moreover, statistical lossiness requires $\mathsf{hEval}\left(\mathsf{pk}_{\mathsf{los}}, X\right) \approx_s U_{\ell(\lambda)}$ for $X \xleftarrow{\$} \{0,1\}^{d(\lambda)}$. Since the left-hand side is a deterministic function of a $d(\lambda)$-bit source, it has support size at most $2^{d(\lambda)}$. Consequently,

$$\Delta\left(\mathsf{hEval}(\mathsf{pk}_{\mathsf{los}}, X), U_{\ell(\lambda)}\right) \geq 1 - 2^{d(\lambda)-\ell(\lambda)}.$$

Thus, the long-message condition $\ell(\lambda) \geq d(\lambda)+1$ is incompatible with LTDHF statistical lossiness.

6.3.1 Fiber-Samplable LTDHFs The canonical construction treats both the hardcore value and the remaining information in the input as random coins. A more randomness-efficient construction would instead use the plaintext to determine the hardcore value and spend fresh randomness only to sample an input consistent with that value.

This motivates the following strengthening.

Definition 17 (Fiber-samplable LTDHF). *Let* $\mathsf{LTDHF} = (\mathsf{KGen}_{\mathsf{rec}}, \mathsf{KGen}_{\mathsf{los}}, \mathsf{Eval}, \mathsf{hEval}, \mathsf{pInv})$ *be an* LTDHF *with input space* $\mathcal{D}_\lambda$ *and hardcore range* $\{0,1\}^{\ell(\lambda)}$.

We say that LTDHF *is* fiber samplable *if there exists a* PPT *algorithm*

$$\mathsf{FSamp}\left(\mathsf{pk}, z; r\right) \to x,$$

where $z \in \{0,1\}^{\ell(\lambda)}$ *and* $r \in \{0,1\}^{s(\lambda)}$, *satisfying the following properties in both the recoverable and lossy key modes.*

Fiber consistency. *For every valid public key* pk, *every* $z \in \{0,1\}^{\ell(\lambda)}$, *and every* $r \in \{0,1\}^{s(\lambda)}$, *if*

$$x := \mathsf{FSamp}(\mathsf{pk}, z; r), \qquad \textit{then} \qquad \mathsf{hEval}(\mathsf{pk}, x) = z.$$

Distributional fidelity. *For each* $\mathsf{mode} \in \{\mathsf{rec}, \mathsf{los}\}$, *let* $\mathsf{tdhf.pk}_{\mathsf{mode}}$ *be generated in that mode, and sample independently* $Z \xleftarrow{\$} \{0,1\}^{\ell(\lambda)}$ *and* $R \xleftarrow{\$} \{0,1\}^{s(\lambda)}$. *Then*

$$\left(\mathsf{tdhf.pk}_{\mathsf{mode}}, \mathsf{FSamp}\left(\mathsf{tdhf.pk}_{\mathsf{mode}}, Z; R\right)\right) \approx_s \left(\mathsf{tdhf.pk}_{\mathsf{mode}}, X\right),$$

where $X \xleftarrow{\$} \mathcal{D}_\lambda$.

The parameter $s(\lambda)$ is the number of random bits used by the fiber sampler after the designated value has been fixed. We call it the fiber-sampling randomness length. For perfectly regular fibers of size $2^{d(\lambda)-\ell(\lambda)}$,

$$s(\lambda) = d(\lambda) - \ell(\lambda).$$

Fiber samplability gives a distributional decomposition of an LTDHF input. A uniform designated value Z selects a fiber, and independent randomness R selects a correctly distributed point within that fiber. The trapdoor recovers Z, while R represents the residual sampling randomness.

Definition 18 (Fiber hiding). *Let* LTDHF *be a fiber-samplable* LTDHF *with designated-value length* $\ell(\lambda)$ *and fiber-sampling randomness length* $s(\lambda)$.

For a lossy public key $\mathsf{pk}_{\mathsf{los}}$ *and* $z_0, z_1 \in \{0,1\}^{\ell(\lambda)}$, *define the distribution*

$$Y_{\mathsf{pk}_{\mathsf{los}}, z_b} := \mathsf{Eval}\left(\mathsf{pk}_{\mathsf{los}}, \mathsf{FSamp}\left(\mathsf{pk}_{\mathsf{los}}, z_b; R\right)\right),$$

where $R \xleftarrow{\$} \{0,1\}^{s(\lambda)}$. *The* LTDHF *satisfies* fiber hiding *if there exists a negligible function* ε *such that, for every security parameter* λ, *every lossy public key* $\mathsf{pk}_{\mathsf{los}} \in \mathrm{Supp}\left(\mathsf{KGen}_{\mathsf{los}}(1^\lambda)\right)$, *and every* $z_0, z_1 \in \{0,1\}^{\ell(\lambda)}$, *we have*

$$\Delta\left(Y_{\mathsf{pk}_{\mathsf{los}}, z_0}, Y_{\mathsf{pk}_{\mathsf{los}}, z_1}\right) \le \varepsilon(\lambda).$$

Definition 19 (Fiber-hiding LTDHF). *A* fiber-hiding LTDHF *is a secure* LTDHF *that is fiber samplable and satisfies fiber hiding.*

Fiber samplability and fiber hiding serve different purposes. Fiber samplability provides efficient access to a prescribed designated-value fiber and reproduces the honest input distribution for a uniform fiber label. Fiber hiding requires the lossy evaluation to hide which prescribed fiber was selected.

We next use this decomposition to construct lossy encryption whose message length is $\ell(\lambda)$ and whose randomness length is only $s(\lambda)$.

6.3.2 Parameter-Preserving Lossy Encryption A fiber-hiding LTDHF allows the plaintext to select a designated-value fiber while ensuring that the lossy evaluation hides that choice. Encryption randomness is used only to sample a point within the selected fiber.

Theorem 4 (Parameter-preserving lossy encryption). *Let* LTDHF *be a perfectly correct and secure fiber-hiding* LTDHF *with designated-value length* $\ell(\lambda)$ *and fiber-sampling randomness length* $s(\lambda)$*. Then there exists a perfectly correct and secure lossy public-key encryption scheme with message space* $\{0,1\}^{\ell(\lambda)}$ *and randomness space* $\{0,1\}^{s(\lambda)}$.

Proof. We construct a lossy public-key encryption scheme $\mathsf{LPKE} = \big(\mathsf{LPKE.KGen_{rec}}, \mathsf{LPKE.KGen_{los}}, \mathsf{LPKE.Enc}, \mathsf{LPKE.Dec}\big)$.

The message itself selects the designated-value fiber, and the encryption randomness is used only to sample a point within that fiber.

function $\mathsf{LPKE.KGen_{rec}}(1^\lambda)$
  $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen_{rec}}(1^\lambda)$
  return $(\mathsf{pk}, \mathsf{sk})$
end function

function $\mathsf{LPKE.KGen_{los}}(1^\lambda)$
  $\mathsf{pk} \leftarrow \mathsf{KGen_{los}}(1^\lambda)$
  return pk
end function

function $\mathsf{LPKE.Enc}(\mathsf{pk}, m; r)$
  $x \leftarrow \mathsf{FSamp}(\mathsf{pk}, m; r)$
  return $\mathsf{Eval}(\mathsf{pk}, x)$
end function

function $\mathsf{LPKE.Dec}(\mathsf{sk}, c)$
  return $\mathsf{pInv}(\mathsf{sk}, c)$
end function

Perfect correctness. Fix a recoverable-mode key pair, a message $m \in \{0,1\}^{\ell(\lambda)}$, and randomness $r \in \{0,1\}^{s(\lambda)}$. Let $x := \mathsf{FSamp}(\mathsf{pk}, m; r)$. Fiber consistency gives $\mathsf{hEval}(\mathsf{pk}, x) = m$. By LTDHF correctness,

$$\begin{aligned}\mathsf{LPKE.Dec}\left(\mathsf{sk}, \mathsf{LPKE.Enc}(\mathsf{pk}, m; r)\right) &= \mathsf{pInv}\left(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, x)\right) \\ &= \mathsf{hEval}(\mathsf{pk}, x) \\ &= m.\end{aligned}$$

Mode indistinguishability. Mode indistinguishability follows directly from LTDHF mode indistinguishability.

Ciphertext lossiness. Fix a lossy public key $\mathsf{pk}_{\mathsf{los}} \in \mathrm{Supp}\left(\mathsf{KGen}_{\mathsf{los}}(1^\lambda)\right)$ and messages $m_0, m_1 \in \{0,1\}^{\ell(\lambda)}$. For $R_0, R_1 \xleftarrow{\$} \{0,1\}^{s(\lambda)}$, we have

$$\mathsf{LPKE.Enc}\left(\mathsf{pk}_{\mathsf{los}}, m_b; R_b\right) \equiv Y_{\mathsf{pk}_{\mathsf{los}}, m_b}$$

for $b \in \{0,1\}$. Hence, fiber hiding gives

$$\Delta\left(\mathsf{LPKE.Enc}\left(\mathsf{pk}_{\mathsf{los}}, m_0; R_0\right), \mathsf{LPKE.Enc}\left(\mathsf{pk}_{\mathsf{los}}, m_1; R_1\right)\right) \leq \varepsilon(\lambda).$$

Thus, the constructed scheme satisfies ciphertext lossiness.

Corollary 5 (Parameter-preserving characterization). *Fiber-samplable, fiber-hiding* LTDHF*s with designated-value length $\ell(\lambda)$ and fiber-sampling randomness length $s(\lambda)$ are black-box equivalent to lossy public-key encryption schemes with message length $\ell(\lambda)$ and encryption-randomness length $s(\lambda)$.*

Proof. The forward direction follows from Theorem 4. For the reverse direction, let $\Pi = (\mathsf{Gen}_{\mathsf{rec}}, \mathsf{Gen}_{\mathsf{lossy}}, \mathsf{Enc}, \mathsf{Dec})$ be a perfectly correct and secure lossy public-key encryption scheme with message space $\mathcal{M}_\lambda = \{0,1\}^{\ell(\lambda)}$ and randomness space $\mathcal{R}_\lambda = \{0,1\}^{s(\lambda)}$. Apply the lossy-encryption-to-LTDHF construction from Theorem 11. Its input space is $\mathcal{D}_\lambda := \mathcal{M}_\lambda \times \mathcal{R}_\lambda$, and its public algorithms are

$$\mathsf{Eval}\left(\mathsf{pk}, (m, r)\right) := \mathsf{Enc}(\mathsf{pk}, m; r)$$

and

$$\mathsf{hEval}\left(\mathsf{pk}, (m, r)\right) := m.$$

Define $\mathsf{FSamp}(\mathsf{pk}, z; r) := (z, r)$. Fiber consistency holds perfectly: $\mathsf{hEval}\left(\mathsf{pk}, \mathsf{FSamp}(\mathsf{pk}, z; r)\right) = z$. For independent $Z \xleftarrow{\$} \{0,1\}^{\ell(\lambda)}$ and $R \xleftarrow{\$} \{0,1\}^{s(\lambda)}$, we have $\mathsf{FSamp}(\mathsf{pk}, Z; R) = (Z, R)$, which is uniform over $\mathcal{D}_\lambda$.

Thus, distributional fidelity holds with statistical distance zero in both key modes. It remains to verify fiber hiding. Fix a lossy public key $\mathsf{pk}_{\mathsf{los}} \in \mathrm{Supp}\left(\mathsf{Gen}_{\mathsf{lossy}}(1^\lambda)\right)$ and any $z_0, z_1 \in \{0,1\}^{\ell(\lambda)}$. For independent $R_0, R_1 \xleftarrow{\$} \{0,1\}^{s(\lambda)}$, we have, for each $b \in \{0,1\}$,

$$\mathsf{Eval}\left(\mathsf{pk}_{\mathsf{los}}, \mathsf{FSamp}\left(\mathsf{pk}_{\mathsf{los}}, z_b; R_b\right)\right) = \mathsf{Enc}\left(\mathsf{pk}_{\mathsf{los}}, z_b; R_b\right).$$

Therefore, ciphertext lossiness gives

$$\Delta\Big(\mathsf{Eval}\left(\mathsf{pk}_{\mathsf{los}}, \mathsf{FSamp}\left(\mathsf{pk}_{\mathsf{los}}, z_0; R_0\right)\right), \mathsf{Eval}\left(\mathsf{pk}_{\mathsf{los}}, \mathsf{FSamp}\left(\mathsf{pk}_{\mathsf{los}}, z_1; R_1\right)\right)\Big) \leq \varepsilon(\lambda).$$

Hence, the constructed LTDHF satisfies fiber hiding. The designated-value length is $\ell(\lambda)$, and the fiber sampler uses exactly $s(\lambda)$ random bits. The construction therefore preserves both parameters.

6.3.3 Route to Full Inversion A fiber-hiding LTDHF separates the designated-value length $\ell(\lambda)$ from the fiber-sampling randomness length $s(\lambda)$. By Corollary 5, it yields lossy encryption with message length $\ell(\lambda)$ and randomness length $s(\lambda)$. The Hemenway–Ostrovsky transformation applies when

$$\ell(\lambda) \geq s(\lambda) + 1.$$

Corollary 6 (Full inversion from fiber-hiding LTDHFs [9]). *Let* LTDHF *be a perfectly correct and secure fiber-samplable* LTDHF *that satisfies fiber hiding, with designated-value length $\ell(\lambda)$ and fiber-sampling randomness length $s(\lambda)$. If $\ell(\lambda) \geq s(\lambda) + 1$, then* LTDHF *yields an almost-always lossy trapdoor function through a black-box construction.*

Relation to direct inversion. Neither fiber samplability nor fiber hiding directly provides inversion of the LTDHF evaluation, nor does it directly provide inversion of a given public evaluation. Given $y = \mathsf{Eval}(\mathsf{pk}, x)$, the trapdoor recovers the fiber label $z = \mathsf{hEval}(\mathsf{pk}, x)$. The sampler then generates points in $\{x' : \mathsf{hEval}(\mathsf{pk}, x') = z\}$, but it does not identify which such point maps to y. Fiber samplability exposes the residual randomness within the designated-value fiber, while fiber hiding is a lossy-mode indistinguishability property. Neither identifies a preimage of a given evaluation. The LTDF consequence above is obtained indirectly through the parameter-preserving lossy-encryption construction.

7 Open Problems

The characterizations developed in this work suggest several directions for future research.

1. **Can ATDHFs be obtained directly from TDHFs?** Our characterization identifies ATDHF as the recovery interface corresponding to IND-CCA2 security. Gertner, Malkin, and Myers [4] rule out shielding black-box reductions from CCA-secure primitives to CPA-secure primitives, but non-shielding constructions remain possible. This leaves open perhaps the most natural question raised by our framework:

$$\mathsf{TDHF} \overset{?}{\underset{\text{non-shielding}}{\Longrightarrow_{\mathsf{BB}}}} \mathsf{ATDHF}.$$

2. **Can lossiness lead directly to adaptivity?** Fiber-hiding LTDHFs with $\ell(\lambda) \geq s(\lambda) + 1$ yield LTDFs (Corollary 6) and consequently ATDHFs via Peikert–Waters [20]. However, canonical LTDHFs do not satisfy this parameter condition. It remains unclear whether the hiding dimension alone already contains the ingredients needed for adaptive security:

$$\mathsf{LTDHF} \overset{?}{\Longrightarrow_{\mathsf{BB}}} \mathsf{ATDHF}.$$

3. **Can auxiliary assumptions bridge the gap to full inversion?** One route to adaptive security proceeds through full inversion. Existing CPA-to-CCA transformations suggest studying whether additional assumptions permit a generic upgrade from TDHFs to injective trapdoor functions:

$$\mathsf{TDHF} + \mathcal{X} \overset{?}{\Longrightarrow}_{\mathsf{BB}} \mathsf{iTDF},$$

for assumptions $\mathcal{X}$ such as:

(a) HPRG,

(b) KDM-security.

Partial progress in related KEM/SKE settings is known [14,15].

4. **Are adaptivity and lossiness fundamentally independent?** The framework separates adaptivity and hiding into distinct dimensions. A natural question is whether one dimension generically implies the other:

$$\mathsf{ATDHF} \overset{?}{\Longrightarrow}_{\mathsf{BB}} \mathsf{LTDHF}.$$

A positive answer would reveal unexpected structure in adaptive recovery, while a negative answer would further support the orthogonality of the two dimensions.

5. **Can every LTDHF lead to parameter preserving (or improving) LPKE?** Fiber-hiding LTDHFs provide a parameter-preserving bridge from lossy encryption to lossy trapdoor functions. It is therefore natural to ask whether fiber samplability and fiber hiding can be obtained generically:

$$\mathsf{LTDHF} \overset{?}{\Longrightarrow}_{\mathsf{BB}} \mathsf{FH\text{-}LTDHF}.$$

6. **Characterizing CCA-secure lossy encryption.** The present work separately characterizes the adaptive and lossy dimensions. An obvious next step is to understand their interaction. Is there a primitive that simultaneously captures adaptive recovery and statistical lossiness, and thereby characterizes chosen-ciphertext-secure lossy encryption?

$$\mathsf{Adaptive\text{-}Lossy\ TDHF} \overset{?}{\Longleftrightarrow}_{\mathsf{BB}} \mathsf{CCA2\text{-}Secure\ Lossy\ Encryption}.$$

7. **The minimal trapdoor primitive for chosen-ciphertext security.** Our characterization suggests that adaptive designated-value recovery captures an essential aspect of CCA2 security, but whether this perspective ultimately leads to the minimal trapdoor abstraction remains unclear. Thus, a more fundamental question remains open:

$$\mathsf{Minimal\ Trapdoor\ Primitive} \overset{?}{\Longleftrightarrow}_{\mathsf{BB}} \mathsf{IND\text{-}CCA2\ Secure\ PKE}.$$

Can one identify the minimal trapdoor primitive whose structure explains chosen-ciphertext security as fundamentally as one-way functions explain Minicrypt?

References

1. Bellare, M., Halevi, S., Sahai, A., Vadhan, S.P.: Many-to-one trapdoor functions and their relation to public-key cryptosystems. In: Advances in Cryptology — CRYPTO '98. Lecture Notes in Computer Science, vol. 1462, pp. 283–298. Springer, Berlin, Heidelberg (1998). https://doi.org/10.1007/bfb0055735
2. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Advances in Cryptology – EUROCRYPT 2004. Lecture Notes in Computer Science, vol. 3006, pp. 207–222. Springer (2004)
3. Fujisaki, E., Okamoto, T.: How to enhance the security of public-key encryption at minimum cost. In: International workshop on public key cryptography. pp. 53–68. Springer (1999)
4. Gertner, Y., Malkin, T., Myers, S.: Towards a separation of semantic and cca security for public key encryption. In: Theory of Cryptography Conference. pp. 434–455. Springer (2007)
5. Gertner, Y., Malkin, T., Reingold, O.: On the impossibility of basing trapdoor functions on trapdoor predicates. In: Proceedings 42nd IEEE Symposium on Foundations of Computer Science. pp. 126–135. IEEE (2001)
6. Goldwasser, S., Micali, S.: Probabilistic encryption & how to play mental poker keeping secret all partial information. In: Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing. p. 365–377. STOC '82, Association for Computing Machinery, New York, NY, USA (1982). https://doi.org/10.1145/800070.802212, https://doi.org/10.1145/800070.802212
7. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences **28**(2), 270–299 (1984). https://doi.org/https://doi.org/10.1016/0022-0000(84)90070-9, https://www.sciencedirect.com/science/article/pii/0022000084900709
8. Hemenway, B., Ostrovsky, R.: Building injective trapdoor functions from oblivious transfer. Electronic Colloquium on Computational Complexity (2008)
9. Hemenway, B., Ostrovsky, R.: Building lossy trapdoor functions from lossy encryption. In: Sako, K., Sarkar, P. (eds.) Advances in Cryptology - ASIACRYPT 2013. pp. 241–260. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
10. Hohenberger, S., Koppula, V., Waters, B.: Chosen ciphertext security from injective trapdoor functions. In: Annual International Cryptology Conference. pp. 836–866. Springer (2020)
11. Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Advances in Cryptology – EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 581–598. Springer (2006)
12. Kiltz, E., Mohassel, P., O'Neill, A.: Adaptive trapdoor functions and chosen-ciphertext security. In: Gilbert, H. (ed.) Advances in Cryptology – EUROCRYPT 2010. pp. 673–692. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
13. Kitagawa, F., Matsuda, T.: Cpa-to-cca transformation for kdm security. In: Theory of Cryptography: 17th International Conference, TCC 2019, Nuremberg, Germany, December 1–5, 2019, Proceedings, Part II. p. 118–148. Springer-Verlag, Berlin, Heidelberg (2019). https://doi.org/10.1007/978-3-030-36033-7_5, https://doi.org/10.1007/978-3-030-36033-7_5
14. Kitagawa, F., Matsuda, T., Tanaka, K.: Cca-secure key encapsulation mechanisms from adaptive trapdoor functions. In: Advances in Cryptology — CRYPTO 2019. Lecture Notes in Computer Science, vol. 11692, pp. 184–214. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26954-8_7

15. Kitagawa, F., Matsuda, T., Tanaka, K.: CCA-secure key encapsulation mechanisms from adaptive trapdoor functions. Journal of Cryptology **35**(3), 23 (2022). https://doi.org/10.1007/s00145-022-09435-y
16. Koppula, V., Waters, B.: Realizing CCA security from injective trapdoor functions. In: Advances in Cryptology – CRYPTO 2019. Lecture Notes in Computer Science, vol. 11692, pp. 215–245. Springer (2019)
17. Koppula, V., Waters, B.: Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II. Lecture Notes in Computer Science, vol. 11693, pp. 671–700. Springer (2019). https://doi.org/10.1007/978-3-030-26951-7_23, https://doi.org/10.1007/978-3-030-26951-7_23
18. Mao, X., Yi, H.: Adaptive TDFs from injective TDFs. Cryptology ePrint Archive, Paper 2025/1011 (2025), https://eprint.iacr.org/2025/1011
19. Myers, S., Shelat, A.: Bit encryption is complete. In: 2009 50th Annual IEEE Symposium on Foundations of Computer Science. pp. 607–616 (2009). https://doi.org/10.1109/FOCS.2009.65
20. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Proceedings of the fortieth annual ACM symposium on Theory of computing. pp. 187–196 (2008)
21. Rabin, M.O.: Digitalized signatures and public-key functions as hard as factorization. Tech. Rep. MIT/LCS/TR-212, MIT Laboratory for Computer Science (1979)
22. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM **21**(2), 120–126 (1978)
23. Rosen, A., Segev, G.: Chosen-ciphertext security via correlated products. SIAM Journal on Computing **39**(7), 3058–3088 (2010). https://doi.org/10.1137/100782929, https://doi.org/10.1137/100782929
24. Wee, H.: Efficient chosen-ciphertext security via extractable hash proofs. In: Rabin, T. (ed.) Advances in Cryptology – CRYPTO 2010. pp. 314–332. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
25. Xue, H., Lu, X., Li, B., Liu, Y.: Lossy trapdoor relation and its applications to lossy encryption and adaptive trapdoor relation. In: Chow, S.S.M., Liu, J.K., Hui, L.C.K., Yiu, S.M. (eds.) Provable Security. pp. 162–177. Springer International Publishing, Cham (2014)

A Equivalence of CPA-Secure PKE and TDHF

The equivalence between trapdoor predicates and semantically secure public-key encryption goes back to Goldwasser and Micali [7]. In this section, we prove the corresponding equivalence between perfectly correct, secure TDHFs and perfectly correct, IND-CPA-secure public-key encryption.

We prove the equivalence directly for TDHFs whose hardcore output has length $\ell(\lambda)$. The resulting public-key encryption scheme has message space $\{0,1\}^{\ell(\lambda)}$. For $\ell(\lambda) = 1$, the equivalence recovers the classical single-bit result of Goldwasser and Micali [7].

A.1 PKE using TDHF

The PKE algorithms $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ are realized using the Trapdoor Hardcore Function algorithms $(\mathsf{KGen}, \mathsf{Eval}, \mathsf{hEval}, \mathsf{pInv})$ as follows. Let the input space of the TDHF be $\mathcal{D}_\lambda$, and let its hardcore range be $\{0,1\}^{\ell(\lambda)}$. We construct a public-key encryption scheme with message space $\{0,1\}^{\ell(\lambda)}$. Its encryption randomness is an element $r \in \mathcal{D}_\lambda$.

function $\mathsf{PKE.KGen}(1^\lambda)$
  $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{TDHF.KGen}(1^\lambda)$
  return $(\mathsf{pk}, \mathsf{sk})$
end function

function $\mathsf{PKE.Enc}(\mathsf{pk}, m, r)$
  $\mathsf{ct}_1 \leftarrow \mathsf{TDHF.Eval}(\mathsf{pk}, r)$
  $\mathsf{ct}_2 \leftarrow \mathsf{TDHF.hEval}(\mathsf{pk}, r) \oplus m$ [a]
  return $(\mathsf{ct}_1, \mathsf{ct}_2)$
end function

function $\mathsf{PKE.Dec}(\mathsf{sk}, (\mathsf{ct}_1, \mathsf{ct}_2))$
  $z \leftarrow \mathsf{TDHF.pInv}(\mathsf{sk}, \mathsf{ct}_1)$
  if $z = \perp$ **then**
    return $\perp$
  end if
  return $z \oplus \mathsf{ct}_2$
end function

[a] We make use of the KEM (Key Encapsulation Mechanism) paradigm for this construction. This is a standard technique adopted in several other works [10,20]. This construction highlights the natural alignment between TDHFs and the KEM syntax.

Lemma 1. *If the underlying* TDHF *is perfectly correct, then the public-key encryption scheme constructed above is perfectly correct.*

Proof. Fix any key pair $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{TDHF.KGen}(1^\lambda)$, any message $m \in \{0,1\}^{\ell(\lambda)}$, and any encryption randomness $r \in \mathcal{D}_\lambda$. By perfect correctness of the TDHF, $\mathsf{TDHF.pInv}(\mathsf{sk}, \mathsf{TDHF.Eval}(\mathsf{pk}, r)) = \mathsf{TDHF.hEval}(\mathsf{pk}, r)$. Therefore,

$$\begin{aligned}&\mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{PKE.Enc}(\mathsf{pk}, m; r))\\ &= \mathsf{TDHF.pInv}(\mathsf{sk}, (\mathsf{TDHF.Eval}(\mathsf{pk}, r))) \oplus (\mathsf{TDHF.hEval}(\mathsf{pk}, r) \oplus m)\\ &= m.\end{aligned}$$

Thus, the constructed public-key encryption scheme is perfectly correct.

Theorem 5. *If the underlying* TDHF *is perfectly correct and secure according to Definition 9 and 10, then the construction above is a perfectly correct, IND-CPA-secure public-key encryption scheme according to Definition 2.*

Proof. Perfect correctness follows from Lemma 1. We prove IND-CPA security.

Suppose that there exists a PPT adversary $\mathcal{A}$ that breaks the IND-CPA security of the constructed public-key encryption scheme. We construct a PPT adversary $\mathcal{B}$ that breaks the security of the underlying TDHF.

Adversary $\mathcal{B}$ receives a challenge tuple (pk, y, z) from its TDHF challenger, where $y = \mathsf{TDHF.Eval}(\mathsf{pk}, r)$ for a uniformly sampled $r \xleftarrow{\$} \mathcal{D}_\lambda$, and where z is either $\mathsf{TDHF.hEval}(\mathsf{pk}, r)$ or an independently sampled uniform string in $\{0,1\}^{\ell(\lambda)}$. Adversary $\mathcal{B}$ proceeds as follows.

```
1: function B
2:     receive (pk, y, z) from the TDHF challenger
3:     send pk to A
4:     receive m_0, m_1 ∈ {0,1}^ℓ(λ) from A
5:     b ←$ {0,1}
6:     ct* ← (y, z ⊕ m_b)
7:     send ct* to A
8:     receive b' from A
9:     if b' = b then
10:        return 1
11:    else
12:        return 0
13:    end if
14: end function
```

We analyze the two possible distributions of z. Suppose first that $z = \mathsf{TDHF.hEval}(\mathsf{pk}, r)$. Then

$$\begin{aligned}\mathsf{ct}^\star &= (y, z \oplus m_b)\\ &= (\mathsf{TDHF.Eval}(\mathsf{pk}, r), \mathsf{TDHF.hEval}(\mathsf{pk}, r) \oplus m_b)\,,\end{aligned}$$

which is distributed identically to an honestly generated encryption of m_b. Therefore,

$$\Pr[\mathcal{B} \text{ outputs } 1 \mid z \text{ is real}] = \Pr[b' = b \text{ in the IND-CPA experiment}]\,.$$

Suppose instead that $z \xleftarrow{\$} \{0,1\}^{\ell(\lambda)}$ is sampled uniformly and independently of all other variables. For every fixed value of m_b, the value $z \oplus m_b$ is uniformly distributed over $\{0,1\}^{\ell(\lambda)}$ and is independent of the challenge bit b. Consequently,

$$\Pr\left[\mathcal{B} \text{ outputs } 1 \mid z \text{ is uniform}\right] = \frac{1}{2}.$$

It follows that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{tdhf}}_{\mathsf{TDHF},\mathcal{B}}(\lambda) &= \left|\Pr\left[\mathcal{B} \text{ outputs } 1 \mid z \text{ is real}\right] - \Pr\left[\mathcal{B} \text{ outputs } 1 \mid z \text{ is uniform}\right]\right| \\ &= \left|\Pr\left[b' = b \text{ in the IND-CPA experiment}\right] - \frac{1}{2}\right|.\end{aligned}$$

Thus, any non-negligible IND-CPA advantage for $\mathcal{A}$ yields a non-negligible advantage for $\mathcal{B}$ against the security of the underlying TDHF, contradicting the assumed security of the TDHF. Therefore, the constructed public-key encryption scheme is IND-CPA secure.

A.2 TDHF using PKE

Let PKE be a public-key encryption scheme with message space $\mathcal{M}_\lambda = \{0,1\}^{\ell(\lambda)}$ and randomness space $\mathcal{R}_\lambda$. We define the TDHF input space as $\mathcal{D}_\lambda := \mathcal{M}_\lambda \times \mathcal{R}_\lambda$. The TDHF algorithms $(\mathsf{KGen}, \mathsf{Eval}, \mathsf{hEval}, \mathsf{pInv})$ are realized using the PKE algorithms $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ as follows.

function $\mathsf{TDHF.KGen}(1^\lambda)$
  $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.KGen}(1^\lambda)$
  return $(\mathsf{pk}, \mathsf{sk})$
end function

function $\mathsf{TDHF.Eval}(\mathsf{pk}, x)$
  parse $x = m \| r$, where $m \in \{0,1\}^{\ell(\lambda)}$
  $y \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, m; r)$
  return y
end function

function $\mathsf{TDHF.hEval}(\mathsf{pk}, x)$
  parse $x = m \| r$, where $m \in \{0,1\}^{\ell(\lambda)}$
  return m
end function

function $\mathsf{TDHF.pInv}(\mathsf{sk}, y)$
  $m \leftarrow \mathsf{PKE.Dec}(\mathsf{sk}, y)$
  return m

4: **end function**

Perfect correctness follows directly from perfect correctness of the underlying public-key encryption scheme. Indeed, for every key pair $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{PKE.KGen}(1^\lambda)$ and every input $x = (m, r) \in \mathcal{M}_\lambda \times \mathcal{R}_\lambda$, we have

$$\begin{aligned}
&\mathsf{TDHF.pInv}\left(\mathsf{sk}, \mathsf{TDHF.Eval}(\mathsf{pk}, x)\right) \\
&= \mathsf{PKE.Dec}\left(\mathsf{sk}, \mathsf{PKE.Enc}(\mathsf{pk}, m; r)\right) \\
&= m \\
&= \mathsf{TDHF.hEval}(\mathsf{pk}, x).
\end{aligned}$$

Theorem 6. *If* PKE *is perfectly correct and IND-CPA secure according to Definition 2, then the construction above is a perfectly correct and secure* TDHF *according to Definition 9 and 10.*

Proof. Perfect correctness follows from the calculation above. We prove TDHF security.

Suppose that there exists a PPT adversary $\mathcal{A}$ that distinguishes the real and random distributions in Definition 10 with non-negligible advantage. We construct an IND-CPA adversary $\mathcal{B}$ against the underlying public-key encryption scheme.

Adversary $\mathcal{B}$ proceeds as follows.

1: **function** $\mathcal{B}$
2: **receive** pk from the IND-CPA challenger
3: $m_0, m_1 \xleftarrow{\$} \{0,1\}^{\ell(\lambda)}$
4: **send** (m_0, m_1) to the IND-CPA challenger
5: **receive** the challenge ciphertext $y^\star$ from the IND-CPA challenger
6: **send** $(\mathsf{pk}, y^\star, m_0)$ to $\mathcal{A}$
7: **receive** d from $\mathcal{A}$
8: **return** d
9: **end function**

Let $\beta \xleftarrow{\$} \{0,1\}$ denote the IND-CPA challenge bit, so that $y^\star = \mathsf{PKE.Enc}(\mathsf{pk}, m_\beta; r)$ for uniformly sampled encryption randomness $r \xleftarrow{\$} \mathcal{R}_\lambda$.

If $\beta = 0$, then $y^\star = \mathsf{PKE.Enc}(\mathsf{pk}, m_0; r) = \mathsf{TDHF.Eval}\left(\mathsf{pk}, (m_0, r)\right)$, and $m_0 = \mathsf{TDHF.hEval}\left(\mathsf{pk}, (m_0, r)\right)$. Since m_0 and r are sampled independently and uniformly, the tuple $(\mathsf{pk}, y^\star, m_0)$ is distributed identically to the real distribution in the TDHF security experiment.

If $\beta = 1$, then $y^\star = \mathsf{PKE.Enc}(\mathsf{pk}, m_1; r)$, while m_0 is uniform and independent of pk, m_1, r, and $y^\star$. Thus, the tuple $(\mathsf{pk}, y^\star, m_0)$ is distributed identically to the random distribution in the TDHF security experiment.

Therefore,

$$\begin{aligned}
&\left|\Pr[\mathcal{A}(\mathsf{pk}, y^\star, m_0) = 1 \mid \beta = 0] - \Pr[\mathcal{A}(\mathsf{pk}, y^\star, m_0) = 1 \mid \beta = 1]\right| \\
&\quad = \mathsf{Adv}^{\mathsf{tdhf}}_{\mathsf{TDHF},\mathcal{A}}(\lambda).
\end{aligned}$$

Consequently, if $\mathcal{A}$ distinguishes the real and random TDHF distributions with non-negligible advantage, then its output distinguishes whether $y^\star$ encrypts m_0 or m_1 with the same non-negligible advantage. This contradicts the IND-CPA security of PKE.

Therefore, the constructed TDHF is secure.

Combining Theorems 5 and 6 establishes the black-box equivalence between perfectly correct, secure TDHFs and perfectly correct, IND-CPA-secure public-key encryption schemes, as stated in Theorem 1.

B Equivalence of Adaptive TDHFs and IND-CCA2 Security

Theorem 7 (One-bit CCA2 encryption from ATDHF). *Let* TDHF *be a perfectly correct and secure one-bit* ATDHF. *Then there exists a black-box construction of a perfectly correct, IND-CCA2-secure public-key encryption scheme with message space* $\{0,1\}$.

Let $\mathsf{TDHF} = (\mathsf{KGen}, \mathsf{Eval}, \mathsf{hEval}, \mathsf{pInv})$ be a perfectly correct and secure one-bit ATDHF with input space $\mathcal{D}_\lambda$. We construct a public-key encryption scheme with message space $\{0,1\}$.

Construction.

function PKE.KGen(1^λ)
  $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{TDHF.KGen}(1^\lambda)$
  return $(\mathsf{pk}, \mathsf{sk})$
end function

function PKE.Enc(pk, m)
  for $i = 1$ to λ **do**
    $x_i \xleftarrow{\$} \mathcal{D}_\lambda$
    $z_i \leftarrow \mathsf{TDHF.hEval}(\mathsf{pk}, x_i)$
    if $z_i = m$ **then**
      $y_i \leftarrow \mathsf{TDHF.Eval}(\mathsf{pk}, x_i)$
      return $(y_i, 0)$
    end if
  end for
  return $(m, 1)$
end function

function PKE.Dec($\mathsf{sk}, (c, \mathsf{flag})$)
  if $\mathsf{flag} = 0$ **then**
    $z \leftarrow \mathsf{TDHF.pInv}(\mathsf{sk}, c)$
    return z
  else if $\mathsf{flag} = 1$ and $c \in \{0,1\}$ **then**
    return c

```
    else
        return ⊥
    end if
end function
```

Perfect correctness. Fix any key pair $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{TDHF.KGen}(1^\lambda)$ and any message $m \in \{0,1\}$. If encryption returns $(m, 1)$, then decryption returns m directly. Otherwise, encryption returns $(y_i, 0)$ for some i such that $y_i = \mathsf{TDHF.Eval}(\mathsf{pk}, x_i)$ and $\mathsf{TDHF.hEval}(\mathsf{pk}, x_i) = m$. By perfect correctness of the TDHF,

$$\begin{aligned}\mathsf{PKE.Dec}(\mathsf{sk}, (y_i, 0)) &= \mathsf{TDHF.pInv}(\mathsf{sk}, y_i) \\ &= \mathsf{TDHF.hEval}(\mathsf{pk}, x_i) \\ &= m.\end{aligned}$$

Thus, the constructed public-key encryption scheme is perfectly correct.

We prove the security via sequence of lemmas. To make discussion simple, we make the following definition. For a TDHF trapdoor sk, define the simulated decryption oracle

$$\mathcal{O}_{\mathsf{PKE},\mathsf{sk}}(c, \mathsf{flag}) := \begin{cases} \mathsf{TDHF.pInv}(\mathsf{sk}, c), & \text{if } \mathsf{flag} = 0, \\ c, & \text{if } \mathsf{flag} = 1 \text{ and } c \in \{0,1\}, \\ \bot, & \text{otherwise.} \end{cases}$$

Lemma 2 (Unpredictability of a fresh evaluation). *Let* TDHF *be a secure one-bit* ATDHF. *Let* $\mathcal{B}$ *be any PPT oracle algorithm that, on input* pk *and with oracle access to* $\mathsf{pInv}(\mathsf{sk}, \cdot)$*, outputs a polynomial-size set* $S \subseteq \mathcal{R}_\lambda$. *For* $X \xleftarrow{\$} \mathcal{D}_\lambda$, $Y := \mathsf{Eval}(\mathsf{pk}, X)$, *the probability* $\Pr[Y \in S]$ *is negligible.*

Proof. Let $p(\lambda) := \Pr[Y \in S]$. We construct an ATDHF adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ whose distinguishing advantage is exactly $p(\lambda)/2$.

```
function A_1^{O_sk}(1^λ, pk)
    S ← B^{O_sk}(1^λ, pk)
    for every y ∈ S do
        v_y ← O_sk(y)
    end for
    return σ := (S, {v_y}_{y∈S})
end function

function A_2(1^λ, pk, σ, Y, Z)
    Parse σ = (S, {v_y}_{y∈S})
    if Y ∈ S and Z = v_Y then
        return 1
    else
        return 0
```

7: **end if**
8: **end function**

– If Z is the real hardcore bit, then, whenever $Y \in S$, perfect correctness gives

$$v_Y = \mathsf{TDHF.pInv}(\mathsf{sk}, Y) = \mathsf{TDHF.hEval}(\mathsf{pk}, X) = Z.$$

Hence,

$$\Pr[\mathcal{A} = 1 \mid Z \text{ is real}] = p(\lambda).$$

– If Z is an independent uniform bit, then, conditioned on $Y \in S$, the equality $Z = v_Y$ holds with probability $1/2$. Therefore,

$$\Pr[\mathcal{A} = 1 \mid Z \text{ is uniform}] = \frac{p(\lambda)}{2}.$$

Consequently,

$$\mathsf{Adv}^{\mathsf{atdhf}}_{\mathsf{TDHF},\mathcal{A}}(\lambda) = \frac{p(\lambda)}{2}.$$

ATDHF security implies that $p(\lambda)$ is negligible. The adversary makes no post-challenge oracle queries and is therefore a valid adversary in the ATDHF experiment.

Lemma 3 (Multi-sample adaptive pseudorandomness). *Let* TDHF *be a secure one-bit* ATDHF*, let $q = q(\lambda)$ be polynomially bounded, and let $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$ be a two-stage PPT adversary. Consider the following two experiments.*

In both experiments, the challenger samples $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$*, runs* $\sigma \leftarrow \mathcal{D}_1^{\mathcal{O}_{\mathsf{sk}}}(1^\lambda, \mathsf{pk})$*, where* $\mathcal{O}_{\mathsf{sk}}(y) := \mathsf{TDHF.pInv}(\mathsf{sk}, y)$*, and samples independently* $X_1, \ldots, X_q \xleftarrow{\$} \mathcal{D}_\lambda$*. It computes* $Y_i := \mathsf{Eval}(\mathsf{pk}, X_i)$ *for every* i*.*

In the real experiment, it sets $Z_i := \mathsf{hEval}(\mathsf{pk}, X_i)$*.*

In the random experiment, it samples independent $Z_i \xleftarrow{\$} \{0,1\}$*.*

The second stage receives the state σ *and the ordered tuple* $((Y_1, Z_1), \ldots, (Y_q, Z_q))$ *and post-challenge oracle access to*

$$\mathcal{O}^{\mathbf{Y}}_{\mathsf{sk}}(y) := \begin{cases} \bot, & \text{if } y \in \{Y_1, \ldots, Y_q\}, \\ \mathsf{pInv}(\mathsf{sk}, y), & \text{otherwise.} \end{cases}$$

where $\mathbf{Y} := (Y_1, \ldots, Y_q)$ *Then the real and random experiments are computationally indistinguishable.*

Proof. For $j \in \{0, \ldots, q\}$, let H_j be the hybrid in which the first j values $Z_1, \ldots, Z_j$ are uniform and the remaining values are real. Thus, H_0 is the real experiment and H_q is the random experiment. For every $j \in \{0, \ldots, q\}$, let $p_j(\lambda) := \Pr[H_j \Rightarrow 1]$. Define

$$\delta(\lambda) := |p_0(\lambda) - p_q(\lambda)|.$$

We construct a single ATDHF adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ with advantage $\frac{\delta(\lambda)}{q(\lambda)}$. The case $q(\lambda) = 0$ is immediate, so assume that $q(\lambda) \geq 1$. The first stage samples $J \xleftarrow{\$} [q(\lambda)]$ and records J in its state.

function $\mathcal{B}_1^{\mathcal{O}_{\mathsf{sk}}}(1^\lambda, \mathsf{pk})$
  $\sigma \leftarrow \mathcal{D}_1^{\mathcal{O}_{\mathsf{sk}}}(1^\lambda, \mathsf{pk})$
  $J \xleftarrow{\$} [q]$
  return (σ, J)
end function

function $\mathcal{B}_2^{\mathcal{O}_{\mathsf{sk}}^{Y^\star}}(1^\lambda, \mathsf{pk}, (\sigma, J), Y^\star, Z^\star)$
  $(Y_J, Z_J) \leftarrow (Y^\star, Z^\star)$
  for $i \in [q(\lambda)] \setminus \{J\}$ **do**
    $X_i \xleftarrow{\$} \mathcal{D}_\lambda$
    $Y_i \leftarrow \mathsf{Eval}(\mathsf{pk}, X_i)$
    if $i < J$ **then**
      $Z_i \xleftarrow{\$} \{0,1\}$
    else
      $Z_i \leftarrow \mathsf{hEval}(\mathsf{pk}, X_i)$
    end if
  end for
  Define

$$\widehat{\mathcal{O}}(y) := \begin{cases} \bot, & \text{if } y \in \{Y_1, \ldots, Y_q\}, \\ \mathcal{O}_{\mathsf{sk}}^{Y^\star}(y), & \text{otherwise.} \end{cases}$$

  $d \leftarrow \mathcal{D}_2^{\widehat{\mathcal{O}}}\left(1^\lambda, \mathsf{pk}, \sigma, (Y_1, Z_1), \ldots, (Y_q, Z_q)\right).$
  return d
end function

Every query forwarded to $\mathcal{O}_{\mathsf{sk}}^{Y^\star}$ differs from $Y^\star = Y_J$ and is therefore legal. Queries equal to any sampled evaluation Y_i, including evaluations that collide with $Y^\star$, are answered locally with $\bot$. Hence, $\widehat{\mathcal{O}}$ perfectly simulates the multi-sample restricted oracle. Conditioned on $J = j$, if $Z^\star$ is real, the simulation is exactly H_{j-1}. If $Z^\star$ is uniform, the simulation is exactly H_j. Therefore,

$$\begin{aligned} &\Pr[\mathcal{B} \Rightarrow 1 \mid Z^\star \text{ is real}] - \Pr[\mathcal{B} \Rightarrow 1 \mid Z^\star \text{ is uniform}] \\ &\quad = \frac{1}{q(\lambda)} \sum_{j=1}^{q(\lambda)} \left(\Pr[H_{j-1} \Rightarrow 1] - \Pr[H_j \Rightarrow 1]\right) \\ &\quad = \frac{\Pr[H_0 \Rightarrow 1] - \Pr[H_q \Rightarrow 1]}{q(\lambda)}. \end{aligned}$$

Taking absolute values gives

$$\mathsf{Adv}^{\mathsf{atdhf}}_{\mathsf{TDHF},\mathcal{B}}(\lambda) = \frac{\delta(\lambda)}{q(\lambda)}.$$

Since $\mathcal{B}$ is a PPT adversary and q is polynomially bounded, ATDHF security implies that

$$\frac{\delta(\lambda)}{q(\lambda)}$$

is negligible. Hence, $\delta(\lambda)$ is negligible, and therefore

$$H_0 \approx_c H_q.$$

This proves multi-sample adaptive pseudorandomness.

During challenge encryption, the algorithm samples several candidate evaluations but reveals only the selected one. We call every distinct unselected candidate evaluation a hidden candidate. The next lemma shows that a PPT CCA2 adversary queries such a hidden value only with negligible probability.

Lemma 4 (Hidden-candidate queries are negligible). *Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a PPT IND-CCA2 adversary against the constructed one-bit encryption scheme.*

Consider experiments indexed by $\delta \in \{\mathsf{real}, \mathsf{rand}\}$.

After running $\mathcal{A}_1$ with genuine decryption-oracle access, the challenger obtains (m_0, m_1, σ), samples $b \xleftarrow{\$} \{0,1\}$, $X_1, \ldots, X_\lambda \xleftarrow{\$} \mathcal{D}_\lambda$, and computes $Y_i := \mathsf{TDHF.Eval}(\mathsf{pk}, X_i)$. In the real experiment,

$$Z_i := \mathsf{TDHF.hEval}(\mathsf{pk}, X_i).$$

In the random experiment,

$$Z_i \xleftarrow{\$} \{0,1\}$$

independently. Set

$$I := \begin{cases} \min\{i \in [\lambda] : Z_i = m_b\}, & \text{if this set is nonempty,} \\ \bot, & \text{otherwise.} \end{cases}$$

If $I \in [\lambda]$, set $\mathsf{ct}^\star := (Y_I, 0)$ and define $\mathcal{H}_I := \{Y_j : j \neq I\} \setminus \{Y_I\}$. If $I = \bot$, set $\mathsf{ct}^\star := (m_b, 1)$ and $\mathcal{H}_I := \{Y_1, \ldots, Y_\lambda\}$. Let Bad be the event that $\mathcal{A}_2$ submits a ciphertext $(y, 0)$ with

$$y \in H_I.$$

Then

$$\Pr[\mathsf{Bad} \mid \delta = \mathsf{real}] = \mathsf{negl}(\lambda)$$

and

$$\Pr[\mathsf{Bad} \mid \delta = \mathsf{rand}] = \mathsf{negl}(\lambda).$$

Proof. We first consider the random experiment. Let $Q(\lambda)$ bound the number of post-challenge decryption queries made by $\mathcal{A}_2$.

Define $\mathcal{P} := \big\{(i,j) : i \in [\lambda] \cup \{\bot\},\ j \in [\lambda],\ \text{and } \big(i = \bot \text{ or } j \neq i\big)\big\}$. Thus, $|\mathcal{P}| = \lambda(\lambda - 1) + \lambda = \lambda^2$. We construct a single PPT oracle algorithm $\mathcal{B}$. It samples $(I^\star, J^\star) \xleftarrow{\$} \mathcal{P}$ and simulates the random experiment while deferring the sampling of the candidate indexed by $J^\star$.

function $\mathcal{B}^{\mathcal{O}_{\mathsf{sk}}}(1^\lambda, \mathsf{pk})$
  $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{PKE},\mathsf{sk}}}(1^\lambda, \mathsf{pk})$
  $(I^\star, J^\star) \xleftarrow{\$} \mathcal{P}$
  $b \xleftarrow{\$} \{0,1\}$
  $Z_1, \ldots, Z_\lambda \xleftarrow{\$} \{0,1\}$
  Compute I from $(Z_1, \ldots, Z_\lambda, m_b)$
  if $I \neq I^\star$ **then**
    return $\varnothing$
  end if
  if $I^\star \in [\lambda]$ **then**
    $X_{I^\star} \xleftarrow{\$} \mathcal{D}_\lambda$
    $Y_{I^\star} \leftarrow \mathsf{Eval}(\mathsf{pk}, X_{I^\star})$
    $\mathsf{ct}^\star \leftarrow (Y_{I^\star}, 0)$
  else
    $\mathsf{ct}^\star \leftarrow (m_b, 1)$
  end if
  Initialize $S \leftarrow \varnothing$
  Run $\mathcal{A}_2(\mathsf{ct}^\star, \sigma)$ and answer each decryption query ct as follows:
  if $\mathsf{ct} = \mathsf{ct}^\star$ **then**
    Return $\perp$ to $\mathcal{A}_2$
  else if $\mathsf{ct} = (y, 0)$ **then**
    $S \leftarrow S \cup \{y\}$
    Return $\mathcal{O}_{\mathsf{sk}}(y)$ to $\mathcal{A}_2$
  else if $\mathsf{ct} = (m, 1)$ for some $m \in \{0,1\}$ **then**
    Return m to $\mathcal{A}_2$
  else
    Return $\perp$ to $\mathcal{A}_2$
  end if
  Ignore the final output of $\mathcal{A}_2$
  return S
end function

The set S contains the first component of every flag-zero query other than the exact challenge ciphertext. Since $\mathcal{A}_2$ makes at most $Q(\lambda)$ post-challenge queries, $|S| \leq Q(\lambda)$. The set S is generated without sampling $X_{J^\star}$. Indeed, in the random experiment, the selected index I depends only on $Z_1, \ldots, Z_\lambda$ and m_b. Moreover, if $I^\star \in [\lambda]$, then the definition of $\mathcal{P}$ ensures that $J^\star \neq I^\star$. Hence, the deferred candidate is not used in the challenge ciphertext.

After $\mathcal{B}$ terminates, sample $X_{J^\star} \xleftarrow{\$} \mathcal{D}_\lambda$ and define $Y_{J^\star} := \mathsf{Eval}(\mathsf{pk}, X_{J^\star})$. The value $Y_{J^\star}$ is therefore an independently sampled fresh evaluation relative to the set S. By Lemma 2, $\Pr[Y_{J^\star} \in S] = \mathsf{negl}(\lambda)$.

Whenever Bad occurs in the random experiment, there exists at least one pair $(i, j) \in \mathcal{P}$ such that $I = i$, the adversary submits the query $(Y_j, 0)$, and, if $i \in [\lambda]$, then $Y_j \neq Y_i$. For any such witness pair, if $(I^\star, J^\star) = (i, j)$, then $Y_{J^\star} \in S$.

Since $(I^\star, J^\star)$ is uniform over $\mathcal{P}$,

$$\Pr[Y_{J^\star} \in S] \geq \frac{\Pr[\mathsf{Bad} \mid \delta = \mathsf{rand}]}{|\mathcal{P}|}.$$

Since $|\mathcal{P}| = \lambda^2$ and $\Pr[Y_{J^\star} \in S] = \mathsf{negl}(\lambda)$, we conclude that

$$\Pr[\mathsf{Bad} \mid \delta = \mathsf{rand}] = \mathsf{negl}(\lambda).$$

We next transfer the bound from the random experiment to the real experiment using multi-sample adaptive pseudorandomness.

function $\mathcal{D}^{\mathcal{O}_{\mathsf{sk}}}_{\mathsf{Bad},1}(1^\lambda, \mathsf{pk})$
  $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{PKE},\mathsf{sk}}}(1^\lambda, \mathsf{pk})$
  $\tau \leftarrow (m_0, m_1, \sigma)$
  return τ
end function

Let

$$\mathbf{Y} := (Y_1, \ldots, Y_\lambda).$$

The multi-sample restricted oracle is

$$\mathcal{O}^{\mathbf{Y}}_{\mathsf{sk}}(y) := \begin{cases} \perp, & \text{if } y \in \{Y_1, \ldots, Y_\lambda\}, \\ \mathsf{Plnv}(\mathsf{sk}, y), & \text{otherwise.} \end{cases}$$

function $\mathcal{D}^{\mathcal{O}^{\mathbf{Y}}_{\mathsf{sk}}}_{\mathsf{Bad},2}(1^\lambda, \mathsf{pk}, \tau, \big((Y_1, Z_1), \ldots, (Y_\lambda, Z_\lambda)\big))$
  Parse $\tau = (m_0, m_1, \sigma)$
  $b \xleftarrow{\$} \{0,1\}$
  $$I \leftarrow \begin{cases} \min\{i \in [\lambda] : Z_i = m_b\}, & \text{if the set is nonempty,} \\ \perp, & \text{otherwise.} \end{cases}$$
  if $I \in [\lambda]$ **then**
    $\mathsf{ct}^\star \leftarrow (Y_I, 0)$
    $H_I \leftarrow \{Y_j : j \neq I\} \setminus \{Y_I\}$
  else
    $\mathsf{ct}^\star \leftarrow (m_b, 1)$
    $H_I \leftarrow \{Y_1, \ldots, Y_\lambda\}$
  end if
  Run $\mathcal{A}_2(\mathsf{ct}^\star, \sigma)$ and answer each decryption query ct as follows:
  if $\mathsf{ct} = \mathsf{ct}^\star$ **then**
    Return $\perp$ to $\mathcal{A}_2$
  else if $\mathsf{ct} = (y, 0)$ and $y \in H_I$ **then**
    return 1
  else if $\mathsf{ct} = (y, 0)$ and $y \notin \{Y_1, \ldots, Y_\lambda\}$ **then**

$w \leftarrow \mathcal{O}_{\mathsf{sk}}^{\mathbf{Y}}(y)$
Return w to $\mathcal{A}_2$
else if $\mathsf{ct} = (m, 1)$ for some $m \in \{0,1\}$ **then**
Return m to $\mathcal{A}_2$
else
Return $\perp$ to $\mathcal{A}_2$
end if
Ignore the final output of $\mathcal{A}_2$
return 0
end function

The query simulation covers every possible ciphertext. The exact challenge ciphertext is rejected, as required by the IND-CCA2 experiment. A flag-zero query on a hidden candidate causes $\mathcal{D}_{\mathsf{Bad}}$ to output 1. A flag-zero query whose first component lies outside the protected set is forwarded to $\mathcal{O}_{\mathsf{sk}}^{\mathbf{Y}}$. Any remaining flag-zero query has first component in $\{Y_1, \ldots, Y_\lambda\}$ but not in H_I. If $I \in [\lambda]$, such a value must equal Y_I, possibly because several candidate evaluations collide. The corresponding ciphertext is therefore the exact challenge ciphertext and is answered with $\perp$. Consequently, every query forwarded to $\mathcal{O}_{\mathsf{sk}}^{\mathbf{Y}}$ lies outside the protected set and is legal.

If the multi-sample challenge values are real, then $Z_i = \mathsf{hEval}(\mathsf{pk}, X_i)$ for every $i \in [\lambda]$. Hence, $\mathcal{D}_{\mathsf{Bad}}$ perfectly simulates the experiment with $\delta = \mathsf{real}$. If the challenge values are independent uniform bits, then $\mathcal{D}_{\mathsf{Bad}}$ perfectly simulates the experiment with $\delta = \mathsf{rand}$. In either case, $\mathcal{D}_{\mathsf{Bad}}$ outputs 1 exactly when Bad occurs.

By Lemma 3, $|\Pr[\mathsf{Bad} \mid \delta = \mathsf{real}] - \Pr[\mathsf{Bad} \mid \delta = \mathsf{rand}]| = \mathsf{negl}(\lambda)$. We have already shown that $\Pr[\mathsf{Bad} \mid \delta = \mathsf{rand}] = \mathsf{negl}(\lambda)$. Therefore,

$$\Pr[\mathsf{Bad} \mid \delta = \mathsf{real}] = \mathsf{negl}(\lambda).$$

Proof (Proof of Theorem 7). Perfect correctness follows from above note on *Perfect correctness*. We prove IND-CCA2 security.

The proof has three steps. First, we abort when the adversary queries a hidden candidate evaluation; the preceding lemma shows that this changes the game only negligibly. Second, we replace all candidate hardcore bits by independent uniform bits using multi-sample adaptive pseudorandomness. Third, we remove the negligible fallback event. The resulting challenge ciphertext is independent of the challenge bit.

Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a PPT IND-CCA2 adversary.

Sampling all λ candidate inputs in advance does not change the distribution of the challenge ciphertext, because the samples are independent and encryption uses only the prefix ending at the first matching index.

Each game outputs 1 if $\mathcal{A}_2$'s final guess equals the challenge bit and outputs 0 otherwise. An aborting game outputs an independent uniform bit.

G_0**.** The real IND-CCA2 experiment.

G_0'**.** As G_0, but terminate with an independent uniform output when Bad occurs.

G_1'. As G_0', but replace all candidate hardcore bits by independent uniform bits.

G_1. As G_1', but do not abort upon Bad.

G_2. As G_1, but terminate with an independent uniform output when none of the λ uniform candidate bits matches the challenge message.

The only nontrivial computational transition is $G_0' \approx_c G_1'$. It is witnessed by the following multi-sample distinguisher.

function $\mathcal{D}^{\mathcal{O}_{\mathsf{sk}}}_{\mathsf{hyb},1}(1^\lambda, \mathsf{pk})$
  $(m_0, m_1, \sigma) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{PKE},\mathsf{sk}}}(1^\lambda, \mathsf{pk})$
  return (m_0, m_1, σ)
end function

function $\mathcal{D}^{\mathcal{O}_{\mathsf{sk}}^{Y_1,\ldots,Y_\lambda}}_{\mathsf{hyb},2}(1^\lambda, \mathsf{pk}, m_0, m_1, \sigma, (Y_1, Z_1), \ldots, (Y_\lambda, Z_\lambda))$
  Sample $b \xleftarrow{\$} \{0,1\}$
  Compute I, $\mathsf{ct}^\star$, and $\mathcal{H}_I$
  Run $\mathcal{A}_2(\mathsf{ct}^\star, \sigma)$ and answer each decryption query ct as follows:
  if $\mathsf{ct} = \mathsf{ct}^\star$ **then**
    Return $\perp$ to $\mathcal{A}_2$
  else if $\mathsf{ct} = (y, 0)$ and $y \in \mathcal{H}_I$ **then**
    $\beta \xleftarrow{\$} \{0,1\}$
    return β // abort as in G_0' and G_1'
  else if $\mathsf{ct} = (y, 0)$ and $y \notin \{Y_1, \ldots, Y_\lambda\}$ **then**
    $w \leftarrow \mathcal{O}_{\mathsf{sk}}^{Y_1,\ldots,Y_\lambda}(y)$
    Return w to $\mathcal{A}_2$
  else if $\mathsf{ct} = (m, 1)$ for some $m \in \{0,1\}$ **then**
    Return m to $\mathcal{A}_2$
  else
    Return $\perp$ to $\mathcal{A}_2$
  end if
  Let b' be the final output of $\mathcal{A}_2$
  if $b' = b$ **then**
    return 1
  else
    return 0
  end if
end function

By Lemma 4,

$$|\Pr[G_0 \Rightarrow 1] - \Pr[G_0' \Rightarrow 1]| = \mathsf{negl}(\lambda)$$

and

$$|\Pr[G_1' \Rightarrow 1] - \Pr[G_1 \Rightarrow 1]| = \mathsf{negl}(\lambda).$$

If the multi-sample challenge values supplied to $\mathcal{D}_{\mathsf{hyb}}$ are real, then the reduction simulates G'_0. If they are uniform, it simulates G'_1. Therefore, Lemma 3 gives

$$|\Pr[G'_0 \Rightarrow 1] - \Pr[G'_1 \Rightarrow 1]| = \mathsf{negl}(\lambda).$$

The games G_1 and G_2 differ only if none of the λ uniform candidate bits matches the challenge message. Hence,

$$|\Pr[G_1 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1]| \leq 2^{-\lambda}.$$

In G_2, for every $m \in \{0,1\}$ and $i \in [\lambda]$,

$$\begin{aligned}\Pr[I = i \mid m_b = m] &= \Pr[U_1 \neq m, \ldots, U_{i-1} \neq m, U_i = m] \\ &= 2^{-i}.\end{aligned}$$

Thus, the distribution of I is independent of m_b.

The evaluations $Y_1, \ldots, Y_\lambda$ are independent of the uniform candidate bits. Consequently, conditioned on $I = i$, the selected evaluation $Y_I = Y_i$ is independent of m_b.

The first-stage transcript is generated before b is sampled. The challenge ciphertext is independent of b, and the restricted decryption oracle is a deterministic function of sk and $\mathsf{ct}^\star$. Therefore, by induction over the adaptive post-challenge queries, the entire view of $\mathcal{A}_2$ is independent of b.

Thus,

$$\Pr[G_2 \Rightarrow 1] = \frac{1}{2}.$$

Combining the transitions gives

$$\begin{aligned}\left|\Pr[G_0 \Rightarrow 1] - \frac{1}{2}\right| &\leq |\Pr[G_0 \Rightarrow 1] - \Pr[G'_0 \Rightarrow 1]| \\ &\quad + |\Pr[G'_0 \Rightarrow 1] - \Pr[G'_1 \Rightarrow 1]| \\ &\quad + |\Pr[G'_1 \Rightarrow 1] - \Pr[G_1 \Rightarrow 1]| \\ &\quad + |\Pr[G_1 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1]| \\ &= \mathsf{negl}(\lambda).\end{aligned}$$

Therefore, the constructed one-bit public-key encryption scheme is IND-CCA2 secure.

Lemma 5 (Hardcore-bit projection). *Every perfectly correct and secure* ATDHF *with hardcore range* $\{0,1\}^{\ell(\lambda)}$ *implies a perfectly correct and secure one-bit* ATDHF *through a black-box construction.*

Proof. Let $\mathsf{TDHF} = (\mathsf{KGen}, \mathsf{Eval}, \mathsf{hEval}, \mathsf{pInv})$ be an ATDHF with hardcore range $\{0,1\}^{\ell(\lambda)}$.

Define a one-bit TDHF: $\mathsf{TDHF}_1 = (\mathsf{KGen}, \mathsf{Eval}, \mathsf{hEval}_1, \mathsf{pInv}_1)$ by $\mathsf{hEval}_1(\mathsf{pk}, x) := \mathrm{first}\,(\mathsf{hEval}(\mathsf{pk}, x))$ and

$$\mathsf{pInv}_1(\mathsf{sk}, y) := \begin{cases} \bot, & \text{if } \mathsf{TDHF.pInv}(\mathsf{sk}, y) = \bot, \\ \mathrm{first}\,(\mathsf{TDHF.pInv}(\mathsf{sk}, y)), & \text{otherwise.} \end{cases}$$

Perfect correctness follows immediately from perfect correctness of the original TDHF.

Suppose that an adversary distinguishes the real hardcore bit from a uniform bit in the ATDHF experiment for TDHF_1. A reduction against the original scheme receives either the real hardcore string or an independently uniform string, projects it to its first bit, and runs the adversary.

Partial-inversion queries are answered by invoking the original partial-inversion oracle, returning $\perp$ if the oracle returns $\perp$, and otherwise returning the first bit of its response.

Projection maps the real hardcore string to the real projected bit and maps an independent uniform string to an independent uniform bit. Thus, the reduction preserves the adversary's distinguishing advantage.

Corollary 7 (CCA2 encryption from ATDHF). *Every perfectly correct and secure* ATDHF *implies perfectly correct, IND-CCA2-secure public-key encryption through a black-box construction.*

Proof (Proof of Corollary 7). By Lemma 5, every secure $\ell(\lambda)$-bit ATDHF implies a secure one-bit ATDHF. By Theorem 7, the latter implies a perfectly correct, IND-CCA2-secure one-bit public-key encryption scheme.

Myers and Shelat [19] show that one-bit IND-CCA2-secure public-key encryption implies arbitrary-length IND-CCA2-secure public-key encryption through a black-box construction. Applying their transformation completes the proof.

Theorem 8. *If* PKE *is perfectly correct and IND-CCA2 secure, then the* TDHF *constructed in Section A.2 is perfectly correct and satisfies* ATDHF *security.*

Proof. Perfect correctness follows from perfect correctness of PKE, as shown in Section A.2. We prove ATDHF security.

Suppose that there exists a two-stage PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that distinguishes the real and random ATDHF experiments with non-negligible advantage. We construct an IND-CCA2 adversary $\mathcal{B}$ against PKE.

function $\mathcal{B}$
  Run $\mathcal{A}_1^{\mathcal{O}_{\mathsf{Dec}}}(1^\lambda, \mathsf{pk})$ and obtain state σ
  $m_0, m_1 \xleftarrow{\$} \{0,1\}^{\ell(\lambda)}$
  send (m_0, m_1) to the IND-CCA2 challenger
  receive the challenge ciphertext $y^\star$
  Run $\mathcal{A}_2^{\mathcal{O}^{y^\star}_{\mathsf{Dec}}}(1^\lambda, \mathsf{pk}, \sigma, y^\star, m_0)$ and obtain d
  if $d = 1$ **then**
    return 0
  else
    return 1
  end if
end function

Here, every partial-inversion query y made by $\mathcal{A}$ is answered by forwarding y to the IND-CCA2 decryption oracle and returning its response. After challenge generation, a query on $y^\star$ is rejected rather than forwarded. This perfectly simulates the restricted partial-inversion oracle because $\mathsf{TDHF.pInv}(\mathsf{sk}, y) = \mathsf{PKE.Dec}(\mathsf{sk}, y)$.

Let $\beta \xleftarrow{\$} \{0,1\}$ be the IND-CCA2 challenge bit, so that $y^\star = \mathsf{PKE.Enc}(\mathsf{pk}, m_\beta; r)$ for uniformly sampled encryption randomness $r \xleftarrow{\$} \mathcal{R}_\lambda$.

If $\beta = 0$, then $y^\star = \mathsf{TDHF.Eval}\,(\mathsf{pk}, (m_0, r))$ and $m_0 = \mathsf{TDHF.hEval}\,(\mathsf{pk}, (m_0, r))$. Thus, $\mathcal{A}$ receives the real ATDHF challenge distribution.

If $\beta = 1$, the challenge evaluation is an encryption of the independent uniform message m_1, while the supplied value m_0 is uniform and independent of $y^\star$. Thus, $\mathcal{A}$ receives the random ATDHF challenge distribution.

The pre- and post-challenge oracle simulations are perfect, and every post-challenge query forwarded by $\mathcal{B}$ differs from the IND-CCA2 challenge ciphertext $y^\star$. Hence, a non-negligible distinguishing advantage for $\mathcal{A}$ yields a non-negligible IND-CCA2 advantage for one of the two possible output orientations of $\mathcal{B}$, contradicting the assumed IND-CCA2 security of PKE.

Therefore, the constructed TDHF satisfies ATDHF security.

Combining Corollary 7 and Theorem 8 establishes the black-box equivalence between perfectly correct and secure ATDHFs and perfectly correct, IND-CCA2-secure public-key encryption schemes.

C Equivalence of Lossy Encryption and LTDHF

In this section, we prove that perfectly correct and secure lossy trapdoor hardcore functions are equivalent to perfectly correct and secure lossy public-key encryption schemes. We establish the equivalence through black-box constructions in both directions.

Theorem 9 (Characterization of lossy encryption). *Perfectly correct and secure lossy trapdoor hardcore functions are black-box equivalent to perfectly correct and secure lossy public-key encryption schemes.*

C.1 Lossy Encryption from Lossy TDHFs

Let $\mathsf{LTDHF} = (\mathsf{KGen}_{\mathsf{rec}}, \mathsf{KGen}_{\mathsf{los}}, \mathsf{Eval}, \mathsf{hEval}, \mathsf{pInv})$ be a perfectly correct and secure lossy trapdoor hardcore function. Let the input space of the LTDHF be $\mathcal{D}_\lambda$, and let its hardcore range be $\{0,1\}^{\ell(\lambda)}$. We construct a lossy public-key encryption scheme $\mathsf{LPKE} = (\mathsf{KGen}_{\mathsf{inj}}, \mathsf{KGen}_{\mathsf{lossy}}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}_\lambda := \{0,1\}^{\ell(\lambda)}$ and encryption-randomness space $\mathcal{D}_\lambda$.

The construction is the natural analogue of the public-key encryption construction in Section A.1. The hardcore value is used as a one-time pad for the message.

Construction.

```
function LPKE.KGen_inj(1^λ)
    (pk_i, sk_i) ← LTDHF.KGen_rec(1^λ)
    return (pk_i, sk_i)
end function
```

```
function LPKE.KGen_lossy(1^λ)
    pk_ℓ ← LTDHF.KGen_los(1^λ)
    return pk_ℓ
end function
```

```
function LPKE.Enc(pk, m, r)
    ct_1 ← LTDHF.Eval(pk, r)
    z ← LTDHF.hEval(pk, r)
    ct_2 ← z ⊕ m
    return (ct_1, ct_2)
end function
```

```
function LPKE.Dec(sk_i, (ct_1, ct_2))
    z ← LTDHF.pInv(sk_i, ct_1)
    if z = ⊥ then
        return ⊥
    end if
    return z ⊕ ct_2
end function
```

Lemma 6 (Perfect correctness). *If the underlying* LTDHF *is perfectly correct, then the lossy public-key encryption scheme constructed above is perfectly correct in the recoverable mode.*

Proof. Fix any key pair $(\mathsf{pk}_i, \mathsf{sk}_i) \in \mathrm{Supp}\left(\mathsf{LTDHF.KGen}_{\mathsf{rec}}(1^\lambda)\right)$, any message $m \in \{0,1\}^{\ell(\lambda)}$, and any encryption randomness $r \in \mathcal{D}_\lambda$.

Encryption produces $\mathsf{ct}_1 = \mathsf{LTDHF.Eval}(\mathsf{pk}_i, r)$ and $\mathsf{ct}_2 = \mathsf{LTDHF.hEval}(\mathsf{pk}_i, r) \oplus m$. By perfect correctness of the LTDHF,

$$\begin{aligned}\mathsf{LTDHF.pInv}(\mathsf{sk}_i, \mathsf{ct}_1) &= \mathsf{LTDHF.pInv}\left(\mathsf{sk}_i, \mathsf{LTDHF.Eval}(\mathsf{pk}_i, r)\right)\\ &= \mathsf{LTDHF.hEval}(\mathsf{pk}_i, r).\end{aligned}$$

Therefore, decryption outputs

$$\begin{aligned}&\mathsf{LTDHF.hEval}(\mathsf{pk}_i, r) \oplus (\mathsf{LTDHF.hEval}(\mathsf{pk}_i, r) \oplus m)\\ &= m.\end{aligned}$$

Thus, the constructed lossy public-key encryption scheme is perfectly correct.

Lemma 7 (Public-key mode indistinguishability). *If the recoverable-mode and lossy-mode public keys of the underlying* LTDHF *are computationally indistinguishable, then the recoverable-mode and lossy-mode public keys of the constructed lossy public-key encryption scheme are computationally indistinguishable.*

Proof. The constructed lossy public-key encryption scheme uses the underlying LTDHF public keys without modification. Therefore, any PPT adversary that distinguishes the two public-key modes of the encryption scheme also distinguishes the corresponding public-key modes of the underlying LTDHF with the same advantage.

Lemma 8 (Ciphertext lossiness). *For every security parameter λ, every lossy public key $\mathsf{pk}_\ell \in \mathrm{Supp}\left(\mathsf{KGen}_{\mathsf{los}}(1^\lambda)\right)$, and every pair of messages $m_0, m_1 \in \{0,1\}^{\ell(\lambda)}$, we have*

$$\mathsf{LPKE.Enc}(\mathsf{pk}_\ell, m_0; X_0) \approx_s \mathsf{LPKE.Enc}(\mathsf{pk}_\ell, m_1; X_1),$$

where $X_0, X_1 \xleftarrow{\$} \mathcal{D}_\lambda$ are sampled independently.

Proof. Fix a lossy public key $\mathsf{pk}_\ell \in \mathrm{Supp}\left(\mathsf{KGen}_{\mathsf{los}}(1^\lambda)\right)$ and messages $m_0, m_1 \in \{0,1\}^{\ell(\lambda)}$. Let $X_0, X_1 \xleftarrow{\$} \mathcal{D}_\lambda$ and $U_0, U_1 \xleftarrow{\$} \{0,1\}^{\ell(\lambda)}$ be mutually independent.

Consider the following distributions:

$$\begin{aligned}
H_0 &:= (\mathsf{Eval}(\mathsf{pk}_\ell, X_0), \mathsf{hEval}(\mathsf{pk}_\ell, X_0) \oplus m_0)\,,\\
H_1 &:= (\mathsf{Eval}(\mathsf{pk}_\ell, X_0), U_0 \oplus m_0)\,,\\
H_2 &:= (\mathsf{Eval}(\mathsf{pk}_\ell, X_1), U_1 \oplus m_1)\,,\\
H_3 &:= (\mathsf{Eval}(\mathsf{pk}_\ell, X_1), \mathsf{hEval}(\mathsf{pk}_\ell, X_1) \oplus m_1)\,.
\end{aligned}$$

By statistical lossiness of the LTDHF under the fixed key pk_ℓ,

$$H_0 \approx_s H_1 \qquad \text{and} \qquad H_2 \approx_s H_3.$$

Moreover,

$$H_1 \equiv H_2.$$

Indeed, $\mathsf{Eval}(\mathsf{pk}_\ell, X_0) \equiv \mathsf{Eval}(\mathsf{pk}_\ell, X_1)$, because X_0 and X_1 are identically distributed, and $U_0 \oplus m_0 \equiv U_1 \oplus m_1 \equiv U_{\ell(\lambda)}$. In each distribution, the uniform second component is independent of the first component.

Therefore, by the triangle inequality, $H_0 \approx_s H_3$. Finally, $H_0 \equiv \mathsf{LPKE.Enc}(\mathsf{pk}_\ell, m_0; X_0)$ and $H_3 \equiv \mathsf{LPKE.Enc}(\mathsf{pk}_\ell, m_1; X_1)$. Hence, the constructed encryption scheme satisfies ciphertext lossiness for every lossy public key in the support of lossy key generation.

Theorem 10. *Every perfectly correct and secure* LTDHF *yields a perfectly correct and secure lossy public-key encryption scheme through the black-box construction above.*

Proof. Perfect correctness follows from Lemma 6. Public-key mode indistinguishability follows from Lemma 7. Ciphertext lossiness follows from Lemma 8.

C.2 Lossy TDHFs from Lossy Encryption

Let $\mathsf{LPKE} = (\mathsf{KGen}_{\mathsf{inj}}, \mathsf{KGen}_{\mathsf{lossy}}, \mathsf{Enc}, \mathsf{Dec})$ be a perfectly correct and secure lossy public-key encryption scheme with message space $\mathcal{M}_\lambda = \{0,1\}^{\ell(\lambda)}$ and randomness space $\mathcal{R}_\lambda$.

We define an LTDHF with input space $\mathcal{D}_\lambda := \mathcal{M}_\lambda \times \mathcal{R}_\lambda$ and hardcore range $\{0,1\}^{\ell(\lambda)}$.

Construction.

function $\mathsf{LTDHF.KGen}_{\mathsf{rec}}(1^\lambda)$
  $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{LPKE.KGen}_{\mathsf{inj}}(1^\lambda)$
  return $(\mathsf{pk}_i, \mathsf{sk}_i)$
end function

function $\mathsf{LTDHF.KGen}_{\mathsf{los}}(1^\lambda)$
  $\mathsf{pk}_\ell \leftarrow \mathsf{LPKE.KGen}_{\mathsf{lossy}}(1^\lambda)$
  return pk_ℓ
end function

function $\mathsf{LTDHF.Eval}(\mathsf{pk}, x)$
  parse $x = m \| r$
  $\mathsf{ct} \leftarrow \mathsf{LPKE.Enc}(\mathsf{pk}, m; r)$
  return ct
end function

function $\mathsf{LTDHF.hEval}(\mathsf{pk}, x)$
  parse $x = m \| r$
  return m
end function

function $\mathsf{LTDHF.pInv}(\mathsf{sk}_i, \mathsf{ct})$
  $m \leftarrow \mathsf{LPKE.Dec}(\mathsf{sk}_i, \mathsf{ct})$
  return m
end function

Lemma 9 (Perfect correctness). *If* LPKE *is perfectly correct, then the constructed* LTDHF *is perfectly correct in its recoverable mode.*

Proof. Fix any recoverable-mode key pair $(\mathsf{pk}_i, \mathsf{sk}_i) \in \mathrm{Supp}\left(\mathsf{LPKE.KGen}_{\mathsf{inj}}(1^\lambda)\right)$ and any input $x = (m, r) \in \mathcal{M}_\lambda \times \mathcal{R}_\lambda$. Then

$$\begin{aligned}
&\mathsf{LTDHF.pInv}\left(\mathsf{sk}_i, \mathsf{LTDHF.Eval}(\mathsf{pk}_i, x)\right) \\
&= \mathsf{LPKE.Dec}\left(\mathsf{sk}_i, \mathsf{LPKE.Enc}(\mathsf{pk}_i, m; r)\right) \\
&= m \\
&= \mathsf{LTDHF.hEval}(\mathsf{pk}_i, x).
\end{aligned}$$

Thus, the constructed LTDHF is perfectly correct.

Lemma 10 (Public-key mode indistinguishability). *If the recoverable-mode and lossy-mode public keys of* LPKE *are computationally indistinguishable, then the corresponding public-key modes of the constructed* LTDHF *are computationally indistinguishable.*

Proof. The construction uses the lossy-encryption public keys without modification. Therefore, any PPT adversary that distinguishes the two public-key modes of the constructed LTDHF also distinguishes the corresponding public-key modes of LPKE with the same advantage.

Lemma 11 (Pairwise closeness implies near-independence). *Let* $\{D_m\}_{m\in\mathcal{M}}$ *be a finite family of distributions such that* $\Delta(D_m, D_{m'}) \le \varepsilon$ *for every* $m, m' \in \mathcal{M}$. *Let* $M, U \xleftarrow{\$} \mathcal{M}$ *be independent and uniform, and sample* $C \leftarrow D_M$. *Then*

$$(C, M) \approx_s (C, U).$$

Proof. Let $N := |\mathcal{M}|$. For every $m \in \mathcal{M}$ and $\omega \in \Omega$, write

$$p_m(\omega) := \Pr[D_m = \omega].$$

Define the average distribution

$$\overline{D} := \frac{1}{N} \sum_{m\in\mathcal{M}} D_m,$$

whose probability mass function is

$$\overline{p}(\omega) := \frac{1}{N} \sum_{m\in\mathcal{M}} p_m(\omega).$$

For every $\omega \in \Omega$ and $z \in \mathcal{M}$, we have

$$\Pr[(C, M) = (\omega, z)] = \frac{1}{N} p_z(\omega),$$

because M is uniform and, conditioned on $M = z$, the random variable C is distributed according to D_z. The marginal distribution of C is $\overline{D}$. Since U is uniform and independent of (M, C),

$$\Pr[(C, U) = (\omega, z)] = \frac{1}{N} \overline{p}(\omega).$$

It follows that

$$\begin{aligned} \Delta\left((C, M), (C, U)\right) &= \frac{1}{2} \sum_{z\in\mathcal{M}} \sum_{\omega\in\Omega} \left| \frac{1}{N} p_z(\omega) - \frac{1}{N} \overline{p}(\omega) \right| \\ &= \frac{1}{N} \sum_{z\in\mathcal{M}} \Delta(D_z, \overline{D}). \end{aligned}$$

For every fixed $z \in \mathcal{M}$, convexity of statistical distance gives

$$\begin{aligned}\Delta(D_z, \overline{D}) &= \Delta\left(D_z, \frac{1}{N}\sum_{m\in\mathcal{M}} D_m\right)\\ &\le \frac{1}{N}\sum_{m\in\mathcal{M}} \Delta(D_z, D_m)\\ &\le \varepsilon.\end{aligned}$$

Therefore,

$$\begin{aligned}\Delta\left((C,M),(C,U)\right) &= \frac{1}{N}\sum_{z\in\mathcal{M}} \Delta(D_z, \overline{D})\\ &\le \frac{1}{N}\sum_{z\in\mathcal{M}} \varepsilon\\ &= \varepsilon.\end{aligned}$$

Lemma 12 (Statistical lossiness). *If* LPKE *satisfies ciphertext lossiness according to Definition 7, then the constructed* LTDHF *satisfies statistical lossiness.*

Proof. Fix a security parameter λ and an arbitrary lossy public key $\mathsf{pk}_\ell \in \mathrm{Supp}\left(\mathsf{LPKE.KGen}_{\mathsf{lossy}}(1^\lambda)\right)$.

Let $\mathcal{M}_\lambda = \{0,1\}^{\ell(\lambda)}$. For every message $m \in \mathcal{M}_\lambda$, define the ciphertext distribution

$$D_m := \mathsf{LPKE.Enc}\left(\mathsf{pk}_\ell, m; R\right),$$

where $R \xleftarrow{\$} \mathcal{R}_\lambda$. By ciphertext lossiness, for every $m, m' \in \mathcal{M}_\lambda$, we have

$$\Delta(D_m, D_{m'}) \le \varepsilon(\lambda)$$

for some negligible function ε.

Let $M \xleftarrow{\$} \mathcal{M}_\lambda$, $R \xleftarrow{\$} \mathcal{R}_\lambda$, $U \xleftarrow{\$} \mathcal{M}_\lambda$ be independent, and define $C := \mathsf{LPKE.Enc}\left(\mathsf{pk}_\ell, M; R\right)$. By Lemma 11,

$$(C, M) \approx_s (C, U).$$

More precisely,

$$\Delta\left((C,M),(C,U)\right) \le \varepsilon(\lambda).$$

In the constructed LTDHF, the input space is $\mathcal{D}_\lambda = \mathcal{M}_\lambda \times \mathcal{R}_\lambda$, and $\mathsf{LTDHF.Eval}\left(\mathsf{pk}_\ell, (M,R)\right) = \mathsf{LPKE.Enc}\left(\mathsf{pk}_\ell, M; R\right) = C$, while $\mathsf{LTDHF.hEval}\left(\mathsf{pk}_\ell, (M,R)\right) = M$. Therefore,

$$\begin{aligned}&\left(\mathsf{LTDHF.Eval}\left(\mathsf{pk}_\ell, (M,R)\right), \mathsf{LTDHF.hEval}\left(\mathsf{pk}_\ell, (M,R)\right)\right)\\ &\quad\approx_s \left(\mathsf{LTDHF.Eval}\left(\mathsf{pk}_\ell, (M,R)\right), U\right).\end{aligned}$$

The argument holds for every lossy public key $\mathsf{pk}_\ell \in \mathrm{Supp}\left(\mathsf{LPKE.KGen}_{\mathsf{lossy}}(1^\lambda)\right)$. Hence, the constructed LTDHF satisfies statistical lossiness according to Definition 15.

Theorem 11. *Every perfectly correct and secure lossy public-key encryption scheme yields a perfectly correct and secure* LTDHF *through the black-box construction above.*

Proof. Perfect correctness follows from Lemma 9. Public-key mode indistinguishability follows from Lemma 10. Statistical lossiness follows from Lemma 12.

Conclusion. Combining Theorems 10 and 11 proves Theorem 9.