STRONGER SECURITY FOR PRACTICAL ENCRYPTION SCHEMES

A Dissertation
submitted to the Faculty of the
Graduate School of Arts and Sciences
of Georgetown University
in partial fulfillment of the requirements for the
degree of
Doctor of Philosophy
in Computer Science

By

Mohammad Zaheri Darkahi, M.S.

Washington, DC
August 4, 2020
Stronger Security for Practical Encryption Schemes

Mohammad Zaheri Darkahi, M.S.

Dissertation Advisors: Adam O’Neill, Ph.D. and Kobbi Nissim, Ph.D.

Abstract

Despite recent advances in cryptography, security analyses of encryption schemes fall short of ruling out some possible attacks. Here we study two such types of attacks: selective-opening attacks (SOA) and attacks making use of the code of hash functions employed by the protocol rather than treating them as “black-boxes.”

Selective opening attacks are attacks where the adversary has the ability to “open” the ciphertexts of its choice. For instance, consider the scenario where the adversary sees the encrypted emails sent to Alice and is able to break into some of the senders’ machines and obtain the underlying emails. The question is what one can say about security of the other encrypted emails. The study of such attacks was initiated by Dwork et al. (Journal of the ACM 2003). However, it was only considered for the randomized encryption (R-PKE, where encryption algorithm is randomized). Here we extend the study of selective opening attacks to deterministic encryption (D-PKE, where encryption algorithm is deterministic).

Bellare, Dowley, and Keelveedhi (PKC 2015) were first to study SOA in the deterministic setting and showed that a certain “simulation-based” definition of selective-opening security (SOA) is impossible to achieve for D-PKE. However, their notion seems overly demanding and they left it open to formulate an achievable definition. We start our work on SOA in the deterministic setting by answering this open question and giving a new “comparison-based” security notion which we call D-SO-CPA. We then proceed to give constructions meeting this new notion.
We next explore attacks making use of the code of hash functions. Hash functions are very important building blocks of practical schemes used on Internet. Unfortunately, many practical schemes are not shown to be immune to attacks making use of the code of hash functions that practical schemes use. Rather, they are shown to be secure in a model where hash functions are “truly random” functions. This model is called random oracle (RO) model and was first introduced by Bellare and Rogaway (CCS 1993). The RO model has been enormously enabling in the design of practical protocols for various goals; examples include public-key encryption, digital signatures, and identity-based encryption. However, Canetti et al. (Journal of the ACM 2004) show that there exist RO model schemes for which any instantiation of truly random functions yields a scheme that can be broken efficiently by the adversary that has access to the code of instantiated hash functions. This result shows that practical schemes are not necessarily immune to such attacks.

We study effects of such attacks on practical schemes. Specifically, we study the classical encryption transforms OAEP (EUROCRYPT 1994) and (slightly tweaked) Fujasaki-Okamoto (EUROCRYPT 1998). We show that, under suitable assumptions, there exist standard model hash functions that suffice to prevent such attacks on these encryption transforms. Our hash functions are obtained via a new unified paradigm. The core idea obfuscates an extremely lossy function, a notion introduced by Zhandry (CRYPTO 2016).

**INDEX WORDS:** Public Key Encryption, Selective Opening Security, Fujasaki-Okamoto, RSA-OAEP, Random Oracle, Chosen-Ciphertext Security, Extremely Lossy Functions, Extractable Functions
Acknowledgments

This dissertation would have not been possible without the support of many people. First of all, I would like to thank my advisors Adam O’Neill and Kobbi Nissim. I am so grateful for all of their help, support, insights, and guidance throughout my Ph.D. journey in all matters, both personal and professional. I truly respect and appreciate their knowledge, judgment, and generosity.

I am also grateful to Amos Beimel for his insightful advise and invaluable research lessons. I would also like to thank Ran Canetti and Micah Sherr for their constructive feedback and comments on my dissertation.

I would also like to thank my collaborators, both on works that are in this thesis and outside. I would particularly like to thank Nairen Cao, Viet Tung Hoang and Jonathan Katz. It has been a pleasure to work with and learn from them.

I am deeply fortunate to have the supports of many members of the Computer Science department that without them my Ph.D. would have not been the same. I will try to mention a few and apologize in advance for anyone I might have inadvertently missed. Among them, I would like to thank Arman, Ali, Akshaya, Bart, Eugene, Howard, Jakob, Jianan, Katina, Luca, Rob, Sajad, Sean, Shabnam, Tavish, Yifang and Yuankai. I would also like to specially thank Iman Vakilinia and Mehdi Azvari for being such wonderful friends.

I would not be where I am or who I am today without the unconditional love and support through the years from my parents Neda and Mahmoud, and my brother Ahmad. I am deeply indebted to them.
# Table of Contents

## Chapter 1 Introduction

1.1 Background and Motivation .............................................. 1  
1.2 Our Goals and Approach .............................................. 5  
1.3 Results ........................................................................... 8  
1.3.1 Results on Security Against Selective-Opening Attacks .... 8  
1.3.2 Results on Optimal Asymmetric Encryption Padding ....... 10  
1.3.3 Results on Fujisaki-Okamoto Transform ........................ 18  
1.4 Related and Follow-Up Work ........................................... 20  
1.5 Organization and Credits .............................................. 23  

## Chapter 2 Preliminaries

2.1 Notation and Conventions ................................................ 24  
2.2 Statistical Notions .......................................................... 25  
2.2.1 Statistical Distance ..................................................... 25  
2.2.2 Unpredictable Distribution .......................................... 26  
2.3 Cryptographic Primitives .................................................. 26  
2.3.1 Private-Key Encryption and Its Security ....................... 26  
2.3.2 Public-Key Encryption and Its Security ....................... 28  
2.3.3 Trapdoor Permutations and Their Security ................... 32  
2.3.4 Function Families and Associated Security Notions ......... 35  
2.3.5 The Optimal Asymmetric Encryption Padding Framework ... 39  
2.3.6 The Fujisaki-Okamoto Transform ................................ 41  
2.3.7 Obfuscation .............................................................. 42  
2.3.8 Puncturable Pseudo-Random Functions ......................... 45  
2.3.9 Extremely Lossy Functions ......................................... 46  

## Chapter 3 A Hierarchy of Extractability Notions

3.1 EXT0 Functions .............................................................. 48  
3.2 EXT1 Functions .............................................................. 50  
3.3 EXT2 Functions .............................................................. 50  
3.4 EXT-RO Functions .......................................................... 52  
3.5 Plausibility .................................................................... 53  

## Chapter 4 Results for Padding Schemes and Optimal Asymmetric Encryption Padding

4.1 Scope and Perspective ..................................................... 54  
4.2 Our Results ................................................................. 55
**List of Figures**

2.1 Games to define IND-CCA (left) and INT-CTXT (right) security for private-key encryption ........................................... 27
2.2 Game to define IND-ATK security ........................................ 29
2.3 Game to define PA-RO security ......................................... 30
2.4 Games to define PA1 security ........................................... 31
2.5 Games to define PA2 security ........................................... 32
2.6 Games to define PRG-DIST security ................................. 36
2.7 Games to define HCF-DIST security ................................. 38
2.8 Games to define PHCF-DIST security ............................... 39
2.9 Padding based encryption scheme $\text{PAD}[\mathcal{F}] = (K_{\text{g}}, \text{Enc}, \text{Dec})$ ......................... 40
2.10 OAEP padding scheme $\text{OAEP}[G, H]$ ............................. 41
2.11 FO transform $\text{FO}_{H, G}[\text{PKE}, \text{SE}] = (\text{FO}.K_{\text{g}}, \text{FO}.\text{Enc}, \text{FO}.\text{Dec})$ .......... 41
2.12 Games to define AIPO security ...................................... 45
2.13 Games to define PRF-DIST security ............................... 46
3.1 Game to define EXT1 security ......................................... 49
3.2 Game to define EXT2 security ......................................... 51
3.3 Game to define EXT-RO security ..................................... 52
4.1 Adversaries $B$ and $C$ in the proof of Theorem 6 ......... 56
4.2 Adversary $B$ in the proof of Theorem 7 ......................... 57
4.3 Adversaries $A_{H}, A_{G}$ in the proof of Theorem 8 .... 58
4.4 EXT0 extractor $\text{Ext}$ in the proof of Theorem 8 ......... 59
4.5 EXT1 adversary $A_{G}$ in the proof of Theorem 9 ......... 60
4.6 EXT1 extractor $\text{Ext}$ in the proof of Theorem 9 ......... 61
4.7 EXT-RO extractor $\text{Ext}$ in the proof of Theorem 10 ...... 63
5.1 Games $G_{1}–G_{4}$ in the proof of Theorem 13 ............. 71
5.2 Games $G_{5}, G_{6}$ in the proof of Theorem 13 ............. 72
5.3 Adversary $B$ in the proof of Theorem 13 ....................... 73
5.4 Adversary $C$ in the proof of Theorem 13 ....................... 73
5.5 Inverter $I$ in the proof of Theorem 13 ......................... 74
5.6 Adversary $D$ in the proof of Theorem 13 ....................... 75
5.7 PA-RO extractor $\text{Ext}$ in the proof of Theorem 14 ...... 76
5.8 Adversary $D$ in the proof of Theorem 14 ....................... 78
5.9 Games $G_{1}–G_{4}$ in the proof of Theorem 15 ............. 80
5.10 Adversary $B$ in the proof of Theorem 15 ....................... 81
5.11 PA-RO extractor $\text{Ext}$ in the proof of Theorem 16 ...... 83
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.12</td>
<td>Games to define XOR-ATK security</td>
</tr>
<tr>
<td>5.13</td>
<td>Games $G_1$–$G_3$ in the proof of Theorem 17</td>
</tr>
<tr>
<td>5.14</td>
<td>Adversary $B$ (left) and adversary $C$ (right) in the proof of Theorem 17</td>
</tr>
<tr>
<td>5.15</td>
<td>Games to define XOR-NM security</td>
</tr>
<tr>
<td>5.16</td>
<td>Games $G_1$–$G_4$ in the proof of Theorem 18</td>
</tr>
<tr>
<td>5.17</td>
<td>Adversaries $C$ and $D$ in the proof of Theorem 18</td>
</tr>
<tr>
<td>5.18</td>
<td>Games $G_1$–$G_4$ in the proof of Theorem 26</td>
</tr>
<tr>
<td>5.19</td>
<td>Distribution X (left) and adversary $B$ (right) in the proof of Theorem 26</td>
</tr>
<tr>
<td>5.20</td>
<td>Adversary $C_i$ in the proof of Theorem 26</td>
</tr>
<tr>
<td>5.21</td>
<td>Adversaries $A_H, A_G$ in the proof of Theorem 27</td>
</tr>
<tr>
<td>5.22</td>
<td>PA0 extractor $\text{Ext}$ in the proof of Theorem 27</td>
</tr>
<tr>
<td>5.23</td>
<td>CR adversary $B_G$ in the proof of Theorem 27</td>
</tr>
<tr>
<td>5.24</td>
<td>CR adversary $B_H$ in the proof of Theorem 27</td>
</tr>
<tr>
<td>5.25</td>
<td>EXT1 adversary $A_{PAD}$ in the proof of Theorem 28</td>
</tr>
<tr>
<td>5.26</td>
<td>PA1 extractor $\text{Ext}$ in the proof of Theorem 28</td>
</tr>
<tr>
<td>5.27</td>
<td>NCR adversary $B$ in the proof of Theorem 28</td>
</tr>
<tr>
<td>5.28</td>
<td>Games $G_1$–$G_3$ in the proof of Theorem 29</td>
</tr>
<tr>
<td>5.29</td>
<td>Adversary $B$ in the proof of Theorem 29</td>
</tr>
<tr>
<td>5.30</td>
<td>Adversary $C$ in the proof of Theorem 29</td>
</tr>
<tr>
<td>5.31</td>
<td>EXT1 adversary $A_{PAD}$ in the proof of Theorem 31</td>
</tr>
<tr>
<td>5.32</td>
<td>PA1 extractor $\text{Ext}$ in the proof of Theorem 31</td>
</tr>
<tr>
<td>5.33</td>
<td>NCR adversary $B$ in the proof of Theorem 31</td>
</tr>
<tr>
<td>5.34</td>
<td>EXT2 adversary $A_G$ in the proof of Theorem 32</td>
</tr>
<tr>
<td>5.35</td>
<td>PA2 extractor $\text{Ext}$ in the proof of Theorem 32</td>
</tr>
<tr>
<td>5.36</td>
<td>NCR adversary $B_G$ in the proof of Theorem 32</td>
</tr>
<tr>
<td>5.37</td>
<td>CR adversary $B_H$ in the proof of Theorem 32</td>
</tr>
<tr>
<td>5.38</td>
<td>XOR-NM1 adversary $B_F$ in the proof of Theorem 32</td>
</tr>
<tr>
<td>5.39</td>
<td>The hash function family $\mathcal{G}$</td>
</tr>
<tr>
<td>5.40</td>
<td>Games $G_1$–$G_4$ in the proof of Theorem 34</td>
</tr>
<tr>
<td>5.41</td>
<td>Games $G_5$–$G_8$ in the proof of Theorem 34</td>
</tr>
<tr>
<td>5.42</td>
<td>Adversary $D_1$ in the proof of Theorem 34</td>
</tr>
<tr>
<td>5.43</td>
<td>Adversary $B$ and distribution $D$ in the proof of Theorem 34</td>
</tr>
<tr>
<td>5.44</td>
<td>Circuit sample $\text{Samp}$ (left) and adversary $D_7$ (right) in the proof of Theorem 34</td>
</tr>
<tr>
<td>5.45</td>
<td>Distribution $D$ (left) and adversary $A$ (right) in the proof of Lemma 35</td>
</tr>
<tr>
<td>5.46</td>
<td>Instantiability results for RSA-OAEP sorted by scheme variant, where $n$ is modulus length, $k$ is the security parameter, and $\mu$ is message length. Typically $n = 2048$, $k = 128$, and $\mu = 128$</td>
</tr>
<tr>
<td>6.1</td>
<td>Our new transform $\hat{\text{FO}}_{\text{F}, \text{H}, \mathcal{G}}[\text{PKE}, \text{SE}] = (\bar{\text{FO}}, \text{Dec}, \bar{\text{FO}}, \text{Enc}, \bar{\text{FO}}, \text{Dec})$</td>
</tr>
<tr>
<td>6.2</td>
<td>Games $G_1$–$G_3$ in the proof of Theorem 36</td>
</tr>
<tr>
<td>6.3</td>
<td>EXT2 adversary $B$ in the proof of Theorem 36</td>
</tr>
<tr>
<td>6.4</td>
<td>Adversary $C$ in the proof of Theorem 36</td>
</tr>
</tbody>
</table>
6.5 Modified FO transform $\mathcal{F}_H, G[PKE, SE] = (\mathcal{F}_H.\text{Dec}, \mathcal{F}_H.\text{Enc}, \mathcal{F}_H.\text{Dec})$.

6.6 The hash function families $\mathcal{H}$ (left) and $\mathcal{G}$ (right).

6.7 Games $G_1 - G_4$ in the proof of Theorem 37.

6.8 Games $G_5 - G_8$ in the proof of Theorem 37.

6.9 Games $G_9 - G_{12}$ in the proof of Theorem 37.

6.10 PRG adversary $D_1$ (left) and iO adversary $D_2$ (right) in the proof of Theorem 37.

6.11 Adversary $D_3$ in the proof of Theorem 37.

6.12 Adversary $D_5$ in the proof of Theorem 37.

6.13 Adversary $D_7$ in the proof of Theorem 37.

6.14 Adversary $B_i$ in the proof of Theorem 37.

6.15 Adversary $D_{11}$ and circuit sample $\text{Samp}$ in the proof of Theorem 37.

6.16 Sampler $D$ (left) and adversary $A$ (right) in the proof of Lemma 38.

7.1 Games to define the D-SO-CPA security.

7.2 D-PKE scheme $\text{DE[LE, LT, H]}$.

7.3 Our Paillier-based all-but-$N$ lossy encryption scheme.

7.4 D-PKE scheme $\text{DE[LE, NLE, LT, NLT, H, TCR}_1, TCR}_2$.

7.5 D-PKE scheme $\text{DE[H, G, LT]}$.

7.6 Games $G_0, G_1$ of the proof of Theorem 44.

B.1 Games $G_0$ and $G_1$ of the proof of Theorem 40 and Theorem 41.

B.2 Games $G_2 - G_5$ of the proof of Theorem 40 and Theorem 41.

B.3 Games $G_0 - G_2$ of the proof of Theorem 42.

B.4 Games $G_3 - G_5$ of the proof of Theorem 42.

B.5 Games $G_6, G_7$ of the proof of Theorem 42.

B.6 Games $G_8, G_9$ of the proof of Theorem 42.
1.1 Background and Motivation

Despite recent advances in cryptography, security analyses of encryption schemes fall short of ruling out some possible attacks. Here we study two such types of attacks: selective-opening attacks (SOA) and attacks making use of the code of hash functions employed by the protocol rather than treating them as “black-box.”

SOA refers to a notion where the adversary sees a collection of ciphertexts and is able to “corrupt” ciphertexts of its choice, meaning reveal the underlying messages. For example, consider a set of users sending encrypted messages to Alice. The adversary sees these encrypted messages and is able to open a subset of them by breaking into users’ machines and learn the message. The question is what one can say about security of the other encrypted messages.

The study of SOA was initiated by Bellare, Hofheinz, and Yilek [13] in the context of randomized encryption schemes. To recall there are two types of encryption schemes, namely randomized and deterministic which refers to the encryption algorithm being randomized or deterministic. In the randomized encryption schemes a message $m$ can be encrypted to many different ciphertexts depending on the randomness (coin) that was used in the encryption. However, in the deterministic encryption schemes there is a unique ciphertext for each message $m$. To meet stronger security notions, having a randomized encryption is necessary.
At first, in the randomized public key encryptions setting, one may think that security of the unopened ciphertexts follows, if the encryption scheme meets some standard security notion, like indistinguishability under chosen-plaintext (IND-CPA). To recall IND-CPA security notion asks that the ciphertexts of any two messages $m_0, m_1$ be indistinguishable from each other. This is true if the attacker does not learn about the coins that was used in encryption. However, it has been shown that this is not true in general. The difficulty on achieving security against selective opening attacks lies on the exposure of the underlying coins used in encryption. In practice, these coins may be stored in the cache or the hard drive of machine and attacker that is breaking into users’ machines can learn about them. There are several works that studies security against selective-opening attacks (SOA) [13, 14, 24, 64, 67, 69] for randomized public key encryption in the coin-revealing setting (where attacker learns about messages as well as underlying coins used in encryption).

In the deterministic public key encryptions setting, there are no coins and one may think that security of the unopened ciphertexts should be easy to achieve as for the randomized public key encryptions when the adversary does not learn the coin used in the encryption. However, it has been shown that this is not true. The study of deterministic encryption (D-PKE), initiated by Bellare, Boldyreva, and O’Neill (BBO) [10] has proven to be impactful in both theory and practice. In particular, D-PKE has applications to fast search on encrypted outsourced databases [10], D-PKE can be extended to a notion of “hedged encryption” [12], which is a type of randomized encryption (R-PKE) that provides the best-possible security in the face of bad randomness, and D-PKE inspired a new security notion for hash functions used to instantiate random oracles [15].

Recently, Bellare, Dowsey, and Keelveedhi (BDK) [17] made important progress by demonstrating that requiring encryption to be deterministic can impact security in
several subtle ways. In particular, they show that a certain “simulation-based” notion of selective-opening security (SOA) is impossible to achieve in the case of D-PKE. Under this form of selective-opening attack\(^1\), a recipient receives (possibly related) messages from multiple senders encrypted under the recipient’s public key, and an adversary can corrupt some senders to recover messages underlying ciphertexts of its choice. The notion demands that for any adversary who sees messages underlying ciphertexts of its choice, there exists an (efficient) simulator who does not see these messages but such that the probability that the adversary outputs some information about the unopened messages is about the same as the simulator.

SOA security has been well-established as important in the setting of R-PKE, where it is known to be achievable [13, 64]. From a practical perspective, SOA seems especially compelling in the D-PKE setting for the following reason. It is plausible that a sender’s machine maintains copies of sent messages. Therefore, if a break-in occurs, the adversary would recover these messages, leading to an SOA attack. Note there are no coins here which could be erased by the sender’s machine to prevent this attack, as in the R-PKE setting.\(^2\) In this light, impossibility of SOA for D-PKE indeed seems like a serious drawback. Given the desire for positive results on D-PKE in the SOA setting, the starting point of our work is to ask whether there is an alternative meaningful formulation of SOA security on D-PKE that is achievable.

We next explore attacks making use of the code of hash functions rather than treating them as “black-boxes.” The proof of security for encryption schemes is a difficult task. To help us with this task, we usually study the security of encryption schemes in idealized models. These ideal models are not true in the real-world but

---

\(^1\)We clarify that there are actually two forms of SOA security, called coin-revealing and key-revealing [13]. This dissertation concerns coin-revealing.

\(^2\)In the R-PKE setting, if the adversary recovers only the messages but not the coins, standard IND-CPA security suffices [13].
they abstracts away some details of a real-world system. We refer to the attacks in the ideal model as “black-box” attack due to the abstraction of some details of real-world system. The random oracle (RO) is an idealize model that models a hash function as a truly random function. This model was first introduced by Bellare and Rogaway [5] and abstracts away details of hash functions.

Hash functions are very important building blocks of practical schemes used on Internet. A common approach for designing practical schemes is to first design a scheme in the RO model, and prove the security of this ideal scheme. Next, “instantiates” the oracles, that is, replace the truly random functions by a “suitable cryptographic hashing functions” (such as MD5 or SHA), making the code of hash function publicly-available to everyone (including the adversary). Thus, there are many possible “instantiations” of scheme, depending on the choice of the latter. To obtain a practical instantiation, it was suggested by [5] to build these functions from cryptographic hashing in an appropriate way. We call this the canonical instantiation. The RO model thesis of [5] is that if a scheme is secure in the RO model then its canonical instantiation remains secure in the standard (RO devoid) sense.

As the scheme that is formally analyzed differs from any of its instantiations, in particular the canonical instantiation, the security of canonical instantiation is unclear. One can indeed make claim for the security of the ideal system, but it is not clear what happens when one replaces the random oracle by a specific hash function. However, note that a security model always abstracts away some details of a real-world system. For example, the standard model still abstracts away side-effects of physical computation [79]. In particular, a security proof in the RO model guarantees absence of attacks treating the functions that instantiate the oracles as black-boxes, which is a natural form of cryptanalysis. Thus, the RO model thesis amounts to saying there
will also be no attacks on the canonical instantiation taking advantage of the code of these functions.

Unfortunately, the RO model thesis has been refuted in a strong sense, starting with the work of Canetti et al. [41]. These works show that there exist RO model schemes for which any instantiation, let alone the canonical one, yields a scheme that can be broken efficiently in the standard model. Given the widespread use of RO model schemes that have been standardized, the starting point of our work is to ask whether these schemes could withstand attacks that make non-blackbox use of hash functions.

1.2 Our Goals and Approach

Our goal here is to study and address these two type of attacks on encryption schemes. To address the selective opening attacks on deterministic primitives we start by giving a new comparison-based semantic-security style definition of SOA security for D-PKE, which we call D-SO-CPA, namely one that asks that no partial information about the plaintexts of unopened ciphertexts is leaked by encryption, while taking into account that the adversary sees the opened plaintexts. Intuitively, D-SO-CPA does not require the existence of a simulator but rather asks that the probability that the adversary outputs information about the unopened messages is about the same as for messages that are resampled conditioned on the opened ones. Such a definition is similar in style to the original security definition for D-PKE proposed by BBO. Note that for D-SO-CPA to be achievable, we need to require that the conditional resampling is efficient. A similar requirement was used by [13] in their indistinguishability-based definition of SOA security (IND-SO-CPA) for R-PKE encryption. We view this requirement as justified in light of the fact that any notion of SOA security for D-PKE without
it seems unachievable, and having positive results subject to this requirement is far better than having none at all.

Given that D-SO-CPA does not require the existence of a simulator, the BDK impossibility result does not apply. We next turn to the question of whether it is in fact achievable. Note that if the adversary can open $d$ messages, for any meaningful privacy we need that the distribution of messages to be what we call $(\mu,d)$-entropic for sufficiently large $\mu$, meaning that every message has entropy at least $\mu$ conditioned on fixed values of any $d$ others. We will refer to these parameters below. We stress that our definition and constructions allow $d$ to be $\infty$, that is, on proper message distributions, the adversary can open as many messages as it wants.

We next address the attacks that make non-blackbox use of hash functions. In particular, we are concerned with transforms that output a (public-key) encryption scheme, namely the OAEP trapdoor-permutation-based transform [6] and the Fujasaki-Okamoto (FO) hybrid-encryption transform [55]. Accordingly, we recall a bit about how these transforms work and what is known about them. OAEP takes a trapdoor permutation (TDP) $F$ and produces a public-key encryption scheme whose public key is an instance $f$ of the TDP. It uses two hash functions $G,H$ and the encryption algorithm has the form

$$E^{\text{OAEP}}_f(m;r) = f(s||t) \quad \text{where} \quad s = G(r)\oplus m\|0^c \quad \text{and} \quad t = H(s)\oplus r .$$

FO takes a public-key encryption scheme and a symmetric-key encryption scheme, and produces a new public-key encryption scheme. The encryption algorithm has the form

$$E^{\text{hy}}_{pk}(m;r) = E^{\text{asy}}_{pk}(r;H(r))\|E^{\text{asy}}_{K}(m) \quad \text{where} \quad K = G(r) .$$

Accordingly, the main question we study is, do there exist standard model hash functions that suffice to instantiate OAEP and FO (under IND-CCA) for classes of
“practical” base schemes? We ultimately seek plausible standard model properties of $G$, and $H$ that suffice to prove IND-CCA or similar in the standard model (which we just refer to “security” below). To recall IND-CCA security notion asks that the ciphertexts of any two messages $m_0, m_1$ be indistinguishable from each other to the adversary, even if the adversary can ask for the decryption of ciphertexts of its choice. We outline several ways in which we make progress towards it, these ways having been initiated by prior work (see Section 1.4).

One way is to show “partial instantiations” that use a plausible standard model property for one of $G$ or $H$, while still modeling the other as a RO. One may wonder what the point of this is, as security of the scheme is still proven in the RO model. We argue that the RO model is more nuanced, and viewing a scheme as either proven secure in the RO model or not is selling the scientific value of the model short. Indeed, ROs are used in different ways in a scheme, and instantiating one them isolates a property it relies on. In particular, suppose one has partial instantiation results for each of the ROs, as we show for OAEP transform. Then an attacker would need to exploit weakness in the interaction between these functions in order to break the scheme in standard model. In our eyes this makes an attack much less plausible.

Another way is to prove standard model security of variants of the scheme that fall “under the same framework.” Again, one may wonder what the point of this, as the schemes differ. We have a couple answers to this. One is that it can be seen as validating the framework more than simply proving the original scheme secure in the RO model. Another upshot is that it can lead to new versions of the scheme that may offer better security (in that they are both secure in the RO model and under plausible assumptions in the standard model). In fact, our results for one of our variants, namely $s$-clear RSA-OAEP, leads to the most efficient IND-CCA secure scheme in the standard model under arguably plausible (though rather bold) assumptions. This
is of theoretical interest and well as practical interest. Finally, one can try to reduce instantiating the original scheme to instantiating one of the variants, following e.g. [1]. We leave an investigation of this matter for future work.

1.3 RESULTS

We divide our results into two categories regarding each type of attacks that we consider on encryption schemes. We start by giving our results on selective opening security attacks. Then we discuss our results on attacks that make non-blackbox use of hash functions. We focus on two specific schemes, namely the OAEP transform and Fujasaki-Okamoto (FO) transform [55] and show how to make it secure against attacks that make non-blackbox use of underlying hash functions used in these transforms.

1.3.1 Results on Security Against Selective-Opening Attacks

A new definition. Our first contribution is a new comparison-based semantic-security style definition of SOA security for D-PKE, which we call D-SO-CPA. We do not require the existence of a simulator, therefore the BDK impossibility result does not apply. Intuitively, D-SO-CPA does not require the existence of a simulator but rather asks that the probability that the adversary outputs information about the unopened messages is about the same as for messages that are resampled conditioned on the opened ones.

Constructions in the standard model. Turning to constructions in the standard mode, we give a scheme based on the “Encrypt-with-Hardcore” (EwHC) construction of D-PKE due to Fuller et al. [58]. Recall in EwHC one deterministically encrypts a message $x$ by encrypting $f(x)$ under an R-PKE scheme using $h(x)$ as the coins, where $f$ is a TDF and $h$ is a hardcore function for $f$. In our scheme, the TDF
is required to be lossy [84] and the R-PKE scheme is required to be “perfectly lossy,” which is a strengthening of the notion of lossy encryption due to [13] requiring that ciphertexts lose all information about messages in the lossy mode. Intuitively, lossy trapdoor functions have a description that is indistinguishable from that of a function that loses information about its input (i.e., has a bounded range).

As above, our security proof first uses an observation from [13] that switching both the TDF and R-PKE scheme to the lossy modes can be done in the SOA setting, and the remainder of the proof is information-theoretic. Unfortunately, in the D-PKE context there does not seem to be any way of “opening” a ciphertext to an arbitrary message as was done by [13] in the R-PKE context. (Indeed, this is the intuition behind the BDK impossibility result.) Therefore, we “guess” the subset of ciphertexts opened by the adversary ahead of time and then show how to conclude via a variant of the Leftover Hash Lemma due to [58]. Notably, we pay a cost for this only in the information-theoretic part of the proof, in terms of the required entropy of the messages, and are still able to rely on standard polynomial-hardness assumptions. However, this cost means we are only able to handle a bounded number of messages, regardless of \(d\). Bounded-message security for D-PKE was previous considered by [58] without SOA, for the case of arbitrarily correlated messages (so in fact our result is more general, since we handle \((\mu, d)\)-entropic, efficiently resamplable distributions for any \(d\)). However, without SOA it is known how to handle an unbounded number of messages assuming a sufficient independence [11, 27]. Bounded-message security notwithstanding, we show our scheme admits efficient instantiations using the Paillier-based lossy TDF from [27, 54] combined with the Paillier-based lossy encryption scheme from [13].

We then extend our results for an unbounded number of “\(t\)-correlated” messages, meaning each set of up to \(t\) messages may be arbitrarily correlated. We consider the
notion of \( t \)-correlated messages to be interesting in its own right, and it captures a setting with password hashing where a password is correlated with a small number of others (and it is even stronger than that, in that a password may be correlated with any small number of others). Our construction uses \( 2t \)-wise independent hash functions and regular lossy trapdoor function [84], which has practical instantiations, e.g., RSA is regular lossy [76]. A close variant of our scheme is shown to be D-SO-CPA secure in the NPROM [65].

Security against chosen-ciphertext attack. After developing our basic schemes, we extend our treatment of SOA for D-PKE to the setting of chosen-ciphertext security, a notion we call D-SO-CCA. (In the R-PKE setting, CCA security under SOA has been the subject of multiple works, including [52, 63, 66].) As usual, the notion gives the adversary access to a decryption oracle, which it is not allowed to query on its given ciphertexts. To achieve D-SO-CCA in the standard model, we adapt an approach of [27, 63, 84] and augment our basic scheme in this setting using an “all-but-\( N \)” lossy trapdoor function [63, 84] and a new notion of an “all-but-\( N \)” lossy encryption scheme. Again, we show efficient Paillier-based instantiations.

1.3.2 Results on Optimal Asymmetric Encryption Padding

A common thread running through our analyses is the use of plaintext awareness (PA) [4, 6, 9]. PA captures the intuition that an adversary who produces a ciphertext must “know” the corresponding plaintext. It is not itself a notion of privacy, but, at a high level, combined with IND-CPA it implies IND-CCA. We use this approach to obtain modularity in proofs, isolate assumptions needed, and make overall analyses more tractable. Moreover, while it seems that PA necessitates using knowledge
assumptions, this is somewhat inherent anyway due to black-box impossibility results discussed below.

PA comes in various flavors: PA-RO [9], and PA0, PA1, and PA2 [4]. PA-RO refers to a notion in the RO model, while PA0, PA1, and PA2 refer to standard model notions that differ in what extent the adversary can query its decryption or encryption oracles. (In particular, in PA2 the adversary can query for encryptions of unknown plaintexts.) Similarly, IND-CCA comes in flavors [9, 86]: IND-CCA0, IND-CCA1, and IND-CCA2. We use that [4, 9] show that IND-CPA + PA-RO implies IND-CCA2 in the RO model, IND-CPA + PA0 implies IND-CCA1 with one decryption query, IND-CPA + PA1 implies IND-CCA1, and IND-CPA + PA2 implies IND-CCA2.

Partial Instantiation Results. We first give partial instantiation results of OAEP transform under IND-CCA2. Such results have been sought after in prior work [25, 26, 37] but have proven negative results or settled for weaker security notions. The heroes for us here are new generalizations of the notions of “second-input extractability” (SIE) and “common-input extractability” (CIE) proven by Barthe et al. [3] to hold for small-exponent RSA ($e = 3$). SIE says that a TDP image point can be inverted given a sufficiently-long (depending on $e$) part of the preimage, whereas CIE says that two TDP images can be inverted if the preimages share a common part. They were used by [3] where the “part” is the least-significant bits to analyze a no-redundancy, one-round version of RSA-OAEP in the RO model. The assumptions are proven via Coppersmith’s algorithm for finding small roots of a univariate polynomial modulo $N$ [44].

We show that generalized versions where the “part” refers to some of the middle or most-significant bits, rather than least-significant bits, is useful for analyzing RSA-OAEP more generally. We show these versions also hold for small-exponent RSA,
but based on the *bivariate* Coppersmith algorithm [23, 44, 46]. Moreover, despite
the similarity of assumptions, our proof strategies in the partial instantiations are
somewhat different than that of Barthe et al. [3]. Another interesting point is that
while (generalized) SIE and CIE hold for $e = 3$, we argue they have practical value
for larger $e$ as well. Namely, while $e > 3$ would require an impractical “part” length
using Coppersmith’s technique, they could possibly hold for practical parameters via
other (in particular, non-blackbox) techniques. At least, we do not see how to refute
that, which could lend insight into why there is no IND-CCA2 attack on the scheme
for general $e$.  

**Results and intuition.** We show partial instantiations of both oracles $G, H$ under
very mild assumptions on the round functions — roughly, that $G$ is a pseudorandom
generator and $H$ is a hardcore function for TDP, respectively — in both cases
assuming TDP is SIE and CIE. We first prove IND-CPA security in these cases.
Interestingly, the instantiation of $G$ under IND-CPA uses that TDP is SIE while the
instantiation of $H$ does not, the intuition being that in the latter case we assume $H$
is a hardcore function so its output masks $r \in \{0, 1\}^\rho$ used in the challenge ciphertext
unconditionally. Now for PA-RO, in both cases we use SIE and CIE, but wrt. different
bits of the input. In the case of instantiating $G$, it is wrt. the redundancy bits $s_2$.
Intuitively, for a decryption query there are two cases. Firstly, that it has a *different*
$r$-part than the challenge and therefore this must have been queried to the RO, in
which case the SIE extractor works. Secondly, that it has the *same* $r$-part as the
challenge, but it therefore shares $s_2$, in which case the CIE extractor works. In the
case of instantiating $H$, there are again two cases for an encryption query depending

---

3Moreover, we conjecture this is different from the case of “lossiness” [76, 84] as shown for
RSA and used to analyze IND-CPA security of RSA-OAEP in [76]. Namely, to get sufficient
lossiness it seems to inherently require large $e$, since the *only* way to make RSA parameters
lossy is to have $e \mid \phi(N)$. 

12
on whether it shares the same s-part of the challenge or not; thus the assumption is wrt. the whole s-part.

**Full instantiation results on variants.** We next give full instantiation results for two variants of OAEP transform, called t-clear and s-clear OAEP transform. Prior results on t-clear OAEP transform [26] showed only partial instantiations or relatively weak security notions, and s-clear OAEP transform was only considered indirectly by Shoup [93] for negative results. In t-clear OAEP transform, a message is encrypted as \( f(s_1) \parallel s_2 \parallel t \) where \( s_1 \parallel s_2 = G(r) \oplus (m \parallel 0) \) for randomness \( r \in \{0,1\}^\rho \) and message \( m \in \{0,1\}^\mu \), \( t = H(s_1 \parallel s_2) \oplus r \). Here we divide \( s \) into \( s_1 \parallel s_2 \), where \( s_2 \in \{0,1\}^\zeta \), so the name “t-clear” while consistent with prior work [26], is somewhat of a misnomer. On the other hand, in s-clear OAEP transform a message is encrypted as \( s \parallel f(t) \). One of the heroes for us here is a hierarchy of “extractability” notions we define and assume for the round functions, called EXT-RO, EXT0, EXT1, EXT2, roughly paralleling PA-RO, PA0, PA1, PA2 respectively, and significantly generalizing prior work [38, 39]. Besides this parallel, our generalizations consider adversaries that output only part of an image point or an image point along with part of a preimage. These are bold assumptions to make on (functions constructed out of) cryptographic hash functions, but, as discussed above, we believe studying their implications is justified. In the case of s-clear, another hero is a family of new “XOR-type” assumptions we introduce. Again, we view part of our contribution as putting forth novel assumptions that the research community can target for theoretical constructions or proofs in the future.

We make several remarks about our results, particularly how they avoid known impossibility results, before detailing them:

- Extractability is a non-blackbox assumption (saying for every adversary there exists a non-blackbox “extractor”) so we avoid the impossibility result of
Kiltz and Pietrzak [73].\textsuperscript{4} That is, the fact we use extractable hash functions (extractability being an intuitive property used in the original RO model proof) is somewhat unavoidable.

\textbullet{} While extractability of $\mathcal{H}$ would \textit{prima facie} be false, we use it only in a plausible way for a cryptographic hash function. Namely, the adversary also outputs \textit{part of the preimage}. Extractability assumptions we use on $\mathcal{G}$, even where the adversary outputs only part of an image point, remain plausible as it is an expanding function with a sparse range (usually constructed something like $\mathcal{G}(x) = (\mathcal{H}(0\|x)\|\mathcal{H}(1\|x),\ldots)$.

\textbullet{} For extractability we use only bounded key-independent auxiliary input (basically, the keys for the other functions in the scheme), so we avoid the impossibility result of Bitansky \textit{et al.} [21]. Moreover, the key-dependent auxiliary information is just one image query (at least in the proof of IND-CCA2).

\textbullet{} Our “XOR-type” assumptions avoid a negative result of Shoup [93], showing that there is in attack if the general trapdoor permutation is “XOR-malleable.”\textsuperscript{5}

\textbullet{} We typically use the various forms of extractability in combination with (at least) collision-resistance, so that the extractor returns the “right” preimage. The collision-resistant construction of [81] based on knowledge assumptions, albeit where the adversary outputs the entire image point, is on the lowest level of our hierarchy (EXT0); furthermore, it is not known to work when the

\textsuperscript{4}As acknowledged by the authors there was a bug in the proceedings version of this paper, but this has been fixed for the full version [74].

\textsuperscript{5}In more detail, note that for $s$-clear the “overall” TDP (including the part output in the clear) is not partial one-way [57] so their security proof does \textit{not} apply. In fact, Shoup [93] considers the scheme in his proof that RSA-OAEP is not IND-CCA2-secure for general one-way TDPs, exhibiting the above-mentioned attack.
adversary outputs part of the image point. Any theoretical constructions for higher levels (EXT1, EXT2) are similarly open. We hope these are targeted in future work.

Results and intuition for \( t \)-clear. Our results for \( t \)-clear OAEP transform are weaker than those for \( s \)-clear OAEP transform. First, for \( t \)-clear we prove IND-CPA for random, public key independent messages, under mild assumptions on the round functions, namely that \( \mathcal{H} \) is a hardcore function for TDP \( \mathcal{F} \) and \( \mathcal{G} \) is a pseudo-random generator. Intuitively, the high-entropy requirement come from the fact that the adversary attacking \( \mathcal{H} \) needs to know \( r \) to prepare its challenge ciphertext, so the randomness of the input to \( \mathcal{H} \) needs to come from \( m \). (We could avoid it using the stronger assumption of UCE as per the result of [15], which could be viewed as a hedge.) Furthermore, \( m \) needs to be public-key independent so as to not bias the output. Then we can prove PA0 based on forms of EXT0 for \( \mathcal{G} \) and \( \mathcal{H} \), the intuition being that the plaintext extractor first extracts from the part \( \mathcal{G}(r) \) that is left in clear by the redundancy to get \( r \) and then runs the extractor for \( \mathcal{H} \) on \( t \oplus r \) from which it can compute \( m \), with the above part of the preimage to get \( s \). Note that when running the extractor here and below we have to be careful that the constructed extractor uses the same coins as the starting one for consistency (otherwise we will not end up with the right extractor). We can also prove PA1, although we have to make an extractability directly on the padding scheme.\(^6\) Interestingly, even this approach does not work for PA2, which we leave completely open for \( t \)-clear (cf. Remark 25).

Results and intuition for \( s \)-clear. We find that \( s \)-clear is much more friendly to a full instantiation by making novel but plausible assumptions on TDP. One is XOR-nonmalleability (XOR-NM), saying that from \( \mathcal{F}(x) \) it is hard to find some \( \mathcal{F}(x') \) and

\(^6\)At a very high level, we can prove EXT0 of \( \mathcal{G}, \mathcal{H} \) implies EXT0 for the padding scheme, but we do not know how to do this for EXT1 because of an “extractor blow-up” problem.
Such that \( z = x \oplus x' \). Another is XOR-indistinguishability (XOR-IND), saying for random \( x \) and adversarially-chosen \( z \) one cannot tell \( F(x) \) from \( F(x \oplus z) \) given “hint” \( \mathcal{G}(x) \). In our results, \( \mathcal{G} \) is a PRG, which we show also implies \( \mathcal{G} \) is a HCF for \( F \). So, the notion can be viewed as an extension of the classical notion of HCF. In fact, we use XOR-IND just to show IND-CPA. The intuition is that allows breaking the dependency of \( s \) in the input to OAEP with the input to TDP. The proofs of PA0 and PA1 are very similar, and showcase one reason \( s \)-clear is much more friendly to a full instantiation, namely it heavily depends on the extractability of \( \mathcal{G} \). That is, if \( \mathcal{G} \) is suitably extractable, the plaintext extractor can simply recover \( r \) and then compute the plaintext as \( s \oplus \mathcal{G}(r) \). For PA2, one has to be careful as when the adversary makes an encryption query, the plaintext extractor should call the image oracle for \( \mathcal{G} \), where in addition to \( \mathcal{G}(x) \) for random \( x \) it receives the hint of TDP on \( x \). We show that if TDP is XOR-IND then this implies the adversary can get the whole ciphertext as a hint to simulate the encryption oracle. Then we also have the worry about the adversary querying “mauled” ciphertexts to the extract oracle. Intuitively, if the \( r \)-part is the same then it cannot run the extractor for \( \mathcal{G} \), but we show this violates XOR-NM of TDP. On the other hand, if the \( s \)-part is the same then we cannot break XOR-NM but this creates a collision for \( \mathcal{G} \).

**Full instantiation result.** We show new full instantiation under chosen-ciphertext attack (CCA) for the OAEP transform encryption scheme for an appropriate sub-class of TDPs. This helps explain why the scheme, which so far has only been shown to have such security in the random oracle (RO) model, has stood up to cryptanalysis despite the existence of “uninstantiable” RO model schemes.
We instantiate hash functions \(G\) and \(H\) via a new unified paradigm of obfuscating an extremely lossy function (ELF) introduced by Zhandry [97]. We combine this with prior paradigms Brzuska and Mittelbach [35] (using point function obfuscation in the proof). To explain ELFs, we first recall the notion of lossy function (LF). The key for a LF can be generated in one of two possible modes, the injective and lossy modes, where the first induces an injective function and the second induces a highly non-injective one. Further, keys be indistinguishable to any efficient adversary. Note that the image of the lossy function cannot be too low here, else there would be a trivial distinguisher. ELFs achieve much more lossiness by reversing the order of quantifiers. Namely, for an ELF, for every adversary there exists an (adversary-dependent) indistinguishable lossy key-generation mode. ELFs were constructed from exponentially-hard DDH, which is plausible in appropriate elliptic curve groups.

Results and intuition. Using ELFs to instantiate ROs is exactly why they were introduced. However, in prior work they were not obfuscated. To motivate our new approach, note that it seems that ELFs could be useful with the task of “answering decryption queries” in a proof of CCA security for an encryption scheme. Indeed, our strategy is to try all possible answers (there are only polynomially many) and see which one “works.” Yet there is a problem: the hash output used in the challenge ciphertext may no longer look random. To solve this problem, we wrap the ELF in a higher-level program that we obfuscate. This program outputs a programmed point on a special input (used in forming the challenge ciphertext), and otherwise evaluates the ELF.

In our results, \(G\) is an obfuscation of the circuit \(C = ELF(PRFK(\cdot))\) where PRF is a puncturable PRF and TDP is POW and extractable. The idea is to use AIPO to alter the key hard-coded.
the circuit $\mathcal{C}$ on input $r$ to output freshly random value $z$ instead of $\text{ELF}(\text{PRF}_K(r))$. We show that obfuscation of an alternative circuit is indistinguishable from the obfuscation of the original circuit, using differing input obfuscation given the auxiliary information of the differing point $r$. However, in order to do so the adversary attacking diO needs to simulate the decryption oracle for IND-CCA adversary. We do this by using the property of ELF and second input extractor for TDP. Considering the running time of the IND-CCA adversary, we switch to the proper extremely lossy mode of ELF function that is indistinguishable to the adversary. We note that once we are in lossy mode, we can run the extractor for TDP on all possible output of the extremely lossy function to answer the decryption queries. Now that $z$ is uniformly random we conclude that ciphertext $c$ looks uniformly random. We note that we use non-black box extractability rather than black box assumption on TDP due to the fact that the combination of black box extractability assumption and auxiliary information of the differing point $r$ reveals the point $r$ to the AIPO adversary. Thus, we switch to the non-black box assumption. We point out that we require extractability assumption on TDP to answer to the decryption queries.

1.3.3 Results on Fujisaki-Okamoto Transform

We show new instantiation results under chosen-ciphertext security for slightly modified Fujisaki-Okamoto transforms. We give two separate instantiations for two different slightly tweaked FO transforms. We note that the changes that we make to the FO transform are conservative. For our first instantiation, we make use of the extractable functions to instantiate $\mathcal{H}$ while we model $\mathcal{G}$ as one-wayness extractor. In our second instantiation result, we use an indistinguishably obfuscator, puncturable PRF $\text{PRF}$, as well as an extremely lossy function ELF. Additionally, in the proof, we use an auxiliary-input point-function obfuscator. At a high-level, to instantiate $\mathcal{H}$ we
use the composite function $\text{PRF}_K(\text{PRG}(\cdot))$ and to instantiate $G$ we use the composite function $\text{ELF}(\text{PRF}_{K'}(\cdot))$.

**Results and Intuition.** We instantiate hash functions $G$ and $H$ via the same new unified paradigm of as obfuscating an extremely lossy function (ELF). We combine this with prior paradigms Brzuska and Mittelbach [35] (using point function obfuscation in the proof). In our results, $H$ is an obfuscation of circuit $C_1 = \text{PRF}_K(\text{PRG}(\cdot))$ and $G$ is an obfuscation of circuit $C_2 = \text{ELF}(\text{PRF}_K(\cdot))$. First we show that $c_1^* = \text{Enc}(\text{PRG}(r^*); C_1(r^*))$ is indistinguishable from $c_1^* = \text{Enc}(\text{PRG}(r^*); y^*)$ for freshly chosen random $y^*$ to any PPT adversary. To do so, we use a technique similar to the one given in [90]. At a very high-level, the idea of the technique is to alter a program $C_1$ (which is to be obfuscated) by surgically removing a key element of the program (in a way that does not alter the functionality of the program), without which the adversary cannot distinguish between $C_1(r^*)$ and freshly chosen random $y^*$, given $\text{PRG}(r^*)$. We first argue that since $\text{PRG}$ is secure, the adversary cannot distinguish the original security game in which the challenge ciphertext was created as $c_1^* = \text{Enc}(\text{PRG}(r^*); \text{PRF}_K(\text{PRG}(r^*)))$, and a hybrid experiment where the challenge ciphertext is created with a freshly chosen random $x^*$ as $c_1^* = \text{Enc}(x^*; \text{PRF}_K(x^*))$. Note that the point $x^*$ is not functionally accessible by the circuit with high probability (for significantly expanding $\text{PRG}$ we have w.h.p that $x^* \notin \text{PRGRng}$). Thus we can puncture the PRF key $K$ on $x^*$ without effecting the functionality of circuit $C_1$.

Thus, indistinguishability obfuscation guarantees that an obfuscation of an alternative circuit that uses a punctured PRF key that carves out $x^*$ is indistinguishable from the obfuscation of the original circuit, because these two circuits are functionally equivalent. Now, due to the puncturing, the adversary simply does not have enough information to distinguish $c_1^* = \text{Enc}(x^*; \text{PRF}_K(x^*))$ from $c_1^* = \text{Enc}(x^*; y^*)$.  

19
Next we need to show that $\mathcal{E}_{K^*}(m)$ looks uniformly random to any PPT adversary given $c_1^*$, where $K^* = \mathcal{C}_2(x^*)$ and circuit $\mathcal{C}_2 = \text{ELF}(\text{PRF}_K(\cdot))$. To do so, we adapt a new approach by incorporating ELF to the technique in [35] that was used to instantiate UCEs. We use AIPO to alter the circuit $\mathcal{C}_2$ on input $x^*$ to output freshly random value $K^*$ instead of $\text{ELF}(\text{PRF}_K(x^*))$. We show that obfuscation of an alternative circuit is indistinguishable from the obfuscation of the original circuit, using differing input obfuscation given the auxiliary information $c_1^*$ of the differing point $x^*$. However, in order to do so adversary attacking diO need to simulate the decryption oracle for IND-CCA adversary. We do this by using the property of ELF.

Considering the running time of the IND-CCA adversary, we switch to the proper extremely lossy mode of ELF function that is indistinguishable to the adversary. We note that once we are in lossy mode we can answer decryption queries by going over all possible output of the extremely lossy function. Now that once $K^*$ is uniformly random we conclude that $c_2^*$ looks uniformly random using IND-CPA security of symmetric encryption $\text{SE}$.

### 1.4 Related and Follow-Up Work

Contrasting with our positive results on SOA, BDK’s definition is impossible to achieve in the non-programmable random oracle model (NPROM), even if the messages are uniform and independent. BDK’s attack to show this is however unsatisfying: the adversary outputs the ciphertexts and the public key as the “partial information” it learns about the unopened messages. But in the D-PKE setting, one can only hope to protect partial information that is independent of the public key, since the ciphertexts themselves are partial information on the messages [10]. This suggests that our definition is a better way to model SOA security of D-PKE.
An open question that remains is whether there is a standard-model D-PKE scheme that meets D-SO-CPA for an unbounded number of messages. The most desirable setting here would be arbitrarily correlated messages. This setting was solved by [15] without SOA, using their new notion of universal computational extractors (UCE). Unfortunately, we have not been able to make UCE work with SOA. On the other hand, even a standard-model D-PKE scheme in the SOA setting for an unbounded number of independent and uniform messages would be nice. We also mention that in the case of R-PKE, one typically considers SOA in a multi-user setting where there are many public keys. We have not done so in the case of D-PKE because for D-PKE security in the multi-user setting is already quite challenging to achieve even without SOA [17, 33], but this is a worthwhile direction for future work.

There have been a large number of works on both D-PKE and SOA (separately) in recent years. In the case of D-PKE, after the initial work of BBO came works on standard model constructions [11, 15, 27, 58, 95] as well as auxiliary-input security [33] and security for messages that depend on the public key in a limited way [88]. Other advanced security/functionality notions that have been recently considered for D-PKE include continual leakage resilience [77] and incrementality [80]. In the case of SOA for R-PKE, after the initial positive results of BHY, a few works considered chosen-ciphertext attacks [52, 63, 66]. Additionally, there is a great interest in showing whether standard security implies SOA security under various formulations, with works showing both positive and negative results [14, 64, 69]. Works have also studied relations between different formulations of security [24, 67].

Results about security of $F$-OAEP for an abstract TDP $F$ with applications RSA-OAEP in the RO model were shown in [6, 57, 93]. Ultimately, these works showed RSA-OAEP is IND-CCA2 secure in the RO model assuming only one-wayness of RSA, but with a loose security reduction. Interestingly, Shoup [93] considers $s$-clear
RSA-OAEP indirectly in a negative result about RSA-OAEP with a general one-way TDP. Security of $t$-clear RSA-OAEP (under the name “RSA-OAEP++”) has been analyzed in the RO model by Boldyreva, Imai, and Kobara [29], who show tight security in the multi-challenge setting.

Canetti [37] conjectured that his notion of perfectly one-wayness sufficed to instantiate one of the two oracles in $\mathcal{F}$-OAEP. This was disproved in general by Boldyreva and Fischlin [25], but their results do not contradict ours because they use a contrived TDP $\mathcal{F}$. Subsequently, Boldyreva and Fischlin [26] gave partial instantiations for $t$-clear $\mathcal{F}$-OAEP under stronger assumptions on the round functions.

Brown [34] and Paillier and Villar [82] showed negative results for proving RSA-OAEP is IND-CCA secure in restricted models, and Kiltz and Pietrzak [73] showed a general black-box impossibility results. As mentioned above, their results do not contradict ours because we use non-blackbox assumptions. Moving to weaker notions, Kiltz et al. [75] show IND-CPA security of RSA-OAEP using lossiness [84], while Bellare, Hoang, and Keelveedhi [15] show RSA-OAEP is IND-CPA secure for public-key independent messages assuming the round functions meet their notion of universal computational extraction. Boldyreva and Fischlin [26] show a weak form of non-malleability for $t$-clear $\mathcal{F}$-OAEP, again using very strong assumptions on the round functions. Lewko et al. [78] show IND-CPA security of the RSA PKCS v1.5 scheme, with the bounds later being corrected and improved by Smith and Zhang [94].

General notions for function families geared towards instantiating ROs that have been proposed include correlation intractability [41, 42], extractable hash functions [19, 21, 38, 39], perfect one-wayness [37, 40, 53], seed incompressibility [62], non-malleability [2, 28], and universal computational extraction (UCE) [15, 16, 36]. Again, note that the most general forms of several of these notions, namely correlation intractability and extractable hash functions have been shown to be (likely)
impossible. We avoid such general impossibility results in our work by focusing on a specific scheme.

1.5 Organization and Credits

The remaining chapters in this dissertation are organized as follows. In Chapter 2, we give the preliminaries. In Chapter 3, we give a new hierarchy of extractable functions. In Chapter 4, we abstract out some properties of the OAEP padding scheme we use. In Chapter 5, we formalize the algebraic properties of RSA we use and our partial instantiation results for RSA-OAEP. Then, we give novel “XOR-type” assumptions on RSA. Finally, we give our full instantiation results for OAEP transforms and its variants. In Chapter 6, we give slightly tweaked FO transforms and instantiate it in the standard model. In Chapter 7, we give our results on selective open security for deterministic encryption.

The material in Chapter 5 is based on joint work with Nairen Cao and Adam O’Neill [43]. It additionally draws on material from a manuscript of ours and is currently unpublished. The material in Chapter 6 is based on joint work with Adam O’Neill and is currently unpublished. The material in Chapter 7 is based on joint work with Viet Tung Hoang, Jonathan Katz and Adam O’Neill [65]. It additionally draws on material from a manuscript of ours [96] and is currently unpublished.
Chapter 2

Preliminaries

In this chapter, we overview notations and definitions we use throughout the thesis that are mostly from prior work.

2.1 Notation and Conventions

For a probabilistic algorithm $A$, by $y \leftarrow A(x)$ we mean that $A$ is executed on input $x$ and the output is assigned to $y$. We sometimes use $y \leftarrow A(x; r)$ to make $A$’s random coins explicit. If $A$ is deterministic we denote this instead by $y \leftarrow A(x)$. We denote by $\Pr[A(x) = y | x \leftarrow X]$ the probability that $A$ outputs $y$ on input $x$ when $x$ is sampled according to $X$. We denote by $[A(x)]$ the set of possible outputs of $A$ when run on input $x$. For a finite set $S$, we denote by $s \leftarrow S$ the choice of a uniformly random element from $S$ and assigning it to $s$.

The security parameter is denoted $k \in \mathbb{N}$. Unless otherwise specified, all algorithms must run in probabilistic polynomial-time (PPT) in $k$, and an algorithm’s running-time includes that of any overlying experiment as well as the size of its code. Integer parameters often implicitly depend on $k$. The length of a string $s$ is denoted $|s|$. We denote by $s|_i^j$ the $i$-th least significant bits (LSB) to $j$-th least significant bits of $s$ (including the $i$-th and $j$-th bits), where $1 \leq i \leq j \leq |s|$. For convenience, we denote by $s|_1^\ell = s|_1^\ell$ the $\ell$ least significant bits of $s$ and $s|_1^{\ell} = s|_1^{\ell}$ the $\ell$ most significant bits (MSB) of $s$, for $1 \leq \ell \leq |s|$.
Let $\mathbb{N}$ denote the set of all non-negative integers. For any $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$. Vectors are denoted in boldface, for example $\mathbf{x}$. If $\mathbf{x}$ is a vector then $|\mathbf{x}|$ denotes the number of components of $\mathbf{x}$ and $\mathbf{x}[i]$ denotes its $i$-th component, for $1 \leq i \leq |\mathbf{x}|$. For a vector $\mathbf{x}$ of length $n$ and any $I \subseteq [n]$, we denote by $\mathbf{x}[I]$ the vector of length $|I|$ such that $\mathbf{x}[I] = (\mathbf{x}[i])_{i \in I}$, and by $\mathbf{x}[\bar{I}]$ the vector of length $n - |I|$ such that $\mathbf{x}[\bar{I}] = (\mathbf{x}[i])_{i \notin I}$. For convenience, we extend algorithmic notation to operate on each vector of inputs component-wise. For example, if $A$ is an algorithm and $\mathbf{x}, \mathbf{y}$ are vectors then $\mathbf{z} \leftarrow A(\mathbf{x}, \mathbf{y})$ denotes that $\mathbf{z}[i] \leftarrow A(\mathbf{x}[i], \mathbf{y}[i])$ for all $1 \leq i \leq |\mathbf{x}|$.

2.2 Statistical Notions

We write $P_X$ for the distribution of random variable $X$ and $P_X(x)$ for the probability that $X$ puts on value $x$, i.e. $P_X(x) = \Pr[X = x]$. Similarly, we write $P_{X|E}$ for the probability distribution of $X$ conditioned on event $E$, and $P_{XY}$ for the joint distribution $(X, Y)$ of random variables $X, Y$. We denote by $U_\ell$ the uniform distribution on $\{0, 1\}^\ell$. We write $U_S$ for the uniform distribution on the set $S$. Let $X$ be a random variable taking values on a finite domain. The min-entropy of a random variable $X$ is $H_\infty(X) = -\log(\max_x \Pr[X = x])$. The average conditional min-entropy of $X$ given $Y$ is $\bar{H}_\infty(X|Y) = -\log(\sum_y P_Y(y) \max_x \Pr[X = x \mid Y = y])$.

2.2.1 Statistical Distance

Let $X, Y$ be random variables taking values on a common finite domain. The statistical distance between $X$ and $Y$ is given by $\Delta(X, Y) = \frac{1}{2} \sum_x |\Pr[X = x] - \Pr[Y = x]|$. We also define $\Delta(X, Y \mid S) = \frac{1}{2} \sum_{x \in S} |\Pr[X = x] - \Pr[Y = x]|$, for a set $S$.

Entropy after information leakage. Dodis et al. [50] characterized the effect of auxiliary information on average min-entropy in Lemma 1.
Lemma 1 [50] Let $X, Y, Z$ be random variables and $\delta > 0$ be a real number.

(a) If $Y$ has at most $2^\lambda$ possible values then we have $\tilde{H}_\infty(X \mid Z, Y) \geq \tilde{H}_\infty(X \mid Z) - \lambda$.

(b) Let $S$ be the set of values $b$ such that $H_\infty(X \mid Y = b) \geq \tilde{H}_\infty(X \mid Y) - \log(1/\delta)$. Then it holds that $\Pr[Y \in S] \geq 1 - \delta$.

Lemma 2 Let $X, Y$ be random variables where $\tilde{H}_\infty(X \mid Y) \geq \mu$. For any $0 \leq \delta < 1/2$, if the random variable $Y$ is a $\delta$-balanced boolean, then $H_\infty(X \mid Y = b) \geq \mu - \log(1/2 - \delta)$ for all $b \in \{0, 1\}$.

Proof. We know that $\Pr[Y = b] \geq 1/2 - \delta$, for all $b \in \{0, 1\}$. We also have that $\sum_b \Pr[Y = b] \max_x \Pr[X = x \mid Y = b] \leq 2^{-\mu}$. Therefore, we obtain that $\max_x \Pr[X = x \mid Y = b] \leq 2^{-\mu}/(1/2 - \delta)$ for all $b \in \{0, 1\}$. Summing up, we get $H_\infty(X \mid Y = b) \geq \mu - \log(1/2 - \delta)$ for all $b \in \{0, 1\}$. □

2.2.2 Unpredictable Distribution

A distribution ensemble $D = \{D_k = (Z_k, X_k)\}_{k \in \mathbb{N}}$, on pairs of strings, is unpredictable if for every PPT algorithm $A$, we have

$$\Pr[A(1^k, z) = x : (x, z) \leftarrow D_k]$$

is negligible in $k$.

2.3 Cryptographic Primitives

2.3.1 Private-Key Encryption and Its Security

Private-key encryption. A private-key encryption scheme $SE$ with message space $Msg$ is a tuple of algorithms $(K, E, D)$. The randomized algorithm $K$ on input $1^k$ outputs a private key $K$. The encryption algorithm $E$ on inputs $K$ and a message
\[
m \in \text{Msg}(1^k) \text{ outputs a ciphertext } c. \text{ The deterministic decryption algorithm } \mathcal{D} \text{ on inputs } K \text{ and ciphertext } c \text{ outputs a message } m \text{ or } \bot. \text{ We require that for all } K \in [\mathcal{K}(1^k)] \text{ and all } m \in \text{Msg}(1^k), \mathcal{D}_K((\mathcal{E}_K(m))) = m \text{ with probability 1.}
\]

**Security of private-key encryption.** Let \( SE = (\mathcal{K}, \mathcal{E}, \mathcal{D}) \) be a private key encryption scheme and \( A = (A_1, A_2) \) be an adversary. Let \( \mathcal{M} \) be a PPT algorithm that takes inputs \( 1^k \) to return a message \( m \in \text{Msg}(1^k) \). We associate the experiment in Figure 2.1 for every \( k \in \mathbb{N} \). Define the \textit{ind-cca advantage} of \( A \) against \( SE \) as

\[
\text{Adv}_{\text{SE}, A}^{\text{ind-cca}}(k) = 2 \cdot \Pr[\text{IND-CCA}_{SE}^A(k) \Rightarrow 1] - 1.
\]

We note that \( A_2 \) is not allowed to ask \( \mathcal{D} \) to decrypt \( c \). We say \( SE \) is \textit{secure under chosen-ciphertext attack} (IND-CCA) if \( \text{Adv}_{\text{SE}, A}^{\text{ind-cca}}(k) \) is negligible in \( k \) for all PPT \( A \).

**Integrity of private-key encryption.** Let \( SE = (\mathcal{K}, \mathcal{E}, \mathcal{D}) \) be a private key encryption scheme, and \( A \) be an adversary. We associate the experiment in Figure 2.1 for every \( k \in \mathbb{N} \). Define the \textit{int-ctxt advantage} of \( A \) against \( SE \) as

\[
\text{Adv}_{\text{SE}, A}^{\text{int-ctxt}}(k) = \Pr[\text{INT-CTXT}_{SE}^A(k) \Rightarrow 1].
\]

We say that \( SE \) is secure under INT-CTXT, if \( \text{Adv}_{\text{SE}, A}^{\text{int-ctxt}}(k) \) is negligible in \( k \) for all PPT \( A \).
Key binding of private-key encryption. Let \( SE = (K, E, D) \) be a private key encryption scheme. We say \( SE \) is key binding if for any \( K \in [K(1^k)] \), any \( m \in \text{Msg}(1^k) \), and randomness \( r \), there is no key \( K' \) such that \( K \neq K' \) and \( D_{K'}(E_K(m; r)) \neq \bot \).

Robustness of private-key encryption. Let \( SE = (K, E, D) \) be a private key encryption scheme. We say \( SE \) is robust if for any PPT adversary \( A \)

\[
\text{Adv}_{SE, A}^{\text{rob}}(k) = \Pr \left[ E_{K_1}(m_1) = E_{K_2}(m_2) : (K_1, K_2, m_1, m_2) \leftarrow A(1^k) \right],
\]

is negligible in \( k \).

Remark 3 We note that if the private-key encryption scheme \( SE = (K, E, D) \) is key-binding then \( SE \) is also robust.

2.3.2 Public-Key Encryption and Its Security

Public-key encryption. A public-key encryption scheme \( PKE \) with message space \( \text{Msg} \) is a tuple of algorithms \((\text{Kg}, \text{Enc}, \text{Dec})\). The key-generation algorithm \( \text{Kg} \) on input \( 1^k \) outputs a public key \( pk \) and matching secret key \( sk \). The encryption algorithm \( \text{Enc} \) on inputs \( pk \) and a message \( m \in \text{Msg}(1^k) \) outputs a ciphertext \( c \). The deterministic decryption algorithm \( \text{Dec} \) on inputs \( sk \) and ciphertext \( c \) outputs a message \( m \) or \( \bot \). We require that for all \( (pk, sk) \in [\text{Kg}(1^k)] \) and all \( m \in \text{Msg}(1^k) \), \( \text{Dec}(sk, (\text{Enc}(pk, m)) = m \) with probability 1.

Security of public-key encryption [60, 87]. Let \( PKE = (\text{Kg}, \text{Enc}, \text{Dec}) \) be a public key encryption scheme and \( A = (A_1, A_2) \) be an adversary. Let \( \mathcal{M} \) be a PPT algorithm that takes inputs \( 1^k \) and a public key \( pk \) to return a message \( m \in \text{Msg}(1^k) \). For \( \text{ATK} \in \{\text{CPA}, \text{CCA1}, \text{CCA2}\} \) we associate the experiment in Figure 2.2 for every \( k \in \mathbb{N} \). Define the ind-atk advantage of \( A \) against \( PKE \) as

\[
\text{Adv}_{PKE, A}^{\text{ind-atk}}(k) = 2 \cdot \Pr \left[ \text{IND-ATK}^A_{PKE}(k) \Rightarrow 1 \right] - 1.
\]
Game \( \text{IND-ATK}^A_{\text{PKE}}(k) \)

\begin{align*}
  & b \leftarrow \{0, 1\}; \quad (pk, sk) \leftarrow \text{Kg}(1^k) \\
  & (M_0, M_1, state) \leftarrow A^k_1(1^k, pk) \\
  & m_b \leftarrow M_b(1^k, pk) \\
  & c \leftarrow \text{Enc}(pk, m_b) \\
  & d \leftarrow A^k_2(1^k, pk, c, state) \\
  & \text{Return } (b = d)
\end{align*}

Figure 2.2: Game to define IND-ATK security.

If \( \text{atk} = \text{cpa} \), then \( O_1(\cdot) = \varepsilon \), and \( O_2(\cdot) = \varepsilon \). We say \( \text{PKE} \) is secure under chosen-plaintext attack (IND-CPA) if \( \text{Adv}^{\text{ind-}\text{cpa}}_{\text{PKE}, A}(k) \) is negligible in \( k \) for all PPT \( A \). Similarly, if \( \text{atk} = \text{cca}_1 \), then \( O_1(\cdot) = \text{Dec}(sk, \cdot) \), and \( O_2(\cdot) = \varepsilon \); if \( \text{atk} = \text{cca}_2 \), then \( O_1(\cdot) = \text{Dec}(sk, \cdot) \), and \( O_2(\cdot) = \text{Dec}(sk, \cdot) \). In the case of \( \text{cca}_2 \), \( A_2 \) is not allowed to ask \( O_2 \) to decrypt \( c \). We say that \( \text{PKE} \) is secure under non-adaptive chosen-ciphertext attack or IND-CCA1 (resp. adaptive chosen-ciphertext attack or IND-CCA2), if \( \text{Adv}^{\text{ind-cca}_1}_{\text{PKE}, A}(k) \) (resp. \( \text{Adv}^{\text{ind-cca}_2}_{\text{PKE}, A}(k) \)) is negligible in \( k \) for all PPT \( A \).

PA-RO security. We first define plaintext-awareness in the RO model following [8], which builds on the definition in [6]. Note that PA-RO combined with IND-CPA security is strictly stronger than IND-CCA2 security in general. Let \( \text{PKE} = (\text{Kg}, \text{Enc}, \text{Dec}) \) be a public key encryption scheme and let \( \mathcal{M} \) be a PPT algorithm that takes as inputs \( 1^k \) and a public key \( pk \), and outputs a message \( m \in \text{Msg}(1^k) \). To adversary \( A \) and extractor \( \text{Ext} \), we associate the experiment in Figure 2.3 for every \( k \in \mathbb{N} \). We say that \( \text{PKE} \) is PA-RO secure if for every PPT adversary \( A \) there exists an extractor \( \text{Ext} \) such that

\[
\text{Adv}^{\text{pa-}\text{ro}}_{\text{PKE}, A, \text{Ext}}(k) = 2 \cdot \Pr \left[ \text{PA-RO}^A_{\text{PKE}}(k) \Rightarrow 1 \right] - 1
\]

is negligible in \( k \).
**Game** PA-RO\(A,\text{Ext}_{PKE}^k\)

\[
\begin{align*}
b &\leftarrow \{0, 1\}; \quad i \leftarrow 1; \quad j \leftarrow 1 \\
(pk, sk) &\leftarrow Kg(1^k) \\
b' &\leftarrow A^{RO, 1, Enc(pk, \cdot), D(sk, \cdot)}(pk) \\
\text{Return } (b = b')
\end{align*}
\]

**Procedure** RO\((x, i)\)

\[
\begin{align*}
&\text{If } H[x] = \bot \text{ then } H[x] \leftarrow \{0, 1\}^\ell \\
&\text{If } i = 1 \text{ then } \\
&x[j] \leftarrow x; \quad h[j] \leftarrow H[x]; \quad j \leftarrow j + 1 \\
&\text{Return } H[x]
\end{align*}
\]

**Procedure** Enc\((pk, M)\)

\[
\begin{align*}
m &\leftarrow M(1^k, pk) \\
c &\leftarrow Enc^{RO, 2}(pk, m) \\
c[i] &\leftarrow c; \quad i \leftarrow i + 1 \\
\text{Return } c
\end{align*}
\]

**Procedure** D\((sk, c)\)

\[
\begin{align*}
&\text{If } c \in c \text{ then return } \bot \\
m_0 &\leftarrow \text{Dec}(sk, c) \\
m_1 &\leftarrow \text{Ext}^{RO, 3}(x, h, c, c, pk) \\
\text{Return } m_b
\end{align*}
\]

**Figure 2.3:** Game to define PA-RO security.

**Remark 4** Our definition of plaintext awareness in the random oracle model differs from the definition given in [8] in the following way. In our definition, we are giving the extractor access to the random oracle. We observe that the analogous result of [8, Theorem 4.2] that IND-CPA and PA-RO together imply IND-CCA2 still holds for our modified definition, since in the proof the IND-CPA adversary could query its own random oracle to answer to the random oracle queries of the extractor.

We now turn to definitions of plaintext awareness in the standard model, following [4].

**PA1 Security.** Let PKE = \((Kg, Enc, Dec)\) be a public key encryption scheme. To adversary \(A\) and extractor \(\text{Ext}\), we associate the experiment in Figure 2.4 for every \(k \in \mathbb{N}\). We say that PKE is PA1 secure if for every PPT adversary \(A\) with coin space \text{Coins} there exists an extractor \(\text{Ext}\) such that,

\[
\text{Adv}_{PKE, A, Ext}^{\text{PA1}}(k) = 2 \cdot \Pr \left[ \text{PA1}_{PKE}^{A, Ext}(k) \Rightarrow 1 \right] - 1,
\]

is negligible in \(k\).
The idea behind this security notion is that, given the randomness of the adversary \( A \) it should be easy to answer to the decryption queries of the adversary without knowing the secret key \( sk \). In other words, the adversary \( A \) should not be able to come up with a valid ciphertext \( c \) without knowing the underlying message \( m \).

### PA0 security

We define PA0 similarly to PA1, except \( A \) is only allowed to make a single oracle query. Let PA0 be the corresponding experiment, and define

\[ \text{Adv}_{\text{PA0}}^{\text{PKE}, A, \text{Ext}}(k) = 2 \cdot \Pr \left[ \text{PA0}_{\text{PKE}}^{A, \text{Ext}}(k) \Rightarrow 1 \right] - 1. \]

We say \( \text{PKE} \) is PA0 secure if for every PPT adversary \( A \) there exists an extractor \( \text{Ext} \) such that \( \text{Adv}_{\text{PA0}}^{\text{PKE}, A, \text{Ext}}(k) \) is negligible in \( k \).

### PA2 security

Let \( \text{PKE} = (\text{Kg, Enc, Dec}) \) be a public-key encryption scheme. To adversary \( A \) and extractor \( \text{Ext} \), we associate the experiment in Figure 2.5 for every \( k \in \mathbb{N} \). We say that \( \text{PKE} \) is PA2 secure if for every PPT adversary \( A \) there exists an extractor \( \text{Ext} \) such that,

\[ \text{Adv}_{\text{PA2}}^{\text{PKE}, A, \text{Ext}}(k) = 2 \cdot \Pr \left[ \text{PA2}_{\text{PKE}}^{A, \text{Ext}}(k) \Rightarrow 1 \right] - 1, \]

is negligible in \( k \).
Remark 5. Our PA2 definition comes from [4]. Other than PA1 adversary, we will give PA2 adversary extra access to encryption oracle. This models the ability IND-CCA2 adversary obtains ciphertext without knowing the randomness.

Randomness recovery [48]. Let PKE = (Kg, Enc, Dec) be a public key encryption. We say PKE is uniquely randomness recovering if there exist a PT randomness recovery algorithm Rec such that on input a secret key sk and ciphertext c outputs a randomness r. We require that for all (pk, sk) ∈ [Kg(1k)], all randomness r and all m ∈ Msg(1k), Rec(sk, (Enc(pk, m; r)) = r with probability 1.

2.3.3 Trapdoor Permutations and Their Security

Trapdoor permutations. A trapdoor permutation family with domain TDom is a tuple of algorithms $\mathcal{F} = (\text{Kg}, \text{Eval}, \text{Inv})$ that work as follows. Algorithm Kg on input a unary encoding of the security parameter $1^k$ outputs a pair $(f, f^{-1})$, where $f : \text{TDom}(k) \rightarrow \text{TDom}(k)$. Algorithm Eval on inputs a function $f$ and $x \in \text{TDom}(k)$ outputs $y \in \text{TDom}(k)$. We often write $f(x)$ instead of Eval$(f, x)$. Algorithm Inv on
inputs a function $f^{-1}$ and $y \in \text{TDom}(k)$ outputs $x \in \text{TDom}(k)$. We often write $f^{-1}(y)$ instead of $\text{inv}(f^{-1}, y)$. We require that for any $(f, f^{-1}) \in [\text{Kg}(1^k)]$ and any $x \in \text{TDom}(k)$, $f^{-1}(f(x)) = x$. We call $\mathcal{F}$ an $n$-bit trapdoor permutation family if $\text{TDom} = \{0, 1\}^n$. We will think of the RSA trapdoor permutation family [89] $n$-bit for simplicity, although its domain is $\mathbb{Z}^*_N$ for an $n$-bit integer $N$. Additionally, for convenience we define the following. For an $\nu$-bit trapdoor permutation family $\mathcal{F}$ and $\ell \in \mathbb{N}$, we define $\mathcal{F}|_{\ell} = (\text{Kg}|_{\ell}, \text{Eval}|_{\ell}, \text{Inv}|_{\ell})$ as the $(\nu + \ell)$-bit trapdoor permutation families such that for all $k \in \mathbb{N}$, all $(f|_{\ell}, f^{-1}|_{\ell}) \in [\text{Kg}|_{\ell}(1^k)]$, and all $x \in \{0, 1\}^{\nu+\ell}$, we have $f|_{\ell}(x) = f(x)^{n-\ell} \parallel x|_{\ell}$, and analogously for $\mathcal{F}|^{\ell}$.

**Lossiness.** A lossy trapdoor permutation [84] with domain $\text{LDom}$ is a tuple of algorithms $\text{LT} = (\text{IKg}, \text{LKg}, \text{Eval}, \text{Inv})$ that work as follows. Algorithm $\text{IKg}$ on input a unary encoding of the security parameter $1^k$ outputs an “injective” function $f$ and matching inverse function $f^{-1}$, where $f : \text{LDom}(k) \rightarrow \text{LDom}(k)$. Algorithm $\text{LKg}$ on input $1^k$ outputs a “lossy” function key $f$, where also $f : \text{LDom}(k) \rightarrow \text{LDom}(k)$. Algorithm $\text{Eval}$ on inputs an (either injective or lossy) function $f$ and $x \in \text{LDom}(k)$ outputs $y \in \text{LDom}(k)$. We often write $f(x)$ instead of $\text{Eval}(f, x)$. Algorithm $\text{Inv}$ on inputs a function $f^{-1}$ and a $y \in \text{LDom}(k)$ outputs $x \in \text{LDom}(k)$. We often write $f^{-1}(y)$ instead of $\text{inv}(f^{-1}, y)$. We denote by $\text{Img}(lk)$ the co-domain of $\text{Eval}(lk, \cdot)$. We require the following properties.

**Correctness:** For all $k \in \mathbb{N}$ and any $(f, f^{-1}) \in [\text{IKg}(1^k)]$, it holds that $f^{-1}(f(x)) = x$ for every $x \in \text{LDom}(k)$.

**Key-indistinguishability:** For security we require that for every distinguisher $D$, the advantage $\text{Adv}_{\text{LT}, D}(k) = \Pr[D(f) \Rightarrow 1 : (f, f^{-1}) \leftarrow \text{IKg}(1^k)] - \Pr[D(f) \Rightarrow 1 : f \leftarrow \text{LKg}(1^k)]$ is negligible in $k$. 

33
**Lossiness:** The size of the co-domain of $\text{Eval}(f, \cdot)$ is at most $|\text{LDom}(k)|/2^{\tau(k)}$ for all $k \in \mathbb{N}$ and all $f \in [\text{LKg}(1^k)]$. We call $\tau$ the *lossiness* of LT.

Both the RSA [76] and Rabin [91] trapdoor permutation families are lossy under suitable assumptions.

**Regularity.** Let LT be a lossy trapdoor permutation with domain LDom and lossiness $\tau$. We say LT is regular if for all $lk \in [\text{LKg}(1^k)]$ and all $y \in \text{Img}(lk)$, we have

$$\Pr[\text{Eval}(lk, U) = y] = 1/|\text{Img}(lk)|,$$

where $U$ is uniform on LDom($k$).

**One-wayness.** Let $\mathcal{F} = (\text{Kg}, \text{Eval}, \text{Inv})$ be a trapdoor permutation family with domain TDom. We say $\mathcal{F}$ is *one-way* if for every PPT inverter $I$:

$$\text{Adv}_{\mathcal{F},I}^{\text{owf}}(k) = \Pr_{(f,f^{-1}) \leftarrow \text{Kg}(1^k), x \leftarrow \text{TDom}(k)} \left[ x' \leftarrow I(f, f(x)) \mid x' = x \right],$$

is negligible in $k$.

**Partial one-wayness.** Let $\mathcal{F} = (\text{Kg}, \text{Eval}, \text{Inv})$ be a trapdoor permutation family with domain TDom. We say $\mathcal{F}$ is $\zeta$-*partial one way* if for every PPT inverter $I$:

$$\text{Adv}_{\mathcal{F},I}^{\text{pow}}(k) = \Pr_{(f,f^{-1}) \leftarrow \text{Kg}(1^k), x \leftarrow \text{TDom}(k)} \left[ x' \leftarrow I(f, f(x)) \mid x' = x \mid \zeta \right],$$

is negligible in $k$. It is shown in [56] that for RSA one-wayness implies partial one-wayness but the reduction is lossy.

**One-wayness with hint.** Let $\mathcal{F} = (\text{Kg}, \text{Eval}, \text{Inv})$ be a trapdoor permutation family with domain TDom. We say $\mathcal{F}$ is *one-way* with $\mu$-bits hint ($\mu$-OW) if for every PPT inverter $I$:

$$\text{Adv}_{\mathcal{F},I}^{\text{owf}}(k) = \Pr_{(f,f^{-1}) \leftarrow \text{Kg}(1^k), x \leftarrow \text{TDom}(k)} \left[ x' \leftarrow I(f, f(x), x^\mu) \mid x' = x \right],$$

is negligible in $k$. 

34
Non-malleability. Let $\mathcal{F} = (\text{Kg}, \text{Eval}, \text{Inv})$ be a trapdoor permutation family with domain $\text{TDom}$. We say $\mathcal{F}$ is $\mu$-bit non-malleable ($\mu$-NM) if for every PPT adversary $A$:

$$\text{Adv}_{\mathcal{F}, A}^{\mu\text{-nm}}(k) = \Pr_{(f, f^{-1}) \leftarrow \text{Kg}(1^k), x \leftarrow \text{TDom}(k)} \left[ f(x') \leftarrow A(f, f(x)) \mid x' |^\mu = x |^\mu \right] ,$$

is negligible in $k$.

### 2.3.4 Function Families and Associated Security Notions

**Function families.** A function family with domain $\text{F.Dom}$ and range $\text{F.Rng}$ is a tuple of algorithms $\mathcal{F} = (\mathcal{K}, F)$ that work as follows. Algorithm $\mathcal{K}$ on input a unary encoding of the security parameter $1^k$ outputs a key $K_F$. Deterministic algorithm $F$ on inputs $K_F$ and $x \in \text{F.Dom}(k)$ outputs $y \in \text{F.Rng}(k)$. We alternatively write $\mathcal{F}$ as a function $F : \mathcal{K} \times \text{F.Dom} \to \text{F.Rng}$. We call $\mathcal{F}$ an $\ell$-injective function if for all distinct $x_1, x_2 \in \text{F.Dom}(k)$ and $K_F \in [\mathcal{K}(1^k)]$, we have $F(K_F, x_1) |^\ell \neq F(K_F, x_2) |^\ell$.

**$t$-wise independent.** Let $\mathcal{H} : \mathcal{K} \times \text{HDom} \to \text{HRng}$ be a function family. We say that $\mathcal{H}$ is $t$-wise independent if for all $k \in \mathbb{N}$ and all distinct $x_1, \ldots, x_t \in \text{HDom}(k)$

$$\Delta \left( (h(K, x_1), \ldots, h(K, x_t)), (U_1, \ldots, U_t) \right) = 0 ,$$

where $K \leftarrow \text{HKg}(1^k)$ and $U_1, \ldots, U_t$ are uniform and independent in $\text{HRng}(k)$.

**Collision resistance.** Let $\mathcal{H} : \mathcal{K} \times \text{HDom} \to \text{HRng}$ be a function family. We say $\mathcal{H}$ is collision resistant (CR) if for any PPT adversary $A$:

$$\text{Adv}_{H, A}^{\text{cr}}(k) = \Pr_{K_H \leftarrow \mathcal{K}(1^k)} \left[ (x_1, x_2) \leftarrow A(K_H) \wedge \mathcal{H}(K_H, x_1) = \mathcal{H}(K_H, x_2) \mid x_1 \neq x_2 \right] ,$$

is negligible in $k$. Again, this is a standard notion that can be realized in a variety of ways, in particular it is one of the basic properties believed for cryptographic hash function.
Near-collision resistance. Let $\mathcal{H} : \mathcal{K}_H \times \text{HDom} \to \text{HRng}$ be a function family. For $\ell \in \mathbb{N}$, we say $\mathcal{H}$ is *near-collision resistant* with respect to $\ell$-least significant bits of the outputs (NCR$^\ell$) if for any PPT adversary $A$:

$$
\text{Adv}^\text{n-crf}_{H,A}(k) = \mathbb{P}_{\mathcal{K}_H \leftarrow \mathcal{K}_H(1^k)} \left[ (x_1, x_2) \leftarrow A(\mathcal{K}_H) \land \mathcal{H}(\mathcal{K}_H, x_1)|_{\ell} = \mathcal{H}(\mathcal{K}_H, x_2)|_{\ell} \mid x_1 \neq x_2 \right],
$$

is negligible in $k$.

We note that our definition differs slightly from [26] as both $x_1, x_2$ are adversarially chosen. In terms of feasibility, the same construction based on one-way permutations given in [26] works in our case as well. Similarly, we define NCR$^\ell$ where the adversary tries to find a collision on the $\ell$-most significant bits of the output.

Pseudorandom generators. Let $\mathcal{G} : \mathcal{K}_G \times \text{GDom} \to \text{GRng}$ be a function family. To adversary $A$, we associate the experiment in Figure 2.6 for every $k \in \mathbb{N}$. We say that $\mathcal{G}$ is a *pseudorandom generator* if for every PPT adversary $A$,

$$
\text{Adv}^\text{prg}_{G,A}(k) = 2 \cdot \mathbb{P}_{\mathcal{G}-\text{DIST}^A_G(k) \Rightarrow 1} - 1,
$$

is negligible in $k$.

This is a standard notion in theory and can be heuristically constructed from a cryptographic hash function in straightforward ways. We often consider keyless PRG functions.
Pseudorandom generators with image verifier. Let \( G : \mathcal{K}_G \times \text{GDom} \rightarrow \text{GRng} \) be a function family. Pseudorandom generators with \( \ell \)-bit image verifier are similar to pseudorandom generator except we will give adversary \( A \) with oracle access to \( \mathcal{V}_\ell \), where \( \mathcal{V}_\ell \) is an \( \ell \)-bit image verifier that on input \( y \) works as follows:

\[
\mathcal{V}_\ell(y) = \begin{cases} 
1 & \text{if } \exists x : y = G(K_G, x) \mid _\ell \\
0 & \text{otherwise}
\end{cases}
\]

Note that adversary \( A \) is not allowed to query for the challenge to the image verifier oracle. We say that \( G \) is a pseudorandom generator with \( \ell \)-bit image verifier (VPRG\(_\ell\)) if for every PPT adversary \( A \),

\[
\text{Adv}_{G,A}^{\text{VPRG}_\ell}(k) = 2 \cdot \Pr \left[ \text{VPRG-DIST}_G^A(k) \Rightarrow 1 \right] - 1,
\]

is negligible in \( k \).

In our results we do not require \( \mathcal{V}_\ell \) to be efficient, so they are (we believe) plausible for constructions based on cryptographic hash functions. It is weaker than the “adaptivity” assumption made in [83].

Hardcore functions. We define a notion of hardcore functions for non-uniform, correlated distributions as in [58], but we extend it to consider auxiliary input as well. Let \( \mathcal{F} = (\mathcal{K}_g, \mathcal{E}_g, \mathcal{I}_g) \) be a one-way trapdoor permutation family with domain \( \text{TDom} \). Let \( \mathcal{H} : \mathcal{K}_H \times \text{TDom} \rightarrow \text{HRng} \) be a function family. For \( k \in \mathbb{N} \), let \( X(k) \) be a distribution on input vector in \( \text{TDom}(k) \) and auxiliary information \( \alpha \in \{0,1\}^* \). To attacker \( A \) and distribution \( X(k) \), we associate the experiment in Figure 2.7 for every \( k \in \mathbb{N} \). We say that \( \mathcal{H} \) is a hardcore function for the trapdoor permutation family \( \mathcal{F} \) on a family of such distributions \( X \) if for every \( X(k) \in X(k) \) and for every PPT adversary \( A \),

\[
\text{Adv}_{\mathcal{F},H,X,A}^{\text{HCF}}(k) = 2 \cdot \Pr \left[ \text{HCF-DIST}_{\mathcal{F},H}^{A,X}(k) \Rightarrow 1 \right] - 1,
\]

is negligible in \( k \).
is negligible in $k$.

For messages drawn from a block-source, if $\mathcal{F}$ is sufficiently lossy in the sense of [85] then a universal hash function meets this notion. Additionally, a $2t$-wise independent function meets this notion for $t$ arbitrarily correlated, high-entropy messages if $\mathcal{F}$ loses a $1 - o(1)$ fraction of its input. It is an open problem to construct such a hardcore function for an unbounded number of arbitrarily correlated, high-entropy messages. However, we see it as plausible that a cryptographic hash function meets this definition.

**Partial hardcore functions.** For convenience, we also generalize the notion of hardcore function in the following way. Let $\mathcal{F} = (Kg, \text{Eval}, \text{Inv})$ be $n$-bit trapdoor permutation family. Let $\mathcal{H} : K_H \times \{0, 1\}^{n-\ell} \rightarrow HRng$ be a function family, for some $\ell < n$. To attacker $A$, we associate the experiment in Figure 2.8 for every $k \in \mathbb{N}$. We say that $\mathcal{H}$ is a $\ell$-partial hardcore function for the trapdoor permutation family $\mathcal{F}$ if for every PPT adversary $A$,

$$\text{Adv}_{\mathcal{F},\mathcal{H},A}^\text{phec}(k) = 2 \cdot \Pr [ \text{PHCF-DIST}_{\mathcal{F},\mathcal{H}}^A(k) \Rightarrow 1 ] - 1 ,$$

is negligible in $k$. 

---

**Figure 2.7:** Games to define HCF-DIST security.
 deliberate PHCF-DIST $A$ $(k)$

$b \leftarrow \{0, 1\}$

$K_H \leftarrow K_H(1^k)$; $(f, f^{-1}) \leftarrow Kg(1^k)$

$x \leftarrow \{0, 1\}^n$; $h_0 \leftarrow H(K_H, x|^{\ell})$

$h_1 \leftarrow \text{HRng}(k)$

$b' \leftarrow A(K_H, f, f(x), x|^{n-\ell}, h_b)$

Return $(b = b')$

**Figure 2.8:** Games to define PHCF-DIST security.

Note if $(f(x), x|^{n-\ell})$ is a one-way function of $x$, then $\mathcal{H}$ is a $\ell$-partial hardcore function for $\mathcal{F}$ when $\mathcal{H}$ is a computational randomness extractor [47]. This is plausible for the case that $\mathcal{F}$ is $\text{RSA}$ when $n - \ell$ is small enough that Coppersmith’s techniques do not apply. This means $n - \ell \leq n(e - 1)/e - \log 1/\epsilon$ such that $N^\epsilon \geq 2^k$ for security parameter $k$.

**ONE-WAYNESS EXTRACTORS.** Let $\mathcal{H}: K_H \times \text{HDom} \rightarrow \text{HRng}$ be a function family. We say $\mathcal{H}$ is a one-wayness extractor [70] if for any PPT adversary $A$ and any unpredictable distribution $D$ we have

$$\text{Adv}_{\mathcal{H}, A, D}^{\text{cdist}} = \text{Pr} \left[ A(K_H, z, H(K_H, x)) = 1 \right] - \text{Pr} \left[ A(K_H, z, U) = 1 \right],$$

is negligible in $k$, where $K_H \leftarrow K_H(1^k)$, $(z, x) \leftarrow D_k$, and $U \leftarrow \text{HRng}(k)$.

### 2.3.5 The Optimal Asymmetric Encryption Padding Framework

**Padding scheme.** We define a general notion of padding scheme following [6, 73].

For $\nu, \rho, \mu \in \mathbb{N}$, the associated *padding scheme* is a triple of deterministic algorithms $\text{PAD} = (\Pi, \text{PAD}, \text{PAD}^{-1})$ defined as follows. Algorithm $\Pi$ on input a unary encoding of the security parameter $1^k$ outputs a pair $(\pi, \hat{\pi})$ where $\pi : \{0, 1\}^{\mu + \rho} \rightarrow \{0, 1\}^\nu$ and $\hat{\pi} : \{0, 1\}^\nu \rightarrow \{0, 1\}^\mu \cup \{\perp\}$ such that $\pi$ is injective and for all $m \in \{0, 1\}^\mu$ and
Figure 2.9: Padding based encryption scheme $\text{PAD}[\mathcal{F}] = (\text{Kg}, \text{Enc}, \text{Dec})$.

$r \in \{0,1\}^\rho$ we have $\hat{\pi}(\pi(m||r)) = m$. Algorithm $\text{PAD}$ on inputs $\pi$ and $m \in \{0,1\}^\mu$ outputs $y \in \{0,1\}^\nu$. Algorithm $\text{PAD}^{-1}$ on inputs a mapping $\hat{\pi}$ and $y \in \{0,1\}^\nu$ outputs $m \in \{0,1\}^\mu$ or $\perp$.

**PADDING-BASED ENCRYPTION.** Let $\text{PAD}$ be a padding transform from domain $\{0,1\}^{\mu+\rho}$ to range $\{0,1\}^\nu$. Let $\mathcal{F}$ be a TDP with domain $\{0,1\}^\nu$. The associated *padding-based encryption scheme* is a triple of algorithms $\text{PAD}[\mathcal{F}] = (\text{Kg}, \text{Enc}, \text{Dec})$ defined in Figure 2.9.

**OAEP PADDING SCHEME.** We recall the OAEP padding scheme [6]. Let message length $\mu$, randomness length $\rho$, and redundancy length $\zeta$ be integer parameters, and $\nu = \mu + \rho + \zeta$. Let $\mathcal{G}: \mathcal{K}_G \times \{0,1\}^\rho \to \{0,1\}^{\mu+\zeta}$ and $\mathcal{H}: \mathcal{K}_H \times \{0,1\}^{\mu+\zeta} \to \{0,1\}^\rho$ be function families. The associated *OAEP padding scheme* is a triple of algorithms $\text{OAEP}[\mathcal{G}, \mathcal{H}] = (\mathcal{K}_{\text{OAEP}}, \text{OAEP}, \text{OAEP}^{-1})$ defined as follows. On input $1^k$, $\mathcal{K}_{\text{OAEP}}$ returns $(K_G, K_H)$ where $K_G \leftarrow_s \mathcal{K}_G(1^k)$ and $K_H \leftarrow_s \mathcal{K}_H(1^k)$, and $\text{OAEP}, \text{OAEP}^{-1}$ are as defined in Figure 2.10.

**OAEP ENCRYPTION SCHEME AND VARIANTS.** Slightly abusing notation, we denote by $\text{OAEP}[\mathcal{G}, \mathcal{H}, \mathcal{F}]$ the OAEP-based encryption scheme $\mathcal{F}$-OAEP with $n = \nu$. We also consider two other OAEP-based encryption schemes, called $t$-clear and $s$-clear $\mathcal{F}$-OAEP, and denoted $\text{OAEP}_{t\text{-clear}}[\mathcal{G}, \mathcal{H}, \mathcal{F}|_{\zeta+\rho}]$ and $\text{OAEP}_{s\text{-clear}}[\mathcal{G}, \mathcal{H}, \mathcal{F}|_{\mu+\zeta}]$. Here $n =$
Algorithm OAEP<sub>(K<sub>G,H</sub>)</sub>(m∥r)

\[
\begin{align*}
s & \leftarrow (m\|0^\zeta)\oplus G(K_G, r) \\
t & \leftarrow r\oplus H(K_H, s) \\
x & \leftarrow s\|t \\
\text{Return } x
\end{align*}
\]

Algorithm OAEP<sup>-1</sup><sub>(K<sub>G,H</sub>)</sub>(x)

\[
\begin{align*}
s\|t & \leftarrow x \\
r & \leftarrow t\oplus H(K_H, s) \\
m' & \leftarrow s\oplus G(K_G, r) \\
\text{If } m'|\zeta = 0^\zeta \text{ return } m'|\mu \\
\text{Else return } \bot
\end{align*}
\]

Figure 2.10: OAEP padding scheme OAEP[\(G, H\)].

Figure 2.11: FO transform \(\text{FO}_{H,G}[\text{PKE}, \text{SE}] = (\text{FO.Kg, FO.Enc, FO.Dec})\).

\(\mu\) and \(n = \rho\), respectively. We often write \(\text{OAEP}_{t\text{-clear}}\) and \(\text{OAEP}_{s\text{-clear}}\) instead of \(\text{OAEP}_{t\text{-clear}}[G, H, F|\zeta+\rho]\) and \(\text{OAEP}_{s\text{-clear}}[G, H, F|\mu+\zeta]\). We typically think of \(F\) as RSA, and all our results apply to this case under suitable assumptions. Note that, following prior work, despite its name \(t\text{-clear} F\)-OAEP we actually apply \(F\) to only the \(\mu\) most significant bits of the output of the underlying padding scheme, leaving the redundancy part of \(s\) in the clear as well.

2.3.6 The Fujisaki-Okamoto Transform

The Fujisaki-Okamoto (FO) transformation [55] is a technique to convert weak public-key encryption schemes, e.g., IND-CPA secure into strong ones that resist chosen ciphertext attacks (i.e., IND-CCA secure). Let \(\text{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})\) be a private-key
encryption and \( \text{PKE} = (\text{Kg}, \text{Enc}, \text{Dec}) \) be a public-key encryption schemes. Moreover, let \( \mathcal{H} : \mathcal{K}_H \times \text{HDom} \rightarrow \text{HRng} \) and \( \mathcal{G} : \mathcal{K}_G \times \text{GDom} \rightarrow \text{GRng} \) be function families.

We define FO transform \( \text{FO}_{\mathcal{H}, \mathcal{G}}[\text{PKE}, \text{SE}] = (\text{FO.Kg}, \text{FO.Enc}, \text{FO.Dec}) \) in Figure 2.11.

### 2.3.7 Obfuscation

Here we present three different definitions of obfuscation that we use in this paper. We start by recalling the definition of indistinguishability obfuscation from [59].

**Indistinguishability Obfuscation.** A PPT algorithm \( iO \) is called an indistinguishability obfuscator for a circuit ensemble \( \mathcal{C} = \{C_k\}_{k \in \mathbb{N}} \) if the following conditions hold:

- **Correctness:** For all security parameters \( k \in \mathbb{N} \), for all \( C \in \mathcal{C}_k \), and for all inputs \( x \), we have that
  \[
  \Pr [ C'(x) = C(x) : C' \leftarrow \text{iO}(1^k, C) ] = 1.
  \]

- **Security:** For any PPT distinguisher \( D \), for all pairs of circuits \( C_0, C_1 \in \mathcal{C}_k \) such that \( C_0(x) = C_1(x) \) on all inputs \( x \), we have
  \[
  \text{Adv}_{iO,D,C}^\text{iO}(k) = \Pr [ D(1^k, \text{iO}(1^k, C_0)) = 1 ] - \Pr [ D(1^k, \text{iO}(1^k, C_1)) = 1 ] ,
  \]
  is negligible in \( k \).

There are several constructions for indistinguishable obfuscators based on different assumptions. For example, Garg *et al.* in [59] give a construction based on a simplified variant of multilinear maps assumption, which they call multilinear jigsaw puzzles. Moreover, Jain *et al.* in [71] give a construction based on three assumptions of LWE, SXDH and a constant degree perturbation resilient generator. Next, we give a definition for differing-inputs obfuscation.
The definition of differing-inputs obfuscation is similar to indistinguishability obfuscation. While indistinguishability obfuscation requires circuits to be identical on all inputs, differing-inputs obfuscation relaxes this condition and requires circuits to be identical on all inputs except one point. We first recall the definition of differing-inputs obfuscation from [32].

**Differing-inputs circuits.** A sample algorithm \((C_0, C_1, z) \leftarrow \text{Samp}(1^k)\) that samples circuits from a circuit ensemble \(C = \{C_k\}_{k \in \mathbb{N}}\) is a differing-inputs distribution if for all PPT algorithms \(A\), we have

\[
\Pr \left[ C_0(x) \neq C_1(x) : (C_0, C_1, z) \leftarrow \text{Samp}(1^k), x \leftarrow A(1^k, C_0, C_1, z) \right] = \frac{1}{n},
\]

is negligible in \(k\).

**Differing-inputs obfuscation.** A PPT algorithm \(\text{diO}\) is a differing-inputs obfuscator for a differing-inputs distribution \(\text{Samp}\) (for circuit ensemble \(C = \{C_k\}_{k \in \mathbb{N}}\)) if the following conditions hold:

- **Correctness:** For all security parameters \(k \in \mathbb{N}\), for all \(C \in C_k\), and for all inputs \(x\), we have that

\[
\Pr \left[ C''(x) = C(x) : C'' \leftarrow \text{diO}(1^k, C) \right] = 1.
\]

- **Security:** For any PPT distinguisher \(D\) and any \((C_0, C_1, z) \leftarrow \text{Samp}(1^k)\), we have

\[
\text{Adv}_{\text{diO}, D, \text{Samp}}^\text{dio}(k) = \Pr \left[ D(\text{diO}(1^k, C_0, z)) = 1 \right] - \Pr \left[ D(\text{diO}(1^k, C_1, z)) = 1 \right],
\]

is negligible in \(k\).

We note that Boyle, Chung and Pass in [32] show that any general indistinguishability obfuscator for all circuits in \(\mathcal{P}/\mathsf{poly}\) is also a differing-inputs obfuscator for
circuits that differ on at most polynomially many inputs. We now formalizes the def-
inition of unpredictable distributions which are used to define obfuscators for point
functions.

**Point Obfuscation.** Although indistinguishability obfuscation are obfuscation
schemes for general circuits we can also study obfuscation schemes for particular
class of functions such as point functions. A point function $p_x$ for some value $x$ is
defined as follow:

$$p_x(z) = \begin{cases} 
1 & \text{if } z = x \\
\bot & \text{otherwise}
\end{cases}$$

We now recall the definition of point obfuscation from [18]. A PPT algorithm
AIPO is a point obfuscator for unpredictable distributions $\{D_k = (Z_k, X_k)\}_{k \in \mathbb{N}}$ if the
following conditions hold:

- **Correctness:** For all security parameters $k \in \mathbb{N}$, for all $(z, x) \leftarrow D_k$, AIPO on
  input $x$ outputs a polynomial-size circuit that returns 1 on $x$ and 0 everywhere
  else.

- **Security:** To distinguisher $A$, we associate the experiment in Figure 2.12, for
every $k \in \mathbb{N}$. We require that for every PPT distinguisher $A$,

$$\text{Adv}_{\text{AIPO}, A, D}^{\text{aipo}}(k) = 2 \cdot \Pr \left[ \text{AIPO}_{\text{AIPO}}^{D, A}(k) \Rightarrow 1 \right] - 1,$$

be negligible in $k$.

We note that Canetti in [37] give a construction for secure AIPO against computa-
tionally unpredictable samplers under variants of the DDH assumption. Moreover,
Bitansky and Paneth in [18] give a construction for AIPO under assumptions on
pseudorandom permutations.
2.3.8 Puncturable Pseudo-Random Functions

A family of puncturable PRFs PRF with domain \( \text{PRF} \text{.Dom} \) and range \( \text{PRF} \text{.Rng} \) is a tuple of algorithms \( \text{PRF} := (\text{PRF} \text{.Kg}, \text{PRF} \text{.Punct}, \text{PRF} \text{.Eval}) \) that work as follows. Algorithm \( \text{PRF} \text{.Kg} \) on input a unary encoding of the security parameter \( 1^k \) outputs a key \( K \). Algorithm \( \text{PRF} \text{.Eval} \) on inputs a key \( K \) and \( x \in \text{PRF} \text{.Dom}(k) \) outputs \( y \in \text{PRF} \text{.Rng}(k) \). We often write \( \text{PRF}_K(x) \) instead of \( \text{PRF} \text{.Eval}(K, x) \). Additionally, there is a PPT puncturing algorithm \( \text{PRF} \text{.Punct} \) which on inputs a key \( K \) and a polynomial-size set \( S \subseteq \text{PRF} \text{.Dom}(k) \), outputs a special key \( K_S \). We say \( \text{PRF} \) is puncturable PRF if the following two properties hold [90]:

- **Functionality preserved under puncturing**: For every PPT adversary \( A \) such that \( A(1^k) \) outputs a polynomial-size set \( S \subseteq \text{PRF} \text{.Dom}(k) \), it holds for all \( x \in \text{PRF} \text{.Dom}(k) \) where \( x \notin S \) that

  \[
  \Pr \left[ \text{PRF}_K(x) = \text{PRF}_{K_S}(x) : K \leftarrow \text{PRF} \text{.Kg}(1^k), \ K_S \leftarrow \text{PRF} \text{.Punct}(K, S) \right] = 1
  \]

- **Pseudorandom at punctured points**: To attacker \( A = (A_1, A_2) \), we associate the experiment in Figure 2.13 for every \( k \in \mathbb{N} \). We require that for every PPT adversary \( A = (A_1, A_2) \),

  \[
  \text{Adv}_{\text{PRF}, A}^{\text{pprf}}(k) = 2 \cdot \Pr \left[ \text{PRF} \text{-DIST}^A_{\text{PRF}}(k) \Rightarrow 1 \right] - 1,
  \]
be negligible in $k$.

2.3.9 Extremely Lossy Functions

We recall the definition of Extremely Lossy Functions (ELFs) from Zhandry [97]. An Extremely Lossy Function, or ELF, is a similar notion to lossy trapdoor function without a trapdoor in the injective mode, but with a more powerful lossy mode. In a Extremely Lossy Function, the image size in the lossy mode can be taken to be a polynomial $r$. In fact, the polynomial $r$ can be tuned based on the adversary in question to be just large enough to fool the adversary.

A family of extremely lossy functions $\text{ELF}$ with domain $\text{ELF.Dom}$ and range $\text{ELF.Rng}$ is a tuple of algorithms $\text{ELF} = (\text{ELF.IKg}, \text{ELF.LKg}, \text{ELF.Eval})$ that work as follows. Algorithm $\text{ELF.IKg}$ on input a unary encoding of the security parameter $1^k$ outputs the description of a function $f: \text{ELF.Dom}(k) \rightarrow \text{ELF.Rng}(k)$. Algorithm $\text{ELF.LKg}$ on inputs a unary encoding of the security parameter $1^k$ and polynomial $r$ outputs the description of a function $f: \text{ELF.Dom}(k) \rightarrow \text{ELF.Rng}(k)$. Algorithm $\text{ELF.Eval}$ on inputs a function $f$ and $x \in \text{ELF.Dom}(k)$ outputs $y \in \text{ELF.Rng}(k)$. We often write $f(x)$ instead of $\text{ELF.Eval}(f, x)$. We require the following properties:

![Figure 2.13: Games to define PRF-DIST security.](image-url)
• **Correctness:** Let $f$ be the output of $\text{ELF.IKg}(1^k)$. Then, the function $f$ is injective except with negligible probability, for all $k \in \mathbb{N}$.

• **Key-indistinguishability:** For any polynomial $p$ and inverse polynomial function $\delta$, there is a polynomial $q$ such that, for any adversary $A$ running in time at most $p$, and any $r \geq q$, we have that

$$\Pr \left[ A(f) = 1 : f \leftarrow \text{ELF.IKg}(1^k) \right] - \Pr \left[ A(f) = 1 : f \leftarrow \text{ELF.LKg}(1^k, r) \right] < \delta.$$  

In other words, no polynomial-time adversary $A$ can distinguish an injective $f$ from an $f$ with polynomial image size.

• **Lossiness:** Let $f$ be the output of $\text{ELF.LKg}(1^k, r)$. Then, the function $f$ has image size of at most $r$ except with negligible probability, for all polynomial $r$.

• **Pseudorandomness:** For any PPT adversary $A$ and $k \in \mathbb{N}$, we require that

$$\text{Adv}_{\text{ELF},A}^\text{elf}(k) = \Pr [ A(f(x), f) = 1 ] - \Pr [ A(U, f) = 1 ],$$

where $f \leftarrow \text{ELF.IKg}(1^k), x \leftarrow \text{ELF.Dom}(k)$, and $U \leftarrow \text{ELF.Rng}(k)$, be negligible in $k$. 
In this chapter, we talk about the extractability notation. Intuitively, extractability of a function formalizes the idea that an adversary that produces an image point must “know” a corresponding preimage, as there being a non-blackbox extractor that recovers the preimage. Previous work on extractability starting with [38, 39] considers a “one-shot” adversary. Inspired by PA for encryption schemes [4, 9], we define a hierarchy of EXT for function families, namely EXT0, EXT1, EXT2, and EXT-RO, which will in particular be useful for our full instantiation results. (Even our notion of EXT0 generalizes prior work, as explained below.) However, there are some important differences from PA. First, for EXT the extractor should return the entire preimage whereas in PA the extractor need not return the randomness. Second, PA asks the adversary to distinguish between the answers of the decryption and extraction oracles, while EXT asks the adversary to make the extractor fail to return a preimage.

3.1 EXT0 Functions

We first give a “one-time” definition of extractability. Let $\eta, \zeta, \mu$ be integer parameters. A function family $\mathcal{H} : \mathcal{K}_H \times \text{HDom} \rightarrow \text{HRng}$ is $(\eta, \mu)$-EXT0 if for any PPT adversary $A$ with coin space $\text{Coins}$, there exists a PPT extractor $\mathcal{E}$ such that, for any key
independent auxiliary input \( z \in \{0, 1\}^{\eta} \):

\[
\text{Adv}^{(\eta, \mu)-\text{ext}_{0, \zeta}}(k) = \max_{z \in \{0, 1\}^{\eta}} \text{Adv}^{(\eta, \mu)-\text{ext}_{0, \zeta}}(k).
\]

is negligible in \( k \). We define advantage of adversary \( A \) to be \( \text{Adv}^{(\eta, \mu)-\text{ext}_{0, \zeta}}(k) = \max_{z \in \{0, 1\}^{\eta}} \text{Adv}^{(\eta, \mu)-\text{ext}_{0, \zeta}}(k) \).

In other words, the extractor should work when the adversary outputs \( \zeta \) least significant bits of an image point and \( \mu \) bits of a preimage, given \( \eta \) bits of auxiliary information. Previous work considered \( \zeta = \log |\text{HRng}| \) and \( \mu = 0 \). Interestingly, considering \( \mu > 0 \) gives a non-blackbox second-input extractability (SIE) notion compared to Section 5.2.1, which has a black-box notion of SIE. Our non-blackbox notion applies to general function families rather than trapdoor permutations. We also retain the generality afforded by \( \eta, \zeta, \mu \) below.

Similarly, we define the analogous notion \( (\eta, \mu)-\text{EXT}_{0, \zeta} \) where the adversary outputs the \( \zeta \) most significant bits of the image point. We often write \( \eta\text{-EXT}_{0, \zeta} \) and \( \eta\text{-EXT}_{0, \zeta} \) instead of \( (\eta, 0)\text{-EXT}_{0, \zeta} \) and \( (\eta, 0)\text{-EXT}_{0, \zeta} \), respectively. We also often write \( (\eta, \mu)\text{-EXT}_{0} \) instead of \( (\eta, \mu)\text{-EXT}_{0, \zeta} \) when \( \zeta = \log |\text{HRng}| \).
3.2 EXT1 Functions

Next, we give a definition of “many-times” extractability. We note that a central open problem in the theory of extractable functions to construct a “many-times” extractable function from a “one-time” extractable function, see e.g. [61]; the obvious approach suffers an extractor “blow-up” issue. For practical purposes, we simply formalize and assume this property for an appropriate construction from cryptographic hashing.

Let $\eta, \zeta, \mu$ be integer parameters. Let $\mathcal{H}: K_H \times \text{HDom} \to \text{HRng}$ be a hash function family. To an adversary $A$ and extractor $E$, we associate the experiment in Figure 3.1, for every $k \in \mathbb{N}$. We say $\mathcal{H}$ is $(\eta, \mu)$-EXT$1_\zeta$ if for any PPT adversary $A$ with coin space $\text{Coins}$, there exists a stateful extractor $E$ such that, for any key independent auxiliary input $z \in \{0, 1\}^\eta$:

$$\text{Adv}^{(\eta, \mu)-\text{ext1}_\zeta}_{\mathcal{H}, A, E, z}(k) = \Pr_{K_H \leftarrow K_H(1^k), r \leftarrow \text{Coins}(k)} \left( (x, y) \leftarrow \text{EXT1}_{\mathcal{H}}^{A, E, z}(K_H, r) \right. \\
\exists i, \exists x : H(K_H, x)|_\zeta = y[i] \land H(K_H, x[i])|_\zeta \neq y[i] \left. \right)$$

is negligible in $k$. We define advantage of adversary $A$ to be $\text{Adv}^{(\eta, \mu)-\text{ext1}_\zeta}_{\mathcal{H}, A, E}(k) = \max_{z \in \{0, 1\}^\eta} \text{Adv}^{(\eta, \mu)-\text{ext1}_\zeta}_{\mathcal{H}, A, E, z}(k)$. Similarly, we define $(\eta, \mu)$-EXT$1_\zeta$ where the adversary output $\zeta$ most significant bits of the output to be extracted. We often write $\eta$-EXT$1_\zeta$ and $\eta$-EXT$1_\zeta$ instead of $(\eta, 0)$-EXT$1_\zeta$ and $(\eta, 0)$-EXT$1_\zeta$, respectively.

3.3 EXT2 Functions

We now extend the definition to give the adversary access to an oracle $\mathcal{I}$ that on input a unary encoding of the security parameter $1^k$, outputs a random image along with an uninvertible hint of the corresponding preimage. In other words, this is a
form of extractability with key dependent auxiliary information that parallels PA2 for encryption schemes.

Let $\eta, \zeta, \mu$ be integer parameters. Let $\mathcal{H} : \mathcal{K}_H \times \text{HDom} \rightarrow \text{HRng}$ be a hash function family and $\mathcal{F} = (\text{Kg}, \text{Eval}, \text{Inv})$ be a trapdoor permutation family with domain $\text{HDom}$. To adversary $A$ and extractor $\mathcal{E}$, we associate the experiment in Figure 3.2, for every $k \in \mathbb{N}$. We say $\mathcal{H}$ is $(\eta, \mu)$-EXT2$_\zeta$ with respect to trapdoor permutation family $\mathcal{F}$ if for any PPT adversary $A$ with coin space $\text{Coins}$, there exists a stateful extractor $\mathcal{E}$ such that, for any key independent auxiliary input $z \in \{0, 1\}^n$:

$$\text{Adv}^{(\eta, \mu) \text{-ext2}_{\zeta}}_{\mathcal{H}, \mathcal{F}, A, \mathcal{E}, z}(k) = \Pr_{K_H \leftarrow \mathcal{K}_H(1^k)} \left[ \exists i, \exists x : H(K_H, x) = y[i] \land x[i] \mid \mu = x \mid \mu \land H(K_H, x[i]) \neq y[i] \right],$$

is negligible in $k$. The adversary is not allowed to query $y \in h_1$ for extract oracle $\mathcal{O}$. We define advantage of $A$ to be $\text{Adv}^{(\eta, \mu) \text{-ext2}_{\zeta}}_{\mathcal{H}, \mathcal{F}, A, \mathcal{E}, z}(k) = \max_{z \in \{0, 1\}^n} \text{Adv}^{(\eta, \mu) \text{-ext2}_{\zeta}}_{\mathcal{H}, \mathcal{F}, A, \mathcal{E}, z}(k)$. Similarly, we define $(\eta, \mu)$-EXT2$^\zeta$ where the adversary output $\zeta$ most significant bits of the output to be extracted. We often write $\eta$-EXT2$^\zeta$ and $\eta$-EXT2$^\zeta$ instead of $(\eta, 0)$-EXT2$^\zeta$ and $(\eta, 0)$-EXT2$^\zeta$, respectively.
3.4 EXT-RO Functions

Finally, we give a notion of extractability in the RO model, inspired by PA-RO for encryption schemes. In particular, here the adversary has access to an oracle $F$ to which it queries a sampling algorithm, the oracle returning the image of a point in the domain sampled accordingly. Moreover, instead of the adversary’s random coins the extractor gets a transcript of its RO queries and responses, but not those made by $F$.

Let $\zeta$ be an integer parameter. Let $\mathcal{O}: \{0, 1\}^* \rightarrow \{0, 1\}^*$ be a random oracle and $\mathcal{F}: \mathcal{K}_F \times \mathcal{F}.\text{Dom} \rightarrow \mathcal{F}.\text{Rng}$ be a function family. To adversary $A$ and extractor $\mathcal{E}$, we associate the experiment in Figure 3.1, for every $k \in \mathbb{N}$. We say $\mathcal{F}$ is $\zeta$-EXT-RO if for any PPT adversary $A$, there exists a PPT extractor $\mathcal{E}$ such that,

$$\text{Adv}^\zeta_{\mathcal{F}, A, \mathcal{E}}(k) = \Pr_{K_F \leftarrow \mathcal{K}_F(1^k)} \left[ \exists i, \exists x : F(K_F, x)|_\zeta = y[i] \land F(K_F, x[i])|_\zeta \neq y[i] \right],$$

is negligible in $k$. The adversary is not allowed to query $y \in \mathcal{O}$ for extract oracle $\mathcal{O}$. Note that above, the parameter $\zeta$ controls not only what part of the output the
adversary queries to its extract oracle, but also what part of the output the image oracle returns. This stems from how we use EXT-RO. Namely, we only apply it to the OAEP padding scheme (in Theorem 10), which is invertible. As a consequence, in the RO model EXT-RO does not imply EXT2.

3.5 Plausibility

We typically use EXT notions in tandem with other properties such as collision-resistance. In terms of feasibility, there are several constructions proposed for EXT0 with $\zeta = \log |HRng|$ and $\mu = 0$ and collision-resistance in [81] based on knowledge assumptions. (In the weaker case of EXT0 with only one-wayness, which does not suffice for us, the notion is actually achievable for these parameters under standard assumptions [20].) However, for our generalizations and notions of EXT1, EXT2, we are not aware of any constructions in the standard model. Despite the fact that they are difficult to judge, it may be a reality that as a community we need to move to such assumptions in order to make progress on some difficult problems. A similar strategy was used for very different goals by Pandey et al. [83]. It would be interesting for future work to explore relations between our assumptions and theirs.
Chapter 4

Results for Padding Schemes and Optimal Asymmetric Encryption Padding

For increased modularity and understanding, we abstract properties of the OAEP padding scheme we need and give some results about how to achieve them based on assumptions on the round functions. That is, at a very high level we would like to show that if $G$ is $\text{xxx}$-secure and $H$ is $\text{yyy}$-secure then the OAEP padding scheme is $\text{zzz}$ secure. For example, $\text{xxx} = \text{yyy} = \text{zzz} = \text{EXT}0$. Naturally, the actual results, while retaining this flavor, are much more nuanced.

4.1 Scope and Perspective

In all properties of the OAEP padding scheme we consider, the adversary produces part of the output. In particular, there are two parameter regimes we consider, one where the adversary produces the least-significant $(\zeta + \rho)$-bits of the output (corresponding to $t$-clear OAEP), and one where the adversary produces the most-significant $(\mu + \rho)$-bits of the output (corresponding to $s$-clear OAEP). In terms of properties, we consider near-collision resistance, EXT0, EXT1, and EXT-RO. We do not consider EXT2 of the OAEP padding scheme at all.

If we prove a security notion for the OAEP padding scheme based on corresponding assumptions on the round functions, its plausibility reduces to plausibility of the assumptions on the round functions. A case in which we do not know how to do this is that of EXT1 for the “$t$-clear” parameter regime. In this case, we lend some
plausibility to this assumption by showing EXT-RO, which implies EXT1, holds in the RO model. Similar justification was made for assuming a hash function is UCE in [15, Section 6.1].

4.2 Our Results

We first show that the OAEP padding transform is near-collision resistant wrt. its least-significant bits (i.e., for “t-clear” parameters) if \( G \) is near-collision resistant wrt. its least-significant bits and \( H \) is collision-resistant.

**Theorem 6** Let \( \mu, \zeta, \rho \) be integer parameters. Let \( G : \mathcal{K}_G \times \{0, 1\}^{\mu + \zeta} \rightarrow \{0, 1\}^{\mu + \zeta} \) and \( H : \mathcal{K}_H \times \{0, 1\}^{\mu + \zeta} \rightarrow \{0, 1\}^\rho \) be function families. Suppose \( G \) is NCR\( \zeta \) and \( H \) is collision resistant. Then OAEP\([G, H]\) is NCR\( \zeta + \rho \). In particular, for any adversary \( A \), there exists adversaries \( B, C \) such that

\[
\text{Adv}^{n-cr}_{\text{OAEP}[G, H], A}(k) \leq \text{Adv}^{n-cr}_G(k) + \text{Adv}^c_H(k).
\]

The running time of \( B \) and \( C \) are about that of \( A \).

**Proof.** Consider near-collision resistance adversary \( B \) and collision resistance adversary \( C \) in Figure 4.1. Let \( v_1, v_2 \) be the outputs of \( A \). Let \( S \) be the event where \( v_1 \neq v_2 \) are not equal and OAEP\(_{\mathcal{K}_G, \mathcal{K}_H}(v_1)|_{\zeta + \rho} = \text{OAEP}_{\mathcal{K}_G, \mathcal{K}_H}(v_2)|_{\zeta + \rho} \). Let \( E \) be the event where the \( \rho \) least significant bits of \( v_1 \) and \( v_2 \) are not equal. Thus,

\[
\text{Adv}^{n-cr}_{\text{OAEP}[G, H], A}(k) = \text{Pr}[S \land E] + \text{Pr}[S \land \overline{E}].
\]

Note that if the event \( S \) and \( E \) happens the adversary \( B \) finds a collision. Thus,

\[
\text{Pr}[S \land E] \leq \text{Adv}^{n-cr}_G(k).
\]

On the other hand, if \( \overline{E} \) happens then the \( \rho \) least significant bits of \( v_1 \) and \( v_2 \) are equal. Moreover, we know \( v_1 \) and \( v_2 \) are not equal, thus when the event \( \overline{E} \) happens
the $\mu$ most significant bits of $v_1$ and $v_2$ are not equal. Therefore, if the event $S$ and $\overline{E}$ happens adversary $C$ finds a collision. Then

$$\Pr \left[ S \land \overline{E} \right] \leq \Adv_{\H,C}^{\text{cr}}(k).$$

Summing up,

$$\Adv_{\text{OAEP}[G,H],A}^{\mu+\zeta}(k) \leq \Adv_{G,B}^{\mu+\zeta}(k) + \Adv_{H,C}^{\text{cr}}(k).$$

This completes the proof.

**Theorem 7** Let $\mu, \zeta, \rho$ be integer parameters. Let $\mathcal{G} : K_G \times \{0,1\}^\rho \to \{0,1\}^{\mu+\zeta}$ and $\mathcal{H} : K_H \times \{0,1\}^{\mu+\zeta} \to \{0,1\}^\rho$ be function families. Suppose $\mathcal{G}$ is NCR$_\zeta$. Then OAEP$[\mathcal{G},\mathcal{H}]$ is NCR$^{\mu+\zeta}$. In particular, for any adversary $A$, there exists adversary $B$ such that

$$\Adv_{\text{OAEP}[G,H],A}^{\mu+\zeta}(k) \leq \Adv_{G,B}^{\mu+\zeta}(k).$$

The running time of $B$ is about that of $A$.

**Proof.** Let $v_1, v_2$ be the outputs of $A$. Let $S$ be the event where $v_1 \neq v_2$ are not equal and $\text{OAEP}_{K_G,K_H}(v_1)^{\mu+\zeta} = \text{OAEP}_{K_G,K_H}(v_2)^{\mu+\zeta}$. Then, we have $\Adv_{\text{OAEP}[G,H],A}^{\mu+\zeta}(k) = \Pr[S]$. 

---

<table>
<thead>
<tr>
<th>Adversary $B(K_G)$</th>
<th>Adversary $C(K_H)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_H \leftarrow K_H(1^k)$</td>
<td>$K_G \leftarrow K_G(1^k)$</td>
</tr>
<tr>
<td>$v_1, v_2 \leftarrow A(K_H, K_G)$</td>
<td>$v_1, v_2 \leftarrow A(K_H, K_G)$</td>
</tr>
<tr>
<td>$r_1 \leftarrow v_1</td>
<td>\rho$; $r_2 \leftarrow v_2</td>
</tr>
<tr>
<td>If $r_1 \neq r_2$ then return $(r_1, r_2)$</td>
<td>If $r_1 \neq r_2$ then return $\bot$</td>
</tr>
<tr>
<td>Return $\bot$</td>
<td>$s_1 \leftarrow m_1 | 0^\mu \oplus G(K_G, r_1)$</td>
</tr>
<tr>
<td></td>
<td>$s_2 \leftarrow m_2 | 0^\mu \oplus G(K_G, r_2)$</td>
</tr>
<tr>
<td></td>
<td>Return $(s_1, s_2)$</td>
</tr>
</tbody>
</table>

**Figure 4.1:** Adversaries $B$ and $C$ in the proof of Theorem 6.
Adversary $B(K_G)$

$K_H \leftarrow K_H(1^k)$

$v_1, v_2 \leftarrow A(K_H, K_G)$

$r_1 \leftarrow v_1|_\rho; r_2 \leftarrow v_2|_\rho$

Return $(r_1, r_2)$

Figure 4.2: Adversary $B$ in the proof of Theorem 7.

Note that when $\text{OAEP}_{K_G, K_H}(v_1)|^{\mu+\zeta} = \text{OAEP}_{K_G, K_H}(v_2)|^{\mu+\zeta}$, we have $G(K_G, v_1|_\rho)|_\zeta = G(K_G, v_2|_\rho)|_\zeta$. Moreover, since $\text{OAEP}_{K_G, K_H}(v_1)|^{\mu+\zeta} = \text{OAEP}_{K_G, K_H}(v_2)|^{\mu+\zeta}$, if $v_1|_\rho$ equals to $v_2|_\rho$, $v_1$ will be equal to $v_2$. Consider near-collision resistance adversary $B$ in Figure 4.2. When adversary $A$ finds a near collision, $B$ also finds a near collision. Therefore we have $\Pr[S] \leq \text{Adv}^{\text{n-cr}_\zeta}(G, B, k)$. Summing up,

$$\text{Adv}^{\text{n-cr}_\mu+\zeta}_{\text{OAEP}[G, H], A}(k) \leq \text{Adv}^{\text{n-cr}_\zeta}(G, B, k).$$

This completes the proof.

4.2.1 EXT0 Result

In more detail, we show that the OAEP padding transform is EXT0 wrt. its least-significant bits (i.e., for “$t$-clear” parameters) if $G$ is EXT0 wrt. its least significant bits and injective, and $H$ is also suitably EXT0. Namely, for $H$ the extractor gets the image point and $\zeta$-bits of preimage, so since $H$ maps $\{0,1\}^{\zeta+\mu}$ to $\rho$, if $\mu \ll \rho$ this assumption would be reasonable.

**Theorem 8** Let $\eta, \delta, \lambda, \mu, \zeta, \rho$ be integer parameters. Let $G : \mathcal{K}_G \times \{0,1\}^\rho \rightarrow \{0,1\}^{\mu+\zeta}$ and $H : \mathcal{K}_H \times \{0,1\}^{\mu+\zeta} \rightarrow \{0,1\}^\rho$ be function families. Let $\eta = ||\mathcal{K}_H(1^k)|| + \lambda$ and $\delta = ||\mathcal{K}_G(1^k)|| + \lambda$. Suppose $G$ is $\eta$-EXT0$_\zeta$, $\zeta$-injective and $H$ is $(\delta, \zeta)$-EXT0. Then, $\text{OAEP}[G, H]$ is $\lambda$-EXT0$_{\zeta+\rho}$. In particular, for any EXT0 adversary $A$, there exist
Adversary $A_H(K_H, u; w)$
out $\leftarrow \perp$ ; $(K_G, aux) \leftarrow u$
Run $A^{O_{Sim}}((K_G, K_H), aux; w)$
Return $(s_2, out)$

Procedure $O_{Sim}(y)$
$(s_2, t) \leftarrow y$ ; $v \leftarrow (K_H, aux)$
r $\leftarrow Ext_G(K_G, v, \perp, s_2; w)$
out $\leftarrow r \oplus t$
Halt $A$

Adversary $A_G(K_G, v; w)$
out $\leftarrow \perp$ ; $(K_H, aux) \leftarrow v$
Run $A^{O_{Sim}}((K_G, K_H), aux; w)$
Return $(\perp, out)$

Procedure $O_{Sim}(y)$
$(s_2, t) \leftarrow y$
out $\leftarrow s_2$
Halt $A$

Figure 4.3: Adversaries $A_H, A_G$ in the proof of Theorem 8.

The running time of $A_G$ is about that of $A$. The running time of $A_H$ is about that of $A$ plus the time to run $Ext_G$. The running time of $Ext$ is the time to run $Ext_G$ and $Ext_H$.

Proof. Let $w$ be the randomness of adversary $A$, and let $aux \in \{0,1\}^\lambda$ be the key-independent auxiliary input to the adversary $A$ in the game EXT0. Let $K_G$ and $K_H$ be the keys for function families $G$ and $H$ respectively. Let $v = (K_H, aux)$ be the key-independent auxiliary input to $A_G$ in the game EXT0, and let $u = (K_G, aux)$ be the key-independent auxiliary input to $A_H$ in the game EXT0. Note that auxiliary input $v$ and $u$ are independent of keys $K_G$ and $K_H$, respectively. We define adversaries $A_G, A_H$ with random coins $w$ in Figure 4.3. Let $Ext_G$ and $Ext_H$ be the corresponding extractor for $A_G$ and $A_H$, respectively. We define EXT0 extractor $Ext$ as shown in Figure 4.4.

Note that for the extract query $y$ that $A$ makes, if $y$ is not a valid image point then extractor $Ext$ outputs $\perp$. Thus, adversary $A$ does not win by making an invalid image

\[
Adv_{^{\text{OAEP}}, A, Ext}^{\lambda \rightarrow \text{ext0}^{\zeta+\rho}}(k) \leq Adv_G^{\eta \rightarrow \text{ext0}^{\zeta}}_{\text{OAEP}, A, Ext}(k) + Adv_{H, A_H, Ext_H}^{(\delta, \zeta) \rightarrow \text{ext0}}(k).
\]
query. Hence, we assume wlog that $A$ only queries a valid image point $y$. Let $m_y \parallel r_y$ be the corresponding input for $y$ and $s_y = G(K_G; r_y) \oplus (m_y \parallel 0^\zeta)$ be the intermediate value. Let $r$ be the output of $\text{Ext}_G$ and $s_1$ be the output of $\text{Ext}_H$. Wlog, we can assume when extractors $\text{Ext}_G$ and $\text{Ext}_H$ output a non-empty string, they were successful in finding preimages.

Consider EXT0 adversaries $A_G, A_H$ in Figure 4.3. We know $A$ always makes a valid image query. Thus, $A$ wins if extractor $\text{Ext}$ outputs $\perp$. On the other hand, $\text{Ext}$ outputs $\perp$ only if either $\text{Ext}_G$ or $\text{Ext}_H$ fails. Moreover, we know if $\text{Ext}_G$ outputs a non-empty string, then it will return $r = r_y$, since $G$ is $\zeta$-injective. Therefore,

$$\text{Adv}_{\text{EXT0} \zeta + \rho}^{\lambda-\text{OAEP}, A, \text{Ext}}(k) \leq \text{Adv}_{\text{EXT0} \zeta}^{\eta-\text{OAEP}, G, A_G, \text{Ext}_G}(k) + \text{Adv}_{\text{EXT0} \zeta}^{(\delta, \zeta)-\text{EXT0}, H, A_H, \text{Ext}_H}(k).$$

This completes the proof.

4.2.2 EXT1 Result

We next show that the OAEP padding transform is EXT1 wrt. its most-significant bits (i.e., "s-clear" parameters) when $G$ is EXT1 wrt. its least-significant bits. Note this does not use any assumption on $H$.

---

**Algorithm** $\text{Ext}((K_G, K_H), \text{aux}, \perp, y; w)$

$(s_2, t) \leftarrow y; v \leftarrow (K_H, \text{aux}); u \leftarrow (K_G, \text{aux})$

$r \leftarrow \text{Ext}_G(K_G, v, \perp, s_2; w); z \leftarrow r \oplus t$

$s_1 \leftarrow \text{Ext}_H(K_H, u, s_2, z; w); s \leftarrow s_1 \parallel s_2$

$m^* \leftarrow s \oplus G(K_G, r); m \leftarrow m^* \parallel \mu$

If $m^* \parallel \zeta \neq 0^k$ then return $\perp$

If $y \neq \text{OAEP}(K_G, K_H)(m \parallel r) \parallel \zeta + \rho$ then return $\perp$

Return $m \parallel r$

---

**Figure 4.4:** EXT0 extractor $\text{Ext}$ in the proof of Theorem 8.
Theorem 9 Let $\eta, \lambda, \mu, \zeta, \rho$ be integer parameters. Let $G : K_G \times \{0, 1\}^\rho \to \{0, 1\}^{\mu + \zeta}$ and $H : K_H \times \{0, 1\}^{\mu + \zeta} \to \{0, 1\}^\rho$ be function families. Let $\eta = ||K_H(1^k)|| + \lambda$. Suppose $G$ is $\eta$-EXT1$_\zeta$. Then, OAEP[$G, H$] is $\lambda$-EXT1$_{\mu + \zeta}$. In particular, for any EXT1 adversary $A$, there exists an EXT1 adversary $A_G$ and an extractor $Ext$ such that for all extractor $Ext_G$

$$\text{Adv}^{\lambda-\text{ext1}_{\mu + \zeta}}_{\text{OAEP}, A, \text{Ext}}(k) \leq \text{Adv}^{\eta-\text{ext1}_{\zeta}}_{G, A_G, \text{Ext}_G}(k).$$

The running time of $A_G$ is about that of $A$ and the running time of $Ext$ is about that of $Ext_G$.

Proof. Let $w$ be the randomness of adversary $A$, $aux \in \{0, 1\}^\lambda$ be the key independent auxiliary input to adversary $A$ in the game EXT1. Let $K_G$ and $K_H$ be the key for the function family $G$ and $H$ respectively. Let $v = (K_H, aux)$ be the key independent auxiliary input to $A_G$ in the game EXT1. We define adversary $A_G$ with the randomness $w$ in Figure 4.5. Let $Ext_G$ be the corresponding extractor for $A_G$. We also define EXT1 extractor $Ext$ as shown in Figure 4.6.

Note that for every extract query $y$ that $A$ makes, if $y$ is not valid then extractor $Ext$ outputs $\bot$. Thus, adversary $A$ does not win by making an invalid image query. Hence, we assume wlog that the adversary $A$ only queries for the valid images.
We define $S$ to be the event that extractor $\text{Ext}$ outputs empty string on at least one of the extract queries. Note that, since we assume that $A$ queries only for the valid images then $\text{Adv}_{\text{OAEP},A,\text{Ext}}^\lambda_{\mu+\zeta}(k) = \Pr[S]$. Moreover, we know $\text{Ext}$ output empty string only if $\text{Ext}_G$ outputs empty string on valid image query. Thus, we have $\Pr[S] \leq \text{Adv}_{\mathcal{G},A_G,\text{Ext}_G}^\eta_{\mu+\zeta}(k)$.

Summing up,

$$\text{Adv}_{\text{OAEP},A,\text{Ext}}^\lambda_{\mu+\zeta}(k) \leq \text{Adv}_{\mathcal{G},A_G,\text{Ext}_G}^\eta_{\mu+\zeta}(k).$$

This completes the proof.

### 4.2.3 EXT-RO Result

We would also like to show that the OAEP padding transform is EXT1 wrt. its least-significant bits (i.e., “$t$-clear” parameters) but we are unable to do so. (The straightforward approach has an “extractor blow-up” problem.) To lend plausibility to this assumption, we instead turn to the RO model and show that the OAEP padding transform is EXT-RO (which implies EXT1) if $\mathcal{G}$ and $\mathcal{H}$ are modeled as ROs.
Theorem 10 Let $\mathcal{G} : \{0,1\}^\rho \rightarrow \{0,1\}^{\mu+\zeta}$ and $\mathcal{H} : \{0,1\}^{\mu+\zeta} \rightarrow \{0,1\}^\rho$ be ROs. Then $\text{OAEP}[\mathcal{G}, \mathcal{H}]$ is $(\zeta + \rho)$-\text{EXT-RO}. In particular, for any adversary $A$, there exists an extractor $\text{Ext}$ such that,

$$
\text{Adv}^{(\zeta+\rho)\text{-ext-\text{RO}}}_{\text{OAEP}[\mathcal{G},\mathcal{H}],A,\text{Ext}}(k) \leq q_1 \cdot 2^{\mu-\rho} + q_1 \cdot 2^{\mu-\zeta} + \frac{q_1 (q_2 + q_3)}{2^\zeta} + \frac{q_1 (q_2 + q_3)^2}{2^\zeta} .
$$

where $q_1$ is the total number of extract queries, $q_2$ is the total number of image oracle queries and $q_3$ is the total number of random oracle queries made by $A$.

Proof. For any adversary $A$, we define the extractor $\text{Ext}$ as shown in Figure 4.7. Now, we bound the probability that extractor $\text{Ext}$ fails on at least one of the extract queries made by adversary $A$. We define $R_I$ and $S_I$ to be the set of queries made by image oracle to $\mathcal{G}$ and $\mathcal{H}$, respectively. We also define $R_A$ and $S_A$ to be the set of queries made by $A$ to $\mathcal{G}$ and $\mathcal{H}$, respectively. Let $R = R_I \cup R_A$ and $S = S_I \cup S_A$. We define $C$ to be the event where there exists $r_1, r_2 \in R$ such that $\mathcal{G}(r_1)|_\zeta = \mathcal{G}(r_2)|_\zeta$. Note that we have $\Pr[C] \leq (q_2 + q_3)^2/2^\zeta$. Let $y_i$ be $i$-th extract query made by $A$, for all $i \in [q_1]$. For all $i \in [q_1]$, we define $S_i$ to be the event where $y_i$ is a valid image and extractor $\text{Ext}$ outputs $\perp$ on input $y_i$. Then,

$$
\text{Adv}^{(\zeta+\rho)\text{-ext-\text{RO}}}_{\text{OAEP}[\mathcal{G},\mathcal{H}],A,\text{Ext}}(k) \leq \Pr[C] + \sum_i \Pr[S_i \wedge \overline{C}] .
$$

We define $R'$ to be the set of all $r \in R$ where $\mathcal{G}(r)|_\zeta = y_i|_\zeta$. Note that when $\overline{C}$ happens, there are no collision on set $R$ and we get that $|R'| \leq 1$. We define $E$ to be the event where $|R'| = 0$. Note that when $E$ happens, challenge $y_i$ is a valid image if there exist $s \in S$ such that $s|_\zeta = y_i|_\zeta$ and $\mathcal{G}(\mathcal{H}(s) \oplus y_i|_\rho)|_\zeta = y_i|_\zeta$ or if there exist $s \notin S$ such that $s|_\zeta = y_i|_\zeta$ and $\mathcal{G}(\mathcal{H}(s) \oplus y_i|_\rho)|_\zeta = y_i|_\zeta$. Therefore, we obtain that $

\Pr[S \wedge \overline{C} \wedge E] \leq 2^{\mu-\zeta} + (q_2 + q_3)/2^\zeta .

Let $Z = \{ z = r \oplus y_i|_\rho : \forall r \in R' \}$. We know when $\overline{C}$ and $\overline{E}$ happens, size of the set $Z$ is equal to 1. Let $z$ be such a element in $Z$. Let $S' = \{ s \in S : \mathcal{H}(s) = z \wedge s|_\zeta = y_i|_\zeta \}$.
We define $T$ to be the event where $S'$ is empty. Note that when $C$, $E$ and $T$ happens, challenge $y_i$ is a valid image if there exist $s$ such that $H(s) = z$. Therefore, we obtain that $\Pr [S \land \overline{C} \land \overline{E} \land T] \leq 2^{\mu - \rho}$.

Note that when $\overline{T}$ happens, we have two cases. First, we have that $S_A \cap S'$ is non-empty. Note that when $S_A \cap S'$ is non-empty, extractor $\text{Ext}$ can extract the correct preimage. Next, we have that $S_A \cap S'$ is empty. Then, we know that $S_I \cap S'$ is non-empty. Note that when $S_I \cap S'$ is non-empty, we get collision on $G$ if challenge $y_i$ is valid image. Thus, we obtain that $\Pr [S \land \overline{C} \land \overline{E} \land \overline{T}] = 0$.

Summing up,

$$\text{Adv}_{\text{OAEP}[G,H],A,\text{Ext}}^{(\zeta+\rho)\text{-ext-ro}}(k) \leq q_1 \cdot 2^{\mu - \rho} + q_1 \cdot 2^{\mu - \zeta} + \frac{q_1 (q_2 + q_3)}{2^\zeta} + \frac{q_1 (q_2 + q_3)^2}{2^\zeta}.$$ 

This completes the proof.
In this chapter, we show new partial and full instantiations *under chosen-ciphertext attack* (CCA) for the RSA-OAEP encryption scheme [6] and some variants. This helps explain why the scheme, which so far has only been shown to have such security in the random oracle (RO) model, has stood up to cryptanalysis despite the existence of “uninstantiable” RO model schemes. It also leads to the fastest CCA-secure RSA-based public-key encryption scheme in the standard model (where one assumes standard-model properties of cryptographic hash functions) to date.

**5.1 Our Technique**

We use several techniques to instantiate RSA-OAEP and its variants. We proceed to describe our techniques and results in more detail.

**5.1.1 Using PA + IND-CPA**

A common thread running through our analyses is the use of *plaintext awareness* (PA) [4, 6, 9]. PA captures the intuition that an adversary who produces a ciphertext must “know” the corresponding plaintext. It is not itself a notion of privacy, but, at a high level, combined with IND-CPA it implies IND-CCA. We use this approach to obtain modularity in proofs, isolate assumptions needed, and make the overall
analyses more tractable. Moreover, while it seems that PA necessitates using knowledge assumptions, this is somewhat inherent due to black-box impossibility results discussed below.

**Flavors and implications.** PA comes in various flavors: PA-RO [9], and PA0, PA1, and PA2 [4]. PA-RO refers to a notion in the RO model, while PA0, PA1, and PA2 refer to standard model notions that differ in what extent the adversary can query its decryption or encryption oracles. (In particular, in PA2 the adversary can query for encryptions of unknown plaintexts.) Similarly, IND-CCA comes in flavors [9, 86]: IND-CCA0, IND-CCA1, and IND-CCA2. We use that [4, 9] show that IND-CPA + PA-RO implies IND-CCA2 in the RO model, IND-CPA + PA0 implies IND-CCA1 with one decryption query, IND-CPA + PA1 implies IND-CCA1, and IND-CPA + PA2 implies IND-CCA2.

**Extractable functions.** One of the heroes for us to show PA security is a hierarchy of “extractability” notions we define and assume for the round functions, called EXT-RO, EXT0, EXT1, EXT2, roughly paralleling PA-RO, PA0, PA1, PA2 respectively, and significantly generalizing prior work [38, 39].

### 5.1.2 Using ELF + iO

**A unified paradigm.** In this approach, we instantiate hash functions $G$ and $H$ via a new unified paradigm of as *obfuscating an extremely lossy function* (ELF). We combine this with prior paradigms Brzuska and Mittelbach [35] (using point function obfuscation in the proof). To explain ELFs, we first recall the notion of lossy function (LF). The key for a LF can be generated in one of two possible modes, the injective and lossy modes, where the first induces an injective function and the second induces a

---

1By obfuscating an ELF, we mean obfuscating the program that evaluates it, with the key hard-coded.
highly non-injective one. Further, keys be indistinguishable to any efficient adversary. Note that the image of the lossy function cannot be too low here, else there would be a trivial distinguisher. ELFs achieve much more lossiness by reversing the order of quantifiers. Namely, for an ELF, for every adversary there exists an (adversary-dependent) indistinguishable lossy key-generation mode. ELFs were constructed from exponentially-hard DDH, which is plausible in appropriate elliptic curve groups.

**Why obfuscate an ELF?** Using ELFs to instantiate ROs is exactly why they were introduced. However, in prior work they were not obfuscated. To motivate our new approach, note that it seems that ELFs could be useful with the task of “answering decryption queries” in a proof of CCA security for an encryption scheme. Indeed, our strategy is to try all possible answers (there are only polynomially many) and see which one “works.” Yet there is a problem: the hash output used in the challenge ciphertext may no longer look random. To solve this problem, we wrap the ELF in a higher-level program that we obfuscate. This program outputs a programmed point on a special input (used in forming the challenge ciphertext), and otherwise evaluates the ELF.

### 5.2 Partial Instantiation Results

We first give partial instantiations of either $G$ or $H$ for RSA-OAEP under IND-CCA2. Our results use only mild standard model properties of $G$ or $H$. They also use (generalizations of) algebraic properties of RSA proven by Barthe et al. [3] for small enough $e$. For example, using a 2048-bit modulus and encrypting a 128-bit AES key, our results hold for $e = 3$. They may be true for larger $e$; at least, we do not know how they can be disproved. Note that our results first necessitate a separate proof of IND-CPA — the standard model IND-CPA results of Kiltz et al. [76] and Bellare et
al. [15] are not suitable, the first requiring large \( e \) and the second holding only for public-key independent messages.

### 5.2.1 Algebraic Properties of RSA

We first give the (generalizations of) algebraic properties of RSA from Barthe et al. [3] that we use and their parameters. Note that they used these assumptions to analyze security of a zero-redundancy one-round version of RSA-OAEP. We show these are useful for analyzing security of RSA-OAEP more generally.

**Second-input extractability.** Let \( \mathcal{F} = (Kg, Eval, Inv) \) be a trapdoor permutation family with domain \( \{0, 1\}^n \). For \( 1 \leq i \leq j \leq n \), we say \( \mathcal{F} \) is \((\text{blackbox})\ (i, j)\)-second-input-extractable \((\text{BB} (i, j)\text{-SIE})\) if there exists an efficient extractor \( \mathcal{E} \) such that for every \( k \in \mathbb{N} \), every \( f \in [Kg(1^k)] \), and every \( x \in \{0, 1\}^n \), extractor \( \mathcal{E} \) on inputs \( f, f(x), x|_{i+1}^{j} \) outputs \( x \). We often write \( \zeta\text{-SIE} \) instead of \((n - \zeta, n)\text{-SIE}\).

**Common-input extractability.** Let \( \mathcal{F} = (Kg, Eval, Inv) \) be a trapdoor permutation family with domain \( \{0, 1\}^n \). For \( 1 \leq i \leq j \leq n \), we say \( \mathcal{F} \) is \((\text{blackbox})\ (i, j)\)-common-input-extractable if there exists an efficient extractor \( \mathcal{E} \) such that for every \( k \in \mathbb{N} \), every \( f \in [Kg(1^k)] \), and every \( x_1, x_2 \in \text{TDom}(k) \), extractor \( \mathcal{E} \) on inputs \( f, f(x_1), f(x_2) \) outputs \( (x_1, x_2) \) if \( x_1|_{i+1}^{j} = x_2|_{i+1}^{j} \). We often write \( \zeta\text{-CIE} \) instead of \((n - \zeta, n)\text{-CIE}\).

**Comparison to Barthe et al.** Compared to [3], we generalize the notions of SIE and CIE to consider arbitrary runs of consecutive bits. That is, [3] only considers the most significant bits; i.e., \( \zeta\text{-SIE} \) and \( \zeta\text{-CIE} \) in our notation. We also explicitly call the notions \textbf{blackbox} to emphasize the extractor does not make use of the code or random coins of an adversary producing its input. Interestingly, we define analogous notions in Section 3 where this is not the case.
Parameters. Barthe et al. [3] show via the Coppersmith algorithm [44] that RSA is $\zeta$-SIE and $\zeta$-CIE for sufficiently large $\zeta$. Specifically, they show RSA is $\zeta_1$-SIE for $\zeta_1 > n(e-1)/e$, and $\zeta_2$-CIE for $\zeta_2 > n(e^2-1)/e^2$. We show that a generalization to runs of arbitrary consecutive bits holds in Appendix A. Specifically, in Appendix A we show that RSA is $(i,j)$-SIE for $(j-i) > n(e-1)/e$, and $(i,j)$-CIE for $(j-i) > n(e^2-1)/e^2$.

In our partial instantiation results for RSA-OAEP, $j - i$ refers to the length of the redundancy $\zeta$.

5.2.2 Main Results

We now give our main results, namely partial instantiations for RSA-OAEP of either oracle $G$ or $H$. These results refer to IND-CCA security for simplicity, whereas we actually prove PA-RO + IND-CPA.

**Theorem 11** Let $n, \mu, \zeta, \rho$ be integer parameters. Let $G : \mathcal{K}_G \times \{0,1\}^\rho \to \{0,1\}^{\mu + \zeta}$ be a pseudorandom generator and $H : \{0,1\}^{\mu + \zeta} \to \{0,1\}^\rho$ be a RO. Let $F$ be a family of trapdoor permutations with domain $\{0,1\}^n$, where $n = \mu + \zeta + \rho$. Suppose $F$ is one-way, $(\mu + \zeta)$-second input and $(\mu + \zeta)$-common input extractable. Then OAEP$[G,H,F]$ is IND-CCA2 secure. In particular, for any adversary $A$, there is an adversary $D$ and an inverter $I$ such that

$$\text{Adv}_{\text{ind-cca}2}^{\text{OAEP}[G,H,F],A}(k) \leq 2 \cdot \text{Adv}_{\mathcal{F}}^{\text{owf}}(k) + 10 \cdot \text{Adv}_{\mathcal{G},D}^{\text{prg}}(k) + \frac{2p}{2\mu + \zeta} + \frac{4q}{2\zeta},$$

where $q$ is the total number of the decryption queries and $p$ is the total number of RO queries made by $A$.

**Theorem 12** Let $n, \mu, \zeta, \rho$ be integer parameters. Let $H : \mathcal{K}_H \times \{0,1\}^{\mu + \zeta} \to \{0,1\}^\rho$ be a hash function family and $G : \{0,1\}^\rho \to \{0,1\}^{\mu + \zeta}$ be a RO. Let $F$ be a family of trapdoor permutations with domain $\{0,1\}^n$, where $n = \mu + \zeta + \rho$. Suppose $F$ is
\((\rho, \rho + \zeta)\)-second input and \((\rho, \rho + \zeta)\)-common input extractable. Suppose further \(H\) is a \((\mu + \zeta)\)-partial hardcore function for \(F\). Then \(\text{OAEP}[G, H, F]\) is IND-CCA2. In particular, for any adversary \(A = (A_1, A_2)\), there exists an adversary \(B\) such that

\[
\text{Adv}_{\text{OAEP}[G, H, F], A}^{\text{ind-cca2}}(k) \leq 2 \cdot \text{Adv}_{F, H, B}^{\text{phcf}}(k) + \frac{2p}{2^\rho} + \frac{4q}{2^\xi},
\]

where \(q\) the total number of the decryption queries and \(p\) is the total number of RO queries made by \(A\).

The proofs of both theorems follow from below.

**Parameters for RSA-OAEP.** We discuss when our results support RSA-OAEP encryption of an AES key of appropriate length, based on Subsection 5.2.1. The main requirement is encryption exponent \(e = 3\). In this case, with length 2048 bits we can use randomness and message length 128 bits, and for modulus length 4096 we can use randomness length 256. The choice that \(e = 3\) is sometimes used in practice but it is an interesting open problem to extend our results to other common choices such as \(e = 2^{16} + 1\). In particular, it is a reasonable conjecture that results for SIE and CIE hold in this case for the same parameters.

### 5.2.3 Partial Instantiation of \(G\)

We first show how to instantiate \(G\) when modeling \(H\) as a RO. In particular, we show \(\text{OAEP}[G, H, F]\) is IND-CPA + PA-RO when \(G\) is a pseudorandom generator and \(F\) is one-way, (blackbox) \((\mu + \zeta)\)-SIE and \((\mu + \zeta)\)-CIE.

**IND-CPA result.** Under IND-CPA, we show a tight reduction when \(G\) is a pseudorandom generator and \(F\) is one-way and \((\mu + \zeta)\)-SIE. Alternatively, we give can also get IND-CPA security when \(F\) is only partial one-way, but the reduction is lossy. Notes that it is shown in [56] that one-wayness of RSA implies partial one-wayness,
but the reduction is even more lossy, while SIE and CIE unconditionally hold for appropriate parameters.

**Theorem 13** Let \( n, \mu, \zeta, \rho \) be integer parameters. Let \( G : \mathcal{K}_G \times \{0,1\}^\rho \rightarrow \{0,1\}^{\mu+\zeta} \) be a pseudorandom generator and \( H : \{0,1\}^{\mu+\zeta} \rightarrow \{0,1\}^\rho \) be a RO. Let \( \mathcal{F} \) be a family of trapdoor permutations with domain \( \{0,1\}^n \), where \( n = \mu + \zeta + \rho \). Suppose \( \mathcal{F} \) is one-way and \((\mu + \zeta)\)-second input extractable. Then \( \text{OAEP}[G,H,F] \) is IND-CPA. In particular, for any adversary \( A = (A_1,A_2) \), there are an adversary \( D \) and an inverter \( I \) such that

\[
\text{Adv}_{\text{OAEP}[G,H,F],A}^{\text{ind-cca}}(k) \leq 2 \cdot \text{Adv}_{\mathcal{F},I}^{\text{owf}}(k) + 6 \cdot \text{Adv}_{\mathcal{G},D}^{\text{prg}}(k) + \frac{2q}{2^{\mu+\zeta}},
\]

where \( q \) is the total number of RO queries made by \( A \). Furthermore, the running time of \( D \) and \( I \) are about that of \( A \) plus the time to run SIE extractor.

**Proof.** Consider games \( G_1 \)–\( G_6 \) in Figures 5.1–5.2. Each game maintains two independent random oracles RO and \( \overline{\text{RO}} \). Procedure RO maintains a local array \( H \) as follows:

**Procedure RO\((v)\)**

If \( H[v] = \perp \) then \( H[v] \leftarrow \{0,1\}^\rho \)

Return \( H[v] \)

For simplicity, we omit the code of RO, \( \overline{\text{RO}} \) in the games. In each game, we use RO\(_1\) to denote the oracle interface of adversary \( A_1 \) and message samplers \( \mathcal{M}_0, \mathcal{M}_1 \), and we use RO\(_2\) to denote the oracle interface of adversary \( A_2 \). Game \( G_1 \) corresponds to game \( \text{IND-CPA}_{\text{OAEP}[G,H,F]} \). Then

\[
\text{Adv}_{\text{OAEP}[G,H,F],A}^{\text{ind-cca}}(k) \leq 2 \cdot \Pr[G_1(k) \Rightarrow 1] - 1 .
\]

We now explain the game chain. Game \( G_2 \) is identical to game \( G_1 \), except in the encryption of message \( m_b \). Namely, if either adversary \( A_1 \) or message sampler
Games $G_1(k), G_2(k)$
\begin{align*}
  b &\leftarrow \{0, 1\}; \ K_G &\leftarrow \mathcal{K}_G(1^k) \\
  (f, f^{-1}) &\leftarrow \mathcal{K}_G(1^k); \ pk &\leftarrow (K_G, f) \\
  (M_0, M_1, state) &\leftarrow A_1^{RO_1(\cdot)}(1^k, pk) \\
  m_b &\leftarrow M_b^{RO_1(\cdot)}(1^k, pk) \\
  r &\leftarrow \{0, 1\}^\rho; \ x &\leftarrow G(K_G, r) \\
  s &\leftarrow \oplus(m_b||0^k) \\
  If \ H[s] &\neq \bot \ then \quad \text{bad}_1 &\leftarrow \text{true}; \ H[s] &\leftarrow \{0, 1\}^\rho \\
  Else \ H[s] &\leftarrow \{0, 1\}^\rho \\
  z &\leftarrow H[s]; \ t &\leftarrow \oplus r; \ c &\leftarrow f(s||t) \\
  d &\leftarrow A_2^{RO_2(\cdot)}(c, \text{state}) \\
  \text{Return} \ (b = \text{bad}) \\
\end{align*}

Procedure $RO_1(v)$

Return $RO(v)$

Procedure $RO_2(v)$

Return $RO(v)$

Games $G_3(k), G_4(k)$
\begin{align*}
  b &\leftarrow \{0, 1\}; \ K_G &\leftarrow \mathcal{K}_G(1^k) \\
  (f, f^{-1}) &\leftarrow \mathcal{K}_G(1^k); \ pk &\leftarrow (K_G, f) \\
  (M_0, M_1, state) &\leftarrow A_1^{RO_1(\cdot)}(1^k, pk) \\
  m_b &\leftarrow M_b^{RO_1(\cdot)}(1^k, pk) \\
  r &\leftarrow \{0, 1\}^\rho; \ x &\leftarrow G(K_G, r) \\
  s &\leftarrow \oplus(m_b||0^k); \ t &\leftarrow \{0, 1\}^\rho \\
  z &\leftarrow \oplus r; \ H[s] &\leftarrow z; \ c &\leftarrow f(s||t) \\
  d &\leftarrow A_2^{RO_2(\cdot)}(c, \text{state}) \\
  \text{Return} \ (b = d) \\
\end{align*}

Procedure $RO_1(v)$

Return $RO(v)$

Procedure $RO_2(v)$

If $v = s$ then \quad \text{bad}_2 &\leftarrow \text{true}; \ \text{return} \ RO(v) \\

Return $RO(v)$

Figure 5.1: Games $G_1$–$G_4$ in the proof of Theorem 13.

$\mathcal{M}_b$ queries $s$ to their random oracle $RO_1$, it chooses a fresh random value for $H[s]$. Games $G_1$ and $G_2$ are identical-until-bad$_1$, and thus from the Fundamental Lemma of Game-Playing [7],

\[
\Pr [G_1(k) \Rightarrow 1] - \Pr [G_2(k) \Rightarrow 1] \leq \Pr [G_2(k) \text{ sets bad}_1].
\]

Now, consider adversary $B$ attacking the pseudorandom generator $G$ in Figure 5.3. We know that $\text{Adv}^{\text{PRG}}_{G,G_B}(k) = 2 \cdot \Pr [\text{PRG-DIST}^B_G(k) \Rightarrow 1] - 1$. Let $\text{PRG-REAL}^B_G$ be the game identical to game $\text{PRG-DIST}^B_G$ conditioned on $b = 1$, and $\text{PRG-RAND}^B_G$ be the game identical to game $\text{PRG-DIST}^B_G$ conditioned on $b = 0$. Then,

\[
\text{Adv}^{\text{PRG}}_{G,G_B}(k) = \Pr [\text{PRG-REAL}^B_G \Rightarrow 1] - \Pr [\text{PRG-RAND}^B_G \Rightarrow 1].
\]
based on Fundamental Lemma of Game-Playing [7],

Note that $\Pr \left[ \text{PRG-REAL}_G^B \Rightarrow 1 \right] = \Pr \left[ G_2(k) \text{ sets bad}_1 \right]$. Moreover, in the PRG-RAND$_G^B$, the probability that any given of adversary $A$ to its RO equals $s$ is $1/2^{\mu + \zeta}$. Taking a union bound over all queries we have $\Pr[\text{PRG-RAND}_G^B \Rightarrow 1] \leq q/2^{\mu + \zeta}$. Thus

$$\Pr \left[ G_2(k) \text{ sets bad}_1 \right] \leq \text{Adv}^{\text{prg}}_{G,B}(k) + \frac{q}{2^{\mu + \zeta}} .$$

In game $G_3$, we reorder the code of game $G_2$ producing $t$. The change is conservative, meaning that $\Pr[G_2(k) \Rightarrow 1] = \Pr[G_3(k) \Rightarrow 1]$. Game $G_4$ is identical to game $G_3$, except in procedure RO$_2$. Namely, if adversary $A_2$ make a query for $s$, then the oracle lies, calling RO instead. Game $G_3$ and game $G_4$ are identical-until-bad$_2$, thus based on Fundamental Lemma of Game-Playing [7],

$$\Pr \left[ G_3(k) \Rightarrow 1 \right] - \Pr \left[ G_4(k) \Rightarrow 1 \right] \leq \Pr \left[ G_4(k) \text{ sets bad}_2 \right] .$$

![Figure 5.2: Games $G_5, G_6$ in the proof of Theorem 13.](image-url)
Algorithm $B(K_G, x)$

$(f, f^{-1}) \leftarrow K_{g(1^k)}$; $\text{out} \leftarrow 0$

$pk \leftarrow (K_G, f); b \leftarrow \{0, 1\}$

$(M_0, M_1, \text{state}) \leftarrow A^{ROSim_1(\cdot)}_1(1^k, pk)$

$m_b \leftarrow M_b^{ROSim_1(\cdot)}(1^k, pk)$

$s \leftarrow x \oplus (m_b || 0^6)$

If $H[s] \neq \bot$ then $\text{out} \leftarrow 1$

Return $\text{out}$

---

Procedure $ROSim_1(v)$

If $H[v] = \bot$ then

$H[v] \leftarrow \{0, 1\}^\rho$

Return $H[v]$

---

Figure 5.3: Adversary $B$ in the proof of Theorem 13.

Algorithm $C(K_G, x)$

$(f, f^{-1}) \leftarrow K_{g(1^k)}$; $\text{out} \leftarrow 0$

$pk \leftarrow (K_G, f); b \leftarrow \{0, 1\}$

$(M_0, M_1, \text{state}) \leftarrow A^{ROSim_1(\cdot)}_1(1^k, pk)$

$m_b \leftarrow M_b^{ROSim_1(\cdot)}(1^k, pk)$

$s \leftarrow x \oplus (m_b || 0^6)$

$c \leftarrow f(s || t)$

Run $A^{ROSim_2(\cdot)}_2(c, \text{state})$

Return $\text{out}$

---

Procedure $ROSim_1(v)$

If $H[v] = \bot$ then $H[v] \leftarrow \{0, 1\}^\rho$

Return $H[v]$

Procedure $ROSim_2(v)$

If $v = s$ then

$\text{out} \leftarrow 1$; Halt run of $A_2$

If $H[v] = \bot$ then $H[v] \leftarrow \{0, 1\}^\rho$

Return $H[v]$

---

Figure 5.4: Adversary $C$ in the proof of Theorem 13.

Consider the adversary $C$ attacking the pseudorandom generator $G$ in Figure 5.4.

Let $\text{PRG-REAL}^C_G$ be the game identical to game $\text{PRG-DIST}^C_G$ condition on $b = 1$, and $\text{PRG-RAND}^C_G$ be the game identical to game $\text{PRG-DIST}^C_G$ condition on $b = 0$. Then,

$$\text{Adv}^{\text{prg}}_{G,C}(k) = \Pr[\text{PRG-REAL}^C_G \Rightarrow 1] - \Pr[\text{PRG-RAND}^C_G \Rightarrow 1].$$

Note that $\Pr[\text{PRG-REAL}^C_G \Rightarrow 1] = \Pr[G_4(k) \text{ sets } \text{bad}_2]$. Let $\text{Ext}$ be the second-input extractor for $\mathcal{F}$. To bound the probability that $\text{PRG-RAND}^C_G$ outputs 1, we construct an inverter $I$ attacking $\mathcal{F}$ in Figure 5.5. Note that if adversary $A_2$
Algorithm $I(f,c)$

$b \leftarrow \{0,1\}$; $\text{out} \leftarrow \perp$

$K_G \leftarrow K_G(1^k)$; $pk \leftarrow (K_G, f)$

$(M_0,M_1,\text{state}) \leftarrow s \text{A}^\text{ROSIM}_1(\cdot)(1^k,pk)$

$m_b \leftarrow \text{M}^\text{ROSIM}_1(\cdot)b(1^k,pk)$

Run $A^\text{ROSIM}_2(\cdot)(c,\text{state})$

Return out

Procedure $\text{ROSIM}_1(v)$

If $H[v] = \perp$ then $H[v] \leftarrow \{0,1\}^\rho$

Return $H[v]$

Procedure $\text{ROSIM}_2(v)$

$t \leftarrow \text{Ext}(f,c,v)$

If $t \neq \perp$ then $\text{out} \leftarrow t$; Halt run of $A_2$

If $H[v] = \perp$ then $H[v] \leftarrow \{0,1\}^\rho$

Return $H[v]$

Figure 5.5: Inverter $I$ in the proof of Theorem 13.

queries $s$ then inverter $I$ can invert challenge $c$ using extractor $\text{Ext}$. Hence, we have

$$\Pr\left[\text{PRG-RAND}_G^C \Rightarrow 1\right] \leq \text{Adv}^{\text{owf}}_{\mathcal{F},I}(k).$$

Thus,

$$\Pr[G_4(k) \text{ sets } \text{bad}_2] = \text{Adv}^{\text{prg}}_{G,C}(k) + \text{Adv}^{\text{owf}}_{\mathcal{F},I}(k).$$

Next, game $G_5$ is identical to game $G_4$, except it uses a uniformly random $x$ in the encryption phase instead of a pseudorandom value $G(K,r)$. Consider adversary $D$ as shown in Figure 5.6. We have

$$\Pr[G_4(k) \Rightarrow 1] - \Pr[G_5(k) \Rightarrow 1] \leq \text{Adv}^{\text{prg}}_{G,D}(k).$$

In game $G_6$, we reorder the code of game $G_5$ producing $s$. The change is conservative, meaning that $\Pr[G_5(k) \Rightarrow 1] = \Pr[G_6(k) \Rightarrow 1]$. Note that $\Pr[G_6(k) \Rightarrow 1] = 1/2$, since the ciphertexts are independent of the bit $b$. Assuming that the advantage of adversary $D$ is greater than the advantage of $B$ and $C$, we have

$$\text{Adv}^{\text{ind-}\text{cpa}}_{\text{OAEP}[G,H,\mathcal{F}],A}(k) \leq 2 \cdot \text{Adv}^{\text{owf}}_{\mathcal{F},I}(k) + 6 \cdot \text{Adv}^{\text{prg}}_{G,D}(k) + \frac{2q_2}{2^{\mu+\zeta}}.$$ 

This completes the proof.
PA-RO result. We show RSA-OAEP is PA-RO when modeling \( \mathcal{H} \) as a RO if \( \mathcal{G} \) is a pseudorandom generator and \( \mathcal{F} \) is both second-input extractable and common-input extractable.

**Theorem 14** Let \( n, \mu, \zeta, \rho \) be integer parameters. Let \( \mathcal{G} : \mathcal{K}_G \times \{0,1\}^\rho \rightarrow \{0,1\}^{\mu+\zeta} \) be a pseudorandom generator and \( \mathcal{H} : \{0,1\}^{\mu+\zeta} \rightarrow \{0,1\}^\rho \) be a RO. Let \( \mathcal{F} \) be a family of trapdoor permutations with domain \( \{0,1\}^n \), where \( n = \mu + \zeta + \rho \). Suppose \( \mathcal{F} \) is \((\mu + \zeta)\)-second input and \((\mu + \zeta)\)-common input extractable. Then OAEP[\( \mathcal{G}, \mathcal{H}, \mathcal{F} \)] is PA-RO secure. In particular, for any adversary \( A \), there exists an adversary \( D \) and an extractor \( \text{Ext} \) such that

\[
\text{Adv}_{\text{PA-RO}}^{\text{OAEP}[\mathcal{G}, \mathcal{H}, \mathcal{F}], A, \text{Ext}}(k) \leq 2 \cdot \text{Adv}_{\text{PRG}}^{\mathcal{G}, D}(k) + \frac{2q}{2^{\zeta}},
\]

where \( q \) is the total number of the extraction queries made by \( A \). Furthermore, the running time of \( D \) is about that of \( A \) and the running time of \( \text{Ext} \) is about that of SIE and CIE extractors.

**Proof.** Let \( \text{Ext}_1 \) be the second-input extractor and \( \text{Ext}_2 \) be the common-input extractor for \( \mathcal{F} \). For any adversary \( A \), we define the extractor \( \text{Ext} \) as shown in

```plaintext
Algorithm D(K_G, x)
(f, f^{-1}) \leftarrow Kg(1^k)
pk \leftarrow (K_G, f) \colon b \leftarrow \{0, 1\}
(M_0, M_1, state) \leftarrow A_{\text{ROSim}1}(1^k, pk)
m_b \leftarrow M_{\text{ROSim}1}(1^k, pk)
(s, t) \leftarrow x \oplus (m_b \mid |0^z|)
(\mathcal{M}_0, \mathcal{M}_1, state) \leftarrow A_{\text{ROSim}1}(1^k, pk, b)
(b', c) \leftarrow f(s) \mid |t|
Return (b = b')
```

```
Procedure ROSIM1(v)
If \( H[v] = \perp \) then \( H[v] \leftarrow \{0, 1\}^\rho \)
Return \( H[v] \)
```

```
Procedure ROSIM2(v)
If \( v = s \) then \( H[v] \leftarrow \{0, 1\}^\rho \)
If \( H[v] = \perp \) then \( H[v] \leftarrow \{0, 1\}^\rho \)
Return \( H[v] \)
```

Figure 5.6: Adversary \( D \) in the proof of Theorem 13.
Algorithm $\text{Ext}^H(x, h, c, c_i, pk)$

$(f, K) \leftarrow pk$

For $j = 1$ to $|x|$ do

$s_i || t_i \leftarrow \text{Ext}_1(f, c, x[j])$

If $t_i \neq \perp$ then

$r_i \leftarrow h[j] \oplus t_i;$

$m_i \leftarrow x[j] \oplus G(K, r_i)$

If $m_i |_\zeta = 0^\zeta$ then return $m_i |_\mu$

For $j = 1$ to $|c|$ do

$(s_i || t_i, s_i || t'_i) \leftarrow \text{Ext}_2(f, c, c[j])$

If $s_i \neq \perp$ then

$r_i \leftarrow H(s_i) \oplus t_i;$

$m_i \leftarrow s_i \oplus G(K, r_i)$

If $m_i |_\zeta = 0^\zeta$ then return $m_i |_\mu$

Return $\perp$

Figure 5.7: PA-RO extractor $\text{Ext}$ in the proof of Theorem 14.

Figure 5.7. Now, we bound the advantage of adversary $A$ in distinguishing between the decryption algorithm and the extractor $\text{Ext}$.

Assume adversary $A$ makes $q$ extract queries. Let $c_i$ be the $i$-th such query. Let’s denote by $s_i$ and $t_i$ the last $(\mu + \zeta)$-bits and first $\rho$-bits of $f^{-1}(c_i)$, respectively. We define $S$ to be the event that game $\text{PA-RO}_{\text{OAEP}[G, H, F]}^A, \text{Ext}(k)$ outputs 1. We also define $E$ to be the event that adversary $A$ or the encryption oracle query for the value $s_i$ to random oracle $H$, for all $i \in [q]$. Then

$$\text{Adv}_{\text{PA-RO}_{\text{OAEP}[G, H, F]}^A, \text{Ext}}(k) = 2 \cdot \left( \Pr[S \land E] + \Pr[S \land \overline{E}] \right) - 1.$$ 

We know, if event $E$ happens then extractor $\text{Ext}$ can use the second-input extractor or the common-input extractor to recover plaintext $m$. Therefore, we have $\Pr[S \mid E] = 1/2$. Thus,

$$\text{Adv}_{\text{PA-RO}_{\text{OAEP}[G, H, F]}^A, \text{Ext}}(k) \leq \Pr[E] + 2 \cdot \Pr[S \land \overline{E}] - 1.$$ 

Next, we bound the probability that ciphertext $c_i$ with no prior random oracle query $s_i$, is valid. We know when event $\overline{E}$ happens, there exists at least a ciphertext
With no prior query $s_i$ to random oracle $\mathcal{H}$. Let $C$ be the set of such ciphertexts. Note that extractor $\text{Ext}$ always outputs $\bot$ on such ciphertexts. Let $T$ be the event where there exists at least a valid ciphertext $c_i \in C$. Then

$$\Pr[S \land E] = \Pr[S \land \overline{E} \land T] + \Pr[S \land \overline{E} \land \overline{T}] \leq \Pr[\overline{E} \land T] + \Pr[\overline{E} \land \overline{T}] \cdot \Pr[S \mid \overline{E} \land \overline{T}].$$

Note that $\Pr[S \mid \overline{E} \land \overline{T}] = 1/2$, since the outputs of $\text{Ext}$ and decryption algorithm $\text{Dec}$ are always equal. Moreover, we have $\Pr[\overline{E} \land \overline{T}] \leq \Pr[\overline{E}]$. Therefore,

$$\text{Adv}_{\text{OAOEP}[G,H,F],A,\text{Ext}}^{\text{pa-ro}}(k) \leq 2 \cdot \Pr[\overline{E} \land T].$$

We know the challenge ciphertext $c_i$ is valid if and only if there exists a plaintext $m_i$ such that $G(K,r_i) \oplus s_i = m_i \parallel 0^\zeta$, where $r_i = H(s_i) \oplus t_i$. Moreover, when there is no prior random oracle query $s_i$, $r_i$ looks uniformly random to adversary $A$. Therefore, if event $T \mid \overline{E}$ happens then there exists a ciphertext $c_i$ such that the first $\zeta$-bits of $G(K,r_i)$ and $s_i$ are equal, where $r_i$ is chosen uniformly at random. Consider adversary $D$ attacking the pseudorandom generator $G$ in Figure 5.8. Note that when adversary $D$ is in the real game then it simulates the PA-RO game for the adversary $A$. On the other hand, when adversary $D$ is in the ideal game then the first $\zeta$-bits of $s_i$ and $x$ are equal with probability $1/2^\zeta$. Taking a union bound, we get

$$\text{Adv}_{G,D}^{\text{prg}}(k) \geq \Pr[T \land \overline{E}] - \frac{q}{2^\zeta}.$$ 

Hence, we have $\Pr[T \land \overline{E}] \leq \text{Adv}_{G,D}^{\text{prg}}(k) + q \cdot 2^{-\zeta}$. Summing up,

$$\text{Adv}_{\text{OAOEP}[G,H,F],A,\text{Ext}}^{\text{pa-ro}}(k) \leq 2 \cdot \text{Adv}_{G,D}^{\text{prg}}(k) + \frac{2q}{2^\zeta}.$$ 

This completes the proof.
5.2.4 Partial Instantiation of H

Now, we instantiate the hash function $H$ when modeling $G$ as a RO. In particular, we show $\text{OAEP}[G, H, F]$ is IND-CPA + PA-RO when $H$ is a special type of hardcore function and $F$ is one-way, second-input and common-input extractable. Note that Boneh [30] previously showed a simplified RSA-OAEP with one Feistel round $G$ is IND-CCA2 secure and Barthe et al. [3] showed such a scheme does not even need redundancy, but these proofs can not applied to the case of $H$ as a cryptographic hash function.

IND-CPA result. Under IND-CPA, we show a tight reduction when $H$ is a $(\mu + \zeta)$-partial hardcore function for $F$. In particular, it is plausible for $H$ as a computational randomness extractor [47] and that $F$ is RSA in the common setting $\rho = k$ (e.g., $\rho = 128$ for modulus length $n = 2048$), since Coppersmith’s technique fails.
Theorem 15  Let \( n, \mu, \zeta, \rho \) be integer parameters. Let \( \mathcal{H} : \mathcal{K}_H \times \{0,1\}^{\mu+\zeta} \rightarrow \{0,1\}^{\rho} \) be a hash function family and \( \mathcal{G} : \{0,1\}^{\rho} \rightarrow \{0,1\}^{\mu+\zeta} \) be a RO. Let \( \mathcal{F} \) be a family of trapdoor permutations with domain \( \{0,1\}^n \), where \( n = \mu + \zeta + \rho \). Suppose \( \mathcal{H} \) is a \((\mu + \zeta)\)-partial hardcore function for \( \mathcal{F} \). Then \( \text{OAEP}[\mathcal{G}, \mathcal{H}, \mathcal{F}] \) is IND-CPA. In particular, for any adversary \( A = (A_1, A_2) \), there exists an adversary \( B \) such that

\[
\text{Adv}^{\text{ind-CPA}}_{\text{OAEP}[\mathcal{G}, \mathcal{H}, \mathcal{F}], A}(k) \leq 2 \cdot \text{Adv}^{\text{phcf}}_{\mathcal{F}, \mathcal{H}, B}(k) + \frac{2q}{2^\rho},
\]

where \( q \) is the total number of RO queries made by \( A \). The running time of \( B \) is about that of \( A \).

Proof. Consider games \( G_1-G_4 \) in Figure 5.9. Each game maintains two independent random oracles \( \text{RO} \) and \( \overline{\text{RO}} \). Procedure \( \text{RO} \) maintains a local array \( G \) as follows:

**Procedure** \( \text{RO}(v) \)

If \( G[v] = \perp \) then \( G[v] \leftarrow \{0,1\}^{\mu+\zeta} \)

Return \( G[v] \)

For simplicity, we omit the code of \( \text{RO}, \overline{\text{RO}} \) in the games. In each game, we use \( \text{RO}_1 \) to denote the oracle interface of adversary \( A_1 \) and message samplers \( \mathcal{M}_0, \mathcal{M}_1 \), and we use \( \text{RO}_2 \) to denote the oracle interface of adversary \( A_2 \). Game \( G_1 \) corresponds to game \( \text{IND-CPA}^A_{\text{OAEP}[\mathcal{G}, \mathcal{H}, \mathcal{F}]} \). Then

\[
\text{Adv}^{\text{ind-CPA}}_{\text{OAEP}[\mathcal{G}, \mathcal{H}, \mathcal{F}], A}(k) \leq 2 \cdot \Pr[G_1(k) \Rightarrow 1] - 1.
\]

We now explain the game chain. Game \( G_2 \) is identical to game \( G_1 \), except in the encryption of message \( m_b \), if either adversary \( A_1 \) or message sampler \( \mathcal{M}_b \) queried \( r \) to their random oracle \( \text{RO}_1 \), then it chooses a fresh random value for \( G[r] \). Games \( G_1 \) and \( G_2 \) are identical-until-\text{bad}_1, and thus from the Fundamental Lemma of Game-Playing [7],

\[
\Pr\left[G_1(k) \Rightarrow 1\right] - \Pr\left[G_2(k) \Rightarrow 1\right] \leq \Pr\left[G_2(k) \text{ sets } \text{bad}_1\right].
\]
Moreover, the probability is $1/2^\rho$ that any given RO query of $A_1$ or message samplers $\mathcal{M}_0, \mathcal{M}_1$ equals $r$. Let $q_1$ be the number of random oracle query that $A_1$ and $\mathcal{M}_0, \mathcal{M}_1$ make. Taking a union bound over all queries, we have $\Pr [ G_2(k) \text{ sets } \text{bad}_1 ] \leq q_1/2^\rho$. In game $G_3$, we reorder the code of game $G_2$ producing $s$. The change is conservative, meaning that $\Pr[G_2(k) \Rightarrow 1] = \Pr[G_3(k) \Rightarrow 1]$. Game $G_4$ is identical to game $G_3$, except in procedure $\text{RO}_2$, if adversary $A_2$ queries $r$, then the oracle lies, calling $\overline{\text{RO}}$ instead. Game $G_3$ and game $G_4$ are identical-until-$\text{bad}_2$, and based on Fundamental Lemma of Game-Playing [7],

$$
\Pr [ G_3(k) \Rightarrow 1 ] - \Pr [ G_4(k) \Rightarrow 1 ] \leq \Pr [ G_4(k) \text{ sets } \text{bad}_2 ]
$$
Algorithm $B(K_H, f, c, t, z)$

- $pk \leftarrow (K_H, f)$ ; $b \leftarrow s \{0,1\}$
- $(\mathcal{M}_0, \mathcal{M}_1, state) \leftarrow A_1^{\text{ROSim}_1}(1^k, pk)$
- $m_b \leftarrow \mathcal{M}_b^{\text{ROSim}_1}(1^k, pk)$
- $r \leftarrow t \oplus z$ ; $out \leftarrow \bot$
- Run $A_2^{\text{ROSim}_2}(c, state)$
- Return $out$

Procedure $\text{ROSim}_1(v)$

- If $\mathcal{G}[v] = \bot$ then $\mathcal{G}[v] \leftarrow s \{0,1\}$
- Return $\mathcal{G}[v]$

Procedure $\text{ROSim}_2(v)$

- If $v = r$ then $out \leftarrow 1$ ; Halt $A$
- If $\mathcal{G}[v] = \bot$ then $\mathcal{G}[v] \leftarrow s \{0,1\}$
- Return $\mathcal{G}[v]$

Figure 5.10: Adversary $B$ in the proof of Theorem 15.

Now, consider adversary $B$ attacking partial hardcore function $H$ in Figure 5.10. We know that

$$\text{Adv}^{\text{phcf}}_{\mathcal{F}, \mathcal{H}, B}(k) = 2 \cdot \Pr \left[ \text{PHCF-DIST}^B_{\mathcal{F}, \mathcal{H}}(k) \Rightarrow 1 \right] - 1.$$

Let PHCF-REAL be the game identical to game PHCF-DIST conditioned on $b = 1$, and PHCF-RAND be the game identical to game PHCF-DIST conditioned on $b = 0$. Then,

$$\text{Adv}^{\text{phcf}}_{\mathcal{F}, \mathcal{H}, B}(k) = \Pr \left[ \text{PHCF-REAL}^B_{\mathcal{F}, \mathcal{H}} \Rightarrow 1 \right] - \Pr \left[ \text{PHCF-RAND}^B_{\mathcal{F}, \mathcal{H}} \Rightarrow 1 \right].$$

Note that $\Pr \left[ \text{PHCF-REAL}^B_{\mathcal{F}, \mathcal{H}} \Rightarrow 1 \right] = \Pr \left[ G_4(k) \text{ sets bad}_1 \right]$. Moreover, in game PHCF-RAND, the probability is $1/2^\rho$ that any given RO queried by adversary $A_2$ equals $r$. Let $q_2$ be the number of queries that $A_2$ makes. Taking a union bound we have $\Pr[\text{PHCF-RAND}^B_{\mathcal{F}, \mathcal{H}} \Rightarrow 1] \leq q_2/2^\rho$. Thus

$$\Pr \left[ G_4(k) \text{ sets bad}_1 \right] \leq \text{Adv}^{\text{phcf}}_{\mathcal{F}, \mathcal{H}, B}(k) + \frac{q_2}{2^\rho}.$$

Note that, $\Pr[G_4(k) \Rightarrow 1] = 1/2$, since the distribution of the ciphertexts are completely independent of $b$. Summing up,

$$\text{Adv}^{\text{ind-cpa}}_{\text{OAEP}[\mathcal{G}, \mathcal{H}, \mathcal{F}], A}(k) \leq 2 \cdot \text{Adv}^{\text{phcf}}_{\mathcal{F}, \mathcal{H}, B}(k) + \frac{2q}{2^\rho}.$$
This completes the proof.

**PA-RO result.** We show another partial instantiation result modeling $G$ as a RO. Namely, we show RSA-OAEP is PA-RO if $F$ is second-input extractable, and common-input extractable. Note that this does not require any assumption on $H$.

**Theorem 16** Let $n, \mu, \zeta, \rho$ be integer parameters. Let $\mathcal{H} : \{0,1\}^{\mu + \zeta} \rightarrow \{0,1\}^\rho$ be a hash function family and $\mathcal{G} : \mathcal{K}_G \times \{0,1\}^\rho \rightarrow \{0,1\}^{\mu + \zeta}$ be a RO. Let $\mathcal{F}$ be a family of trapdoor permutations with domain $\{0,1\}^n$, where $n = \mu + \zeta + \rho$. Suppose $\mathcal{F}$ is $(\rho, \rho + \zeta)$-second input and $(\rho, \rho + \zeta)$-common input extractable. Then $\text{OAEP}[G, H, F]$ is PA-RO secure. In particular, for any adversary $A$, there exists an extractor $\text{Ext}$ such that,

$$\text{Adv}_{\text{PA-RO OAEP}[G, H, F], A, \text{Ext}}(k) \leq \frac{2q}{2^\zeta},$$

where $q$ is the total number of the extract queries made by $A$. The running time of $\text{Ext}$ is about that of $\text{SIE}$ and $\text{CIE}$ extractors.

**Proof.** Let $\text{Ext}_1$ be the second-input extractor and $\text{Ext}_2$ be the common-input extractor for $\mathcal{F}$. For any adversary $A$, we define the extractor $\text{Ext}$ as shown in Figure 5.11. Now, we bound the advantage of adversary $A$ in distinguishing between the decryption algorithm and the extractor $\text{Ext}$.

Assume $A$ makes $q$ extract queries. Let $c_i$ be the $i$-th such query. Let $r_i$ be the randomness used in creating ciphertext $c_i$. We define $S$ to be the event that game $\text{PA-RO}_{\text{OAEP}[G, H, F]}^A(k)$ outputs 1. We also define $E$ as the event where $A$ or the encryption oracle query $r_i$ to $\mathcal{G}$, for all $i \in [q]$. Then,

$$\text{Adv}_{\text{PA-RO OAEP}[G, H, F], A, \text{Ext}}(k) = 2 \cdot (\Pr[S \land E] + \Pr[S \land \overline{E}]) - 1.$$

We know when the event $E$ happens $\text{Ext}$ can use the second-input extractor or the common-input extractor to recover plaintext $m$. Therefore, we have $\Pr[S \mid E] = 1/2$. 

82
Algorithm $\text{Ext}^H(r, g, c, c_i, pk)$

$(f, K) \leftarrow pk$

For $j = 1$ to $|r|$ do
  $y_i \leftarrow \text{Ext}_1(f, c_i, g[j]|\zeta)$
  If $y_i \neq \bot$ then
    $m_i \leftarrow y_i|\mu+\zeta \oplus g[j]$
  If $m_i|\zeta = 0^\zeta$ then return $m_i|\mu$

For $j = 1$ to $|c|$ do
  $(y_i, y_i') \leftarrow \text{Ext}_2(f, c_i, c[j])$
  $s_i \leftarrow y_i|\mu+\zeta$; $t_i \leftarrow y_i|\rho$
  If $y_i \neq \bot$ then
    $r_i \leftarrow H(K, s_i) \oplus t_i$; $m_i \leftarrow s_i \oplus G(r_i)$
  If $m_i|\zeta = 0^\zeta$ then return $m_i|\mu$

Return $\bot$

Figure 5.11: PA-RO extractor $\text{Ext}$ in the proof of Theorem 16.

Thus,

$$\text{Adv}_{\text{OAEP}[G, H, F], A, \text{Ext}}^{\text{pa-ro}}(k) \leq \Pr[E] + 2 \cdot \Pr[S \land \overline{E}] - 1.$$ 

Next, we bound the probability that ciphertexts $c_i$ with no prior random oracle query $r_i$ is valid. We know when event $\overline{E}$ happens, there exists at least one ciphertext $c_i$ with no prior query $r_i$ to random oracle $G$. Let $C$ be the set of such ciphertexts. Note that extractor $\text{Ext}$ always output $\bot$ on such ciphertexts. Let $T$ be the event where there exists at least a valid ciphertext $c_i \in C$. Then,

$$\Pr[S \land \overline{E}] = \Pr[S \land \overline{E} \land T] + \Pr[S \land \overline{E} \land \overline{T}]$$

$$\leq \Pr[E \land T] + \Pr[E \land \overline{T}] \cdot \Pr[S \land \overline{E} \land T].$$

Note that $\Pr[S \land \overline{E} \land T] = 1/2$, since the outputs of extractor $\text{Ext}$ and decryption algorithm $\text{Dec}$ are always equal. Moreover, we have $\Pr[E \land \overline{T}] \leq \Pr[E]$. Therefore,

$$\text{Adv}_{\text{OAEP}[G, H, F], A, \text{Ext}}^{\text{pa-ro}}(k) \leq 2 \cdot \Pr[E \land T].$$
We know the challenge ciphertext $c_i$ is valid if and only if there exists a plaintext $m_i$ such that $G(r_i) \oplus s_i = m_i \| 0^\zeta$. In other words, the challenge ciphertext $c_i$ is a valid ciphertext when $G(r_i)|_\zeta \oplus s_i|_\zeta = 0^\zeta$. Since $r_i$ was not queried, the ciphertext $c_i$ is valid with probability $2^{-\zeta}$. Taking a union bound over all extract queries, we get

$$\text{Adv}^\text{par-RO}_{\text{OAEP}[G,H,F],A,\text{Ext}}(k) \leq \frac{2q}{2^\zeta}.$$ 

This completes the proof.

### 5.3 XOR Assumptions on RSA

Here, we give classes of novel assumptions on RSA (and trapdoor permutations in general), which are stronger than one-wayness and needed for RSA-OAEP $s$-clear.

**XOR-IND.** Our first class of assumptions speaks to the fact that addition or XOR operations “break up” the multiplicative structure of RSA. Indeed, in a related context of arithmetic progressions on $\mathbb{Z}_N$ we have seen formal evidence of this [78, 94]. It is interesting for future work to give formal evidence in our case as well. Let $F = (K_g, \text{Eval}, \text{Inv})$ be a trapdoor permutation family with domain $\text{TDom}$. Let $G : K_G \times \text{TDom} \rightarrow \text{GRng}$ be a function family. For $\text{ATK} \in \{\text{IND0}, \text{IND1}, \text{IND2}\}$, we associate the experiment in Figure 5.12, for every $k \in \mathbb{N}$. Define the xor-atk advantage of $A$ against $F$ with the hint function family $G$

$$\text{Adv}^\text{xor-atk}_{F,G,A}(k) = 2 \cdot \text{Pr} \left[ \text{XOR-ATK}^{A}_{F,G}(k) \Rightarrow 1 \right] - 1.$$ 

If $\text{atk} = \text{ind0}$, then $O = \epsilon$. We say that $F$ is XOR-IND0 with respect to hint function family $G$ if for every PPT attacker $A$, $\text{Adv}^\text{xor-ind0}_{F,G,A}(k)$ is negligible in $k$. Similarly, if $\text{atk} = \text{ind1}$, then $O = C$, where $C$ is a relation checker oracle that on input $y_1, y_2$
Game XOR-ATK\(_k^2\),\(G\)\(k\)
\begin{align*}
b &\leftarrow \{0, 1\} ; \ (f, f^{-1}) \leftarrow Kg(1^k) \\
K_G &\leftarrow K_G(1^k) ; \ x \leftarrow \text{TDom}(k) \\
(state, z) &\leftarrow A_1(f, K_G, G(K_G, x)) \\
y_0 &\leftarrow f(x) ; \ y_1 \leftarrow f(x \oplus z) \\
b' &\leftarrow A_2^O(state, y_b) \\
\text{Return } (b = b')
\end{align*}

Figure 5.12: Games to define XOR-ATK security.

and \(\omega\) work as follows:

\[
C(y_1, y_2, \omega) = \begin{cases} 
1 & \text{if } \omega = f^{-1}(y_1) \oplus f^{-1}(y_2) \\
0 & \text{otherwise}
\end{cases}
\]

Similarly, if \(\text{atk} = \text{ind2}_\ell\), then \(\mathcal{O} = \mathcal{V}_\ell\), where \(\mathcal{V}_\ell\) is an \(\ell\)-bit image verifier oracle that on input \(y\) works as follows:

\[
\mathcal{V}_\ell(y) = \begin{cases} 
1 & \text{if } \exists x : y = G(K_G, x)|_\ell \\
0 & \text{otherwise}
\end{cases}
\]

Note that adversary \(A\) is not allowed to query for the challenge to the image verifier oracle \(\mathcal{V}_\ell\). We say that \(\mathcal{F}\) is XOR-IND1 (resp. XOR-IND2\(_\ell\)) with respect to hint function family \(G\) if for every PPT attacker \(A\), \(\text{Adv}_{\mathcal{F},G,A}^{\text{xor-ind1}}(k)\) (resp. \(\text{Adv}_{\mathcal{F},G,A}^{\text{xor-ind2}_\ell}(k)\)) is negligible in \(k\).

Observe that the hint is crucial, as otherwise the assumption would trivially hold. In our results, \(G\) is a PRG. In this case, we show that \(G\) is also a HCF function for \(\mathcal{F}\). In other words, the assumption in our use-case can be viewed an extension of the classical notion of HCF — \(G\) is “robust” not in the sense of [58], but in the sense that the view of the adversary is also indistinguishable given \(\mathcal{F}\) applied to either the real input or related one. Note that not all hardcore functions have this property, even
when $F$ is partial one-way. For example, consider a hardcore function $G$ that reveals first bit of its input $x$. Then if a partial one-way function $F$ also reveals the first bit of $x$, XOR-indistinguishability clearly does not hold.

**Theorem 17** Let $F$ be a family of one-way trapdoor permutations with domain $TDom$. Suppose $G : K_G \times TDom \to GRng$ is a pseudorandom generator and $F$ is XOR-IND0 with respect to hint function family $G$. Then $G$ is a hardcore function for $F$ on the uniform distribution. In particular, for any adversary $A$, there are adversaries $B, C$ such that

$$\text{Adv}_{F,G,A}^{\text{hcf}}(k) \leq 2 \cdot \text{Adv}_{F,G,B}^{\text{xor-ind0}}(k) + 2 \cdot \text{Adv}_{G, C}^{\text{prg}}(k).$$

The running time of $B$ and $C$ are about that of $A$.

**Proof.** Consider games $G_1$–$G_3$ in Figure 5.13. We now explain the game chain. Game $G_1$ corresponds to game HCF-DIST$_{F,G}^{A,U}(k)$. Game $G_2$ is identical to game $G_1$, except we are using completely random $y$ instead of using the pseudorandom value $f(x)$. Consider adversary $B$ as shown in Figure 5.14. Note that adversary $B$ simulate games $G_1, G_2$ with respect to its inputs. It returns 1 if adversary $A$ can correctly guess the simulated challenge bit $b$, and returns 0 otherwise. Hence,

$$\Pr[G_1 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1] \leq \text{Adv}_{F,G,B}^{\text{xor-ind0}}(k).$$

Next, game $G_3$ is identical to game $G_2$, except we are using completely random $g_0$ instead of using the pseudorandom value $G(K_G, x)$. Consider adversary $C$ as shown in Figure 5.14. Note that adversary $C$ simulate games $G_2, G_3$ with respect to its inputs. It returns 1 if adversary $A$ can correctly guess the simulated challenge bit $b$, and returns 0 otherwise. Hence,

$$\Pr[G_2 \Rightarrow 1] - \Pr[G_3 \Rightarrow 1] \leq \text{Adv}_{G,C}^{\text{prg}}(k).$$

86
Games $G_1(k), G_2(k)$

- $b \leftarrow \{0, 1\}$
- $K_G \leftarrow sK_G(1^k)$
- $(f, f^{-1}) \leftarrow Kg(1^k)$
- $x \leftarrow \text{TDom}(k)$
- $g_0 \leftarrow G(K_G, x); g_1 \leftarrow \text{GRng}(k)$
- $y \leftarrow f(x); y \leftarrow s\text{TDom}(k)$
- $b' \leftarrow A(K_G, f, y, g_0)$
- Return $(b = b')$

Games $G_3(k)$

- $b \leftarrow \{0, 1\}$
- $K_G \leftarrow sK_G(1^k)$
- $(f, f^{-1}) \leftarrow Kg(1^k)$
- $x \leftarrow s\text{TDom}(k)$
- $g_0 \leftarrow \text{GRng}(k); g_1 \leftarrow s\text{GRng}(k)$
- $y \leftarrow s\text{TDom}(k)$
- $b' \leftarrow A(K_G, f, y, g_0)$
- Return $(b = b')$

Figure 5.13: Games $G_1$–$G_3$ in the proof of Theorem 17.

Algorithm $B_1(f, K_G, G(K_G, x))$

- state $\leftarrow (f, K_G, G(K_G, x))$
- $z \leftarrow \text{TDom}(k)$
- Return (state, z)

Algorithm $B_2(state, y)$

- $b \leftarrow \{0, 1\}$
- $(f, K_G, G(K_G, x)) \leftarrow \text{state}$
- $g_0 \leftarrow G(K_G, x)$
- $g_1 \leftarrow \text{GRng}(k)$
- $b' \leftarrow A(K_G, f, y, g_0)$
- Return $(b = b')$

Algorithm $C(K_G, g_0)$

- $b \leftarrow \{0, 1\}$
- $(f, f^{-1}) \leftarrow Kg(1^k)$
- $g_1 \leftarrow \text{GRng}(k)$
- $y \leftarrow \text{TDom}(k)$
- $b' \leftarrow A(K_G, f, y, g_0)$
- Return $(b = b')$

Figure 5.14: Adversary $B$ (left) and adversary $C$ (right) in the proof of Theorem 17.

Note that $\Pr[G_3(k) \Rightarrow 1] = 1/2$, since $y, g_0$ and $g_1$ are uniformly random. Summing up,

$$\text{Adv}^{\text{hcf}}_{\mathcal{F}, \mathcal{G}, U, A}(k) \leq 2 \cdot \text{Adv}^{\text{xor-ind}0}_{\mathcal{F}, \mathcal{G}, B}(k) + 2 \cdot \text{Adv}^{\text{prg}}_{\mathcal{G}, C}(k).$$

This completes the proof.

XOR-NM0. Our second class of assumptions speak to the fact that RSA is non-malleable wrt. XOR. Intuitively, if RSA was XOR malleable, then since is multiplicatively homomorphic it would be (something like) fully homomorphic, which is
unlikely. (Although we do not claim the exact formulation of our definitions imply a formal definition of fully homomorphic.) A similar argument was made by Hofheinz for a non-malleability assumption on the Paillier trapdoor permutation (which is additively homomorphic) wrt. multiplication [Assumption 4.2][66].

Let $\mathcal{F} = (\text{Kg}, \text{Eval}, \text{Inv})$ be a trapdoor permutation family with domain $\text{TDom}$. To adversary $A$, we associate the experiment in Figure 5.15 for every $k \in \mathbb{N}$. We say that $\mathcal{F}$ is XOR-NM0 if for every PPT attacker $A$,

$$\text{Adv}^{\text{xor-nm0}}_{\mathcal{F}, A}(k) = \Pr[XOR-\text{NM0}_{\mathcal{F}}(k) \Rightarrow 1],$$

is negligible in $k$.

XOR-NM1. Let $\mathcal{F} = (\text{Kg}, \text{Eval}, \text{Inv})$ be a trapdoor permutation family with domain $\text{TDom}$. Let $\mathcal{G} : \mathcal{K}_G \times \text{TDom} \rightarrow \mathcal{GRng}$ be a hash function family. To adversary $A$, we associate the experiment in Figure 5.15 for every $k \in \mathbb{N}$. We say that $\mathcal{F}$ is XOR-NM1 with respect to $\mathcal{G}$ if for every PPT adversary $A$,

$$\text{Adv}^{\text{xor-nm1}}_{\mathcal{F}, \mathcal{G}, A}(k) = \Pr[XOR-\text{NM1}_{\mathcal{F}, \mathcal{G}}(k) \Rightarrow 1],$$

is negligible in $k$. 

**Figure 5.15:** Games to define XOR-NM security.
<table>
<thead>
<tr>
<th>Games $G_1(k), G_2(k)$</th>
<th>Games $G_3(k), G_4(k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(f, f^{-1}) \leftarrow K_g(1^k)$</td>
<td>$(f, f^{-1}) \leftarrow K_g(1^k)$</td>
</tr>
<tr>
<td>$K_G \leftarrow K_G(1^k)$; $x \leftarrow \text{TDom}(k)$</td>
<td>$K_G \leftarrow K_G(1^k)$; $x \leftarrow \text{TDom}(k)$</td>
</tr>
<tr>
<td>$(z, \text{state}) \leftarrow A_1(K_G, f(G(K_G, x)))$</td>
<td>$w \leftarrow G(K_G, x)$; $w \leftarrow G(K_G, x')$</td>
</tr>
<tr>
<td>$y \leftarrow f(x \oplus z)$; $y \leftarrow f(x)$</td>
<td>$(\omega, y') \leftarrow A(K_G, f, f(x), w)$</td>
</tr>
<tr>
<td>$(\omega, y') \leftarrow A_2(\text{state}, y)$; $x' \leftarrow f^{-1}(y')$</td>
<td>$x' \leftarrow f^{-1}(y')$</td>
</tr>
<tr>
<td>out $\leftarrow (\omega \oplus x = x \oplus x') \land (\omega \neq 0)$</td>
<td>Return $(\omega = x \oplus x') \land (\omega \neq 0)$</td>
</tr>
<tr>
<td>out $\leftarrow (\omega = x \oplus x') \land (\omega \neq 0)$</td>
<td>Return $(\omega = x \oplus x') \land (\omega \neq 0)$</td>
</tr>
</tbody>
</table>

**Figure 5.16:** Games $G_1$–$G_4$ in the proof of Theorem 18.

**Relations between definitions.** Interestingly, we show XOR-NM0 and XOR-IND1 together imply XOR-NM1.

**Theorem 18** Let $\mathcal{F} = (K_g, \text{Eval}, \text{Inv})$ be a trapdoor permutation family with domain $\text{TDom}$. Let $\mathcal{G} : K_G \times \text{TDom} \rightarrow \text{GRng}$ be a function family. Suppose $\mathcal{F}$ is XOR-NM0 and XOR-IND1 with respect to $\mathcal{G}$. Then, $\mathcal{F}$ is XOR-NM1 with respect to $\mathcal{G}$. In particular, for any adversary $A$, there are adversaries $B, C$ such that

$$\text{Adv}_{\mathcal{F}, \mathcal{G}, A}^{\text{xor-nm1}}(k) \leq \text{Adv}_{\mathcal{F}, \mathcal{G}, B}^{\text{xor-nm0}}(k) + 2 \cdot \text{Adv}_{\mathcal{F}, \mathcal{G}, C}^{\text{xor-\text{ind}1}}(k).$$

The running time of $B$ and $C$ are about that of $A$.

**Proof.** Consider games $G_1$–$G_4$ in Figure 5.16. Game $G_1$ corresponds to the game XOR-NM1$_{\mathcal{F}, \mathcal{G}}^A(k)$. We now explain the game chain. Game $G_2$ is identical to game $G_1$, except instead of giving $f(x \oplus z)$ as an input to the adversary $A_2$ we are using the value $f(x)$. Consider adversary $C$ as shown in Figure 5.17. Note that adversary $C$ simulate games $G_1, G_2$ with respect to it’s inputs. Hence,

$$\Pr[G_1 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1] \leq \text{Adv}_{\mathcal{F}, \mathcal{G}, C}^{\text{xor-\text{ind}1}}(k).$$

89
In game $G_3$, we merge the adversaries $A_1, A_2$ of game $G_2$. The change is conservative, meaning that $\Pr[G_2(k) \Rightarrow 1] = \Pr[G_3(k) \Rightarrow 1]$. Game $G_4$ is identical to game $G_3$, except instead of giving $G(K_G, x)$ as an input to adversary $A_2$ we are using $G(K_G, x')$ for uniformly random $x'$. Consider adversary $D$ as shown in Figure 5.17. Note that adversary $D$ simulate games $G_3, G_4$ with respect to it’s inputs. Hence,

$$\Pr[G_3 \Rightarrow 1] - \Pr[G_4 \Rightarrow 1] \leq \text{Adv}^{\text{xor-ind}1}_{F,G,D}(k).$$

Note that Game $G_4$ corresponds to the game XOR-NM0$^{A}_{F,G}(k)$. Thus,

$$\text{Adv}^{\text{xor-nm}1}_{F,G,A}(k) \leq \text{Adv}^{\text{xor-nm}0}_{F,G,B}(k) + 2 \cdot \text{Adv}^{\text{xor-ind}1}_{F,G,C}(k).$$

This completes the proof.

DISCUSSION. We caution that these are new assumptions and must be treated with care, although they have some intuitive appeal as discussed where they are introduced. It would be interesting for future work to establish theoretical constructions meeting them or show that RSA meets them under more well-studied assumptions.
5.4 Full Instantiation Results (I)

We next give full instantiation results for two variants of RSA-OAEP, called $t$-clear and $s$-clear RSA-OAEP. Prior results on $t$-clear RSA-OAEP [26] showed only partial instantiations or relatively weak security notions, and $s$-clear RSA-OAEP was only considered indirectly by Shoup [93] for negative results. One of the heroes for us here is a hierarchy of “extractability” notions we define and assume for the round functions, called EXT-RO, EXT0, EXT1, EXT2, roughly paralleling PA-RO, PA0, PA1, PA2 respectively, and significantly generalizing prior work [38, 39]. In the case of $s$-clear, another hero is a family of new “XOR-type” assumptions we introduce, speaking to the fact that XOR "breaks" up the multiplicative structure of RSA.

5.4.1 Main Results I

We show that $t$-clear RSA-OAEP is $IND$-$CCA0$-$KI$ secure and $IND$-$CCA1$-$KI$ secure under respective suitable assumptions. As in Section 5.2 we actually prove corresponding notions of IND-CPA + PA, yielding stronger results. The results in follow from those below combined.

$IND$-$CCA0$-$KI$ result. Interestingly, for $IND$-$CCA0$-$KI$ we use milder assumptions on $G$ by performing a direct analysis of OAEP rather than abstracting a property of the underlying the padding scheme and applying the results of Section 4. Namely, we avoid the assumption that $G$ is $\zeta$-injective.

**Theorem 19** Let $\eta, \delta, \mu, \zeta, \rho$ be integer parameters. Let $\mathcal{F}$ be a family of one-way trapdoor permutations with domain $\{0, 1\}^{\mu}$. Let $\mathcal{M}$ be a class of $(\ell, v)$-entropic message samplers, $\mathcal{G} : K_{G} \times \{0, 1\}^{\rho} \to \{0, 1\}^{\mu+\zeta}$ and $\mathcal{H} : K_{H} \times \{0, 1\}^{\mu+\zeta} \to \{0, 1\}^{\rho}$ be function
families. Let \( \eta = ||[K_g(1^k)]|| + ||[K_H(1^k)]|| \) and \( \delta = ||[K_g(1^k)]|| + ||[K_C(1^k)]|| \). Suppose \( \mathcal{G} \) is \( \eta \)-EXT0, \( \mathcal{H} \) is \( \delta \)-EXT0 and CR. Also suppose \( \mathcal{G} \) is a pseudorandom generator and \( \mathcal{H} \) is a hardcore function for \( \mathcal{F}|_{\zeta} \) on class \( \mathcal{M} \). Then \( \text{OAEP}_{\text{t-clear}}[\mathcal{G}, \mathcal{H}, \mathcal{F}|_{\zeta+\rho}] \) is \$IND-CCA0-KI secure. In particular, for every \( \mathcal{M}_0, \mathcal{M}_1 \in \mathcal{M} \) and any adversary \( A \), there exists adversaries \( A_G, A_H, B_G, B_H, B, C \) and a distribution \( X(k) \in \mathcal{M} \) such that for all extractors \( \text{Ext}_G, \text{Ext}_H \)

\[
\text{Adv}^{\text{IND-CCA0-ki}}_{\text{OAEP}_{\text{t-clear}}[\mathcal{G}, \mathcal{H}, \mathcal{F}|_{\zeta+\rho}]}(A) \leq 2 \cdot \text{Adv}^{\eta\text{-ext0}}_{\mathcal{G}, A_G, \text{Ext}_G}(k) + 2 \cdot \text{Adv}^{\delta\text{-ext0}}_{\mathcal{H}, A_H, \text{Ext}_H}(k) + 2 \cdot \text{Adv}^{n\text{-ct}}_{\mathcal{G}, B_G}(k) + 2 \cdot \text{Adv}^{n\text{-cr}}_{\mathcal{H}, B_H}(k) + 2 \cdot \text{Adv}^{\text{hcf}}_{\mathcal{F}|_{\zeta}, X, B}(k) + 2v \cdot \text{Adv}^{\text{prg}}_{\mathcal{G}, C}(k).
\]

$\text{IND-CCA1-KI result.}$ To prove $\text{IND-CCA1-KI}$, we use EXT1 and near-collision resistance of the overall OAEP padding scheme (which follows from corresponding assumptions on the round functions as per Section 4), as well as the assumption that \( \mathcal{G} \) is a pseudorandom generator and \( \mathcal{H} \) is an appropriate hardcore function.

**Theorem 20** Let \( \eta, \mu, \zeta, \rho \) be integer parameters. Let \( \mathcal{F} \) be a family of one-way trapdoor permutations with domain \( \{0,1\}^\mu \). Let \( \mathcal{M} \) be a class of \( (\ell,v) \)-entropic message samplers and \( \eta = ||[K_g(1^k)]|| \). Let \( \mathcal{G} : K_G \times \{0,1\}^\rho \to \{0,1\}^{\mu+\zeta} \) and \( \mathcal{H} : K_H \times \{0,1\}^{\mu+\zeta} \to \{0,1\}^\rho \) be function families. Suppose \( \mathcal{G} \) is a pseudorandom generator, and let \( \mathcal{H} \) be a hardcore function for \( \mathcal{F}|_{\zeta} \) on class \( \mathcal{M} \). Also suppose \( \text{OAEP}[\mathcal{G}, \mathcal{H}] \) is \( \eta\text{-EXT1}_{\zeta+\rho} \) and NCR\(_{\zeta+\rho} \). Then \( \text{OAEP}_{\text{t-clear}}[\mathcal{G}, \mathcal{H}, \mathcal{F}|_{\zeta+\rho}] \) is \$IND-CCA1-KI secure. In particular, for every \( \mathcal{M}_0, \mathcal{M}_1 \in \mathcal{M} \) and any adversary \( A \), there exists adversaries \( B, C, D, E \) and a distribution \( X(k) \in \mathcal{M} \) such that for all extractors \( \text{Ext}_G, \text{Ext}_H \)

\[
\text{Adv}^{\text{IND-CCA1-ki}}_{\text{OAEP}_{\text{t-clear}}[\mathcal{G}, \mathcal{H}, \mathcal{F}|_{\zeta+\rho}]}(A) \leq 2 \cdot \text{Adv}^{\eta\text{-ext1}_{\zeta+\rho}}_{\text{OAEP}[\mathcal{G}, \mathcal{H}], B, \text{Ext}_B}(k) + 2 \cdot \text{Adv}^{n\text{-ct}_{\zeta+\rho}}_{\text{OAEP}[\mathcal{G}, \mathcal{H}], C}(k) + 2 \cdot \text{Adv}^{\text{hcf}}_{\mathcal{F}|_{\zeta}, X, D}(k) + 2v \cdot \text{Adv}^{\text{prg}}_{\text{E}}(k).
\]
EFFICIENCY. The ciphertext length in the above instantiations is $3n + 3k$ where $n$ is the length of the RSA modulus and $k$ is the security parameter. The scheme has message length $n$. For example, if $n = 2048$ and $k = 128$ then the ciphertext length is 6528 bits. The time to run the encryption and decryption algorithms is basically that of standard RSA-OAEP.

**Remark 21** We note that while the restriction to public-key independent messages is inherent for deterministic encryption, for randomized encryption it is not and we leave it as an interesting open problem to extend the result to public-key dependent messages. Additionally, the high-entropy requirement on messages can be avoided by assuming $G$ and $H$ are “universal computational extractors” (UCE) in the sense of [15], which follows from their results but we omit this for simplicity.

5.4.2 Main Results II

After establishing its security in the RO model, we show that $s$-clear RSA-OAEP is IND-CCA1 and IND-CCA2 under respective suitable assumptions. As in Section 5.2 we actually prove corresponding notions of IND-CPA + PA, yielding stronger results. The results in Section follow from those below.

**IND-CCA2 result in RO model.** First, note that the partial one-wayness result of [57] does not apply to this variant, and in fact the negative result of [93] does apply, demonstrating that one-wayness of the trapdoor permutation is not enough for the scheme to achieve IND-CCA2 security 

\textit{even in the RO model}. We show that XOR-nonmalleability is sufficient.

**Theorem 22** Let $\mu, \zeta, \rho$ be integer parameters. Let $F$ be a XOR-NM0 family of one-way trapdoor permutations with domain $\{0,1\}^\rho$. Suppose $G : K_G \times \{0,1\}^\rho \rightarrow \{0,1\}^{\mu + \zeta}$ is a RO and $H : K_H \times \{0,1\}^{\mu + \zeta} \rightarrow \{0,1\}^\rho$ is collision-resistant. Then
\( \text{OAEP}_{s\text{-clear}}[G, H, \mathcal{F}]^{\mu+\zeta} \) is IND-CCA2 secure in the random oracle model. In particular, for any adversary \( A \), there are adversaries \( B, C \) such that

\[
\text{Adv}_{\text{OAEP}_{s\text{-clear}}, A}(k) \leq \frac{2q}{2^\rho} + \frac{4p}{2^\zeta} + 2 \cdot \text{Adv}_{H, C}^{\text{cr}}(k) + 4 \cdot \text{Adv}_{\mathcal{F}, B}^{\text{xor-nm0}}(k),
\]

where \( p \) is the number of decryption-oracle queries of \( A \) and \( q \) is the total number of random-oracle queries of \( A \) and \( M \). Adversary \( B \) and \( C \) makes at most \( q \) random-oracle queries.

**IND-CCA1 result.** To prove IND-CCA1, we use EXT1 and near-collision resistance of the overall OAEP padding scheme (which follows from assumptions on the round functions as per Section 4), as well as the assumption that \( G \) is a pseudorandom generator and \( \mathcal{F} \) is XOR-IND (as defined in Section 5.3).

**Theorem 23** Let \( \eta, \mu, \zeta, \rho \) be integer parameters. Let \( \mathcal{F} \) be a family of trapdoor permutations with domain \( \{0,1\}^\mu \), and let \( \eta = |\mathbb{K}_G(1^k)| \). Let \( G : \mathcal{K}_G \times \{0,1\}^\rho \rightarrow \{0,1\}^{\mu+\zeta} \) and \( H : \mathcal{K}_H \times \{0,1\}^{\mu+\zeta} \rightarrow \{0,1\}^\rho \) be function families. Suppose \( G \) is a pseudorandom generator, and let \( \mathcal{F} \) is XOR-IND0 with respect to hint function \( G \) (as defined in Section 5.3). Also suppose \( \text{OAEP}[G, H] \) is \( \eta\text{-EXT}_1^{\mu+\zeta} \) and \( \text{NCR}^{\mu+\zeta} \). Then \( \text{OAEP}_{s\text{-clear}}[G, H, \mathcal{F}]^{\mu+\zeta} \) is IND-CCA1 secure. In particular, for any adversary \( A \) that makes \( q \) decryption queries, there exist adversaries \( C, D, E \), and EXT1 adversary \( B \) that makes \( q \) extract queries such that for all extractors \( \text{Ext} \),

\[
\text{Adv}_{\text{OAEP}_{s\text{-clear}}, A}^{\text{ind-cca1}}(k) \leq 2 \cdot \text{Adv}_{\text{OAEP}[G, H], B, \text{Ext}}^{\eta\text{-ext}_1^{\mu+\zeta}}(k) + 2 \cdot \text{Adv}_{\text{OAEP}[G, H], C}^{\text{n-cr}^{\mu+\zeta}}(k) + 6 \cdot \text{Adv}_{\mathcal{F}, D}^{\text{xor-ind0}}(k) + 4 \cdot \text{Adv}_{G, E}^{\text{prg}}(k).
\]

**IND-CCA2 result.** To prove IND-CCA2, we use EXT2 and near-collision resistance of \( G \), as well as the assumptions that \( G \) is a pseudorandom generator, \( H \) is collision-resistant and \( \mathcal{F} \) is XOR-IND and XOR-NM (as defined in Section 5.3). Note
that, EXT2 adversary only makes one image query. Thus, the dependent auxiliary information is bounded by the size of the image.

**Theorem 24** Let $\eta, \mu, \zeta, \rho$ be integer parameters. Let $\mathcal{F}$ be a family of trapdoor permutations with domain $\{0, 1\}^\mu$ and $\eta = ||\mathcal{K}_g(1^k)|| + ||\mathcal{K}_H(1^k)||$. Let $\mathcal{G} : \mathcal{K}_G \times \{0, 1\}^\rho \rightarrow \{0, 1\}^{\mu + \zeta}$ and $\mathcal{H} : \mathcal{K}_H \times \{0, 1\}^{\mu + \zeta} \rightarrow \{0, 1\}^\rho$ be function families. Suppose $\mathcal{G}$ is VPRG$_\zeta$, NCR$_\zeta$ and $\eta$-EXT2$_\zeta$ with respect to $\mathcal{F}$, and $\mathcal{H}$ is collision-resistant. Suppose $\mathcal{F}$ is XOR-NM0, XOR-IND1 and XOR-IND2$_\zeta$ with respect to $\mathcal{G}$. Then OAEP$_{\text{secure}}[\mathcal{G}, \mathcal{H}, \mathcal{F}]^{\mu + \zeta}$ is IND-CCA2 secure. In particular, for any adversary $A$ that makes $q$ decryption queries, there exists adversaries $C_H, C_G, D_1, D_2, D_3, E$, and adversary $B$ that makes $q$ extract queries such that for all extractors $\text{Ext}$,

$$\text{Adv}_{\text{OAEP}_{\text{secure}}}^{\text{ind-cca2}}(A(k)) \leq 6 \cdot \text{Adv}_{\mathcal{G}, \mathcal{F}, B, \text{Ext}}^{\eta\text{-ext2}_\zeta}(k) + 18 \cdot \text{Adv}_{\mathcal{F}, \mathcal{G}, D_1}^{\text{xor-ind2}_\zeta}(k)$$

$$\quad + 10 \cdot \text{Adv}_{\mathcal{G}, C_G}^{\text{n-crcr}_\zeta}(k) + 4 \cdot \text{Adv}_{\mathcal{H}, C_H}^{\text{cr}}(k) + 4 \cdot \text{Adv}_{\mathcal{F}, \mathcal{G}, D_3}^{\text{xor-nm0}}(k)$$

$$\quad + 14 \cdot \text{Adv}_{\mathcal{F}, \mathcal{G}, D_2}^{\text{xor-ind1}}(k) + 16 \cdot \text{Adv}_{\mathcal{G}, E}^{\text{vprg}_\zeta}(k).$$

**Efficiency.** The ciphertext length is $2n + k + \mu$ where $n$ is the length of the RSA modulus, $k$ is the security parameter, and $\mu$ is the message length. For example, if $n = 2048$, $k = 128$, and we encrypt an AES key with $\mu = 128$ (i.e., we use RSA-OAEP as a key encapsulation mechanism, which is typical in practice then the ciphertext length is 4352). It is interesting to compare this with the standard model IND-CCA2 secure key encapsulation mechanism of Kiltz et al. [68]. They describe their scheme based on modular squaring (factoring), but it is straightforward to derive a scheme based on RSA with large hardcore function and a cryptographic hash function being target collision-resistant, which results in the most efficient prior standard-model RSA-based encryption scheme we are aware of. It performs one “small” exponentiation wrt. $e$ and one “full” exponentiation modulo $N$, so is much more computationally expensive than...
our scheme. Thus, one could arguably say ours is the most computationally efficient
RSA-based encryption scheme under “plausible standard-model assumptions” (where
one takes the liberty of making bold assumptions on cryptographic hash functions)
to date. On the other hand, the scheme of [68] has ciphertext length only \(2n\).

**Remark 25** It is worth mentioning why we are able to get IND-CCA2 (i.e., adaptive)
security for \(s\)-clear RSA-OAEP but not \(t\)-clear. The point is that, in the \(t\)-clear setting,
it is not even clear how to define EXT2 of OAEP in a useful way. Since OAEP is
invertible, the image oracle should output only part of the image point. But then it is
not clear how the EXT2 adversary against OAEP can simulate the encryption oracle
for the PA2 adversary against \(t\)-clear RSA-OAEP. On the other hand, for EXT2
of \(G\), the image oracle can output the full image point since \(G\) is not invertible. This
then allows proving that \(s\)-clear RSA-OAEP is PA2 directly (without using monolithic
assumptions on the padding scheme not known to follow from assumptions on the
round functions).

### 5.4.3 Full Instantiation Results for \(t\)-Clear RSA-OAEP

Here, we give full instantiation results for \(t\)-clear RSA-OAEP. Our results for \(t\)-clear
RSA-OAEP are weaker than those for \(s\)-clear RSA-OAEP. First, for \(t\)-clear we prove
IND-CPA for random, public key independent messages, under mild assumptions
on the round functions, namely that \(\mathcal{H}\) is a hardcore function for RSA and \(G\) is a
pseudorandom generator. Then we can prove PA0 based on forms of EXT0 for \(G\) and
\(\mathcal{H}\). Furthermore, we prove PA1, although we have to make an extractability directly
on the padding scheme. Interestingly, even this approach does not work for PA2,
which we leave completely open for \(t\)-clear.
$\text{IND-CPA-KI result.}$ We first show that $t$-clear RSA-OAEP is $\text{IND-CPA-KI}$ for messages independent of the public key. Note that the prior result on full instantiation of $t$-clear RSA-OAEP by Boldyreva and Fischlin [26] also apply to public-key-independent messages. Intuitively, we require additional high min-entropy message sampler because the randomness for the distribution on which $\mathcal{H}$ is hardcore for $\mathcal{F}|_{\zeta}$ comes from message $m$, with coins $r$ fixed. Again, we note that we could avoid the high min-entropy requirement by using the result in [15] that RSA-OAEP if RSA is one-way and $\mathcal{G}, \mathcal{H}$ are UCE. We prefer to stick with our more mild assumptions as our result is novel with a non-trivial proof and probably sufficient in practice. (The UCE result could be seen as a hedge.)

**Theorem 26** Let $\mu, \zeta, \rho$ be integer parameters. Let $\mathcal{F}$ be a family of one-way trapdoor permutations with domain $\{0, 1\}^{\mu}$. Let $\mathcal{M}$ be a class of $(\ell, v)$-entropic message samplers. Suppose $\mathcal{G} : \mathcal{K}_G \times \{0, 1\}^{\rho} \rightarrow \{0, 1\}^{\mu+\zeta}$ is a pseudorandom generator and $\mathcal{H} : \mathcal{K}_H \times \{0, 1\}^{\mu+\zeta} \rightarrow \{0, 1\}^{\rho}$ is a hardcore function for $\mathcal{F}|_{\zeta}$ on class $\mathcal{M}$. Then $\text{OAEP}_{t\text{-clear}}[\mathcal{G}, \mathcal{H}, \mathcal{F}|_{\zeta+\rho}]$ is $\text{IND-CPA-KI}$ secure. In particular, for every $\mathcal{M}_0, \mathcal{M}_1 \in \mathcal{M}$ and any adversary $A$, there are adversaries $B, C$ and distribution $X(k) \in \mathcal{M}$ such that

$$\text{Adv}_{\text{OAEP}_{t\text{-clear}}[\mathcal{G}, \mathcal{H}, \mathcal{F}|_{\zeta+\rho}]}^{\text{IND-CPA-KI}}(A, \mathcal{M}_0, \mathcal{M}_1)(k) \leq 2 \cdot \text{Adv}_{\mathcal{F}|_{\zeta}, \mathcal{H}, X, B}^{\text{hcf}}(k) + 2v \cdot \text{Adv}_{\mathcal{G}, C}^{\text{prg}}(k).$$

The running time of $B$ is up to that of $A$. The running time of $X(k)$ and $C$ is the time to run $A$ plus the running time of $\mathcal{M}$.

**Proof.** Consider games $G_1$–$G_4$ in Figure 5.18. We now explain the game chain. Game $G_1$ corresponds to game $\text{IND-CPA-KI}_{\text{OAEP}_{t\text{-clear}}}[A, \mathcal{M}_0, \mathcal{M}_1]$. Game $G_2$ is identical to game $G_1$, except we are using completely random $z$ in the encryption phase instead of using the hash value. Consider distribution $X$ and adversary $B$ in Figure 5.19. Note
that $X(k) \in \mathscr{M}$. Distribution $X$ and adversary $B$ collaborate to simulate game $G_1$. Adversary $B$ returns 0 if adversary $A$ can correctly guess simulated challenge bit $b$, and returns 1 otherwise. Then

$$\Pr[G_1(k) \Rightarrow 1] - \Pr[G_2(k) \Rightarrow 1] \leq \text{Adv}^{\text{hcf}}_{\mathcal{F}_C \mid \mathcal{H}, X, B}(k).$$

In game $G_3$, we reorder the code of game $G_2$ producing $z$. The change is conservative, meaning that $\Pr[G_2(k) \Rightarrow 1] = \Pr[G_3(k) \Rightarrow 1]$. Next, game $G_4$ is identical to game $G_3$, except we are using completely random $x$ instead of pseudorandom value $G(K_G, r)$ in the encryption phase. For $i \in [v]$, we define the adversary $C_i$ as shown in Figure 5.20. Hence,

$$\Pr[G_3 \Rightarrow 1] - \Pr[G_4 \Rightarrow 1] \leq \sum_{i=1}^{v} \text{Adv}^{\text{prg}}_{G_i, C_i}(k).$$

Assume there exists adversary $C$ such that for all $i \in [v]$, we have $\text{Adv}^{\text{prg}}_{G_i, C_i}(k) \leq \text{Adv}^{\text{prg}}_{G, C}(k)$. Note that $\Pr[G_4(k) \Rightarrow 1] = 1/2$, since the distribution of the ciphertexts

---

**Games $G_1(k), G_2(k)$**

$b \leftarrow \{0, 1\}; \ param \leftarrow A_{\text{pg}}(1^k) \\
K_G \leftarrow Kg(G(1^k)) \ ; \ K_H \leftarrow Kg(H(1^k)) \\
(f, f^{-1}) \leftarrow Kg(1^k) \\
\text{pk} \leftarrow (K_G, K_H, f) \\
\text{m} \leftarrow M_b(1^k, \text{param})$

For $i = 1$ to $|\text{m}|$

\[ r \leftarrow \{0, 1\}^\rho \ ; \ x \leftarrow G(K_G, r) \]
\[ s \leftarrow m[i] \oplus x \ ; \ s_1 \leftarrow s |^\mu \ ; \ s_2 \leftarrow s |^\zeta \]
\[ z \leftarrow H(K_H, s) \ ; \ z \leftarrow \{0, 1\}^\rho \]
\[ t \leftarrow r \oplus z \ ; \ y \leftarrow f(s_1) \]
\[ c[i] \leftarrow (y, s_2, t) \]
\[ b' \leftarrow A.g(pk, c, \text{param}) \]

Return $(b = b')$

---

**Games $G_3(k), G_4(k)$**

$b \leftarrow \{0, 1\}; \ param \leftarrow A_{\text{pg}}(1^k) \\
\text{K}_G \leftarrow Kg(G_2(k)) \ ; \ \text{K}_H \leftarrow Kg(H(1^k)) \\
(f, f^{-1}) \leftarrow Kg(1^k) \\
\text{pk} \leftarrow (K_G, K_H, f) \\
\text{m} \leftarrow M_b(1^k, \text{param})$

For $i = 1$ to $|\text{m}|$

\[ r \leftarrow \{0, 1\}^\rho \ ; \ x \leftarrow G(K_G, r) \ ; \ x \leftarrow \{0, 1\}^\mu \oplus \zeta \]
\[ s \leftarrow m[i] \oplus x \ ; \ s_1 \leftarrow s |^\mu \ ; \ s_2 \leftarrow s |^\zeta \]
\[ t \leftarrow \{0, 1\}^\rho \ ; \ z \leftarrow r \oplus t \]
\[ y \leftarrow f(s_1) \ ; \ c[i] \leftarrow (y, s_2, t) \]
\[ b' \leftarrow A.g(pk, c, \text{param}) \]

Return $(b = b')$
Algorithm \(X(k)\)
\[
\begin{align*}
    b & \leftarrow \{0, 1\} ; \text{param} \leftarrow A.pg(1^k) \\
    K_G & \leftarrow K_G(1^k) \\
    m & \leftarrow M_b(1^k, \text{param}) \\
    \text{For } i = 1 \text{ to } |m| \text{ do} \\
    \quad r[i] & \leftarrow \{0, 1\}^\mu ; x \leftarrow G(K_G, r[i]) \\
    \quad s & \leftarrow m[i] \oplus x ; s_1[i] \leftarrow s^\mu ; s_2[i] \leftarrow s^\zeta \\
    \quad \alpha & \leftarrow (r, s_2, K_G, b, \text{param}) \\
    \text{Return } (s_1, \alpha)
\end{align*}
\]

Algorithm \(B(K_H, f, y, \alpha, z)\)
\[
\begin{align*}
    (r, s_2, K_G, b, \text{param}) & \leftarrow \alpha \\
    pk & \leftarrow (K_G, K_H, f) \\
    \text{For } i = 1 \text{ to } |z| \text{ do} \\
    \quad t & \leftarrow r[i] \oplus z[i] \\
    \quad c[i] & \leftarrow (y[i], s_2[i], t) \\
    \quad b' & \leftarrow A.g(pk, c, \text{param}) \\
    \text{Return } (b \neq b')
\end{align*}
\]

Figure 5.19: Distribution \(X\) (left) and adversary \(B\) (right) in the proof of Theorem 26.

is completely independent of bit \(b\). Summing up,
\[
\operatorname{Adv}^{\operatorname{Sind-CPA-ki}}_{\text{OAEP}_{t-\text{clear}}, A, M_0, M_1}(k) \leq 2 \cdot \operatorname{Adv}^{\operatorname{def}}_{\mathcal{F}_\zeta, \mathcal{H}, X, B}(k) + 2v \cdot \operatorname{Adv}^{\operatorname{prg}}_{G, C}(k)
\]

This completes the proof.

PA0 result. We show PA0 of \(t\)-clear RSA-OAEP. As mentioned above, here we obtain a better result by using properties of the round functions of OAEP directly, rather than properties of the overall padding scheme.

**Theorem 27** Let \(\eta, \delta, \mu, \zeta, \rho\) be integer parameters. Let \(\mathcal{F}\) be a family of one-way trapdoor permutations with domain \(\{0, 1\}^\mu\). Let \(\mathcal{G} : K_G \times \{0, 1\}^\rho \rightarrow \{0, 1\}^{\mu + \zeta}\) and \(\mathcal{H} : K_H \times \{0, 1\}^{\mu + \zeta} \rightarrow \{0, 1\}^\rho\) be function families. Let \(\eta = ||K_g(1^k)|| + ||K_G(1^k)||\) and \(\delta = ||K_g(1^k)|| + ||K_G(1^k)||\). Suppose \(\mathcal{G}\) is \(\eta\)-\(\operatorname{EXT0}_\zeta\) and \(\operatorname{NCR}_\zeta\), \(\mathcal{H}\) is \((\delta, \zeta)\)-\(\operatorname{EXT0}\) and \(\operatorname{CR}\). Then \(\text{OAEP}_{t-\text{clear}}[\mathcal{G}, \mathcal{H}, \mathcal{F}_{\zeta + \rho}]\) is PA0 secure. In particular, for any adversary \(A\), there exist adversaries \(A_G, A_H, B_G, B_H\) and an extractor \(\text{Ext}\) such that for all extractors \(\text{Ext}_G, \text{Ext}_H\)
\[
\operatorname{Adv}^{\text{PA0}}_{\text{OAEP}_{t-\text{clear}}, A, \text{Ext}}(k) \leq \operatorname{Adv}^{\eta-\text{ext0}_\zeta}_{G, A_G, \text{Ext}_G}(k) + \operatorname{Adv}^{(\delta, \zeta)-\text{ext0}}_{H, A_H, \text{Ext}_H}(k)
\]
\[
+ \operatorname{Adv}^{n-\text{cr}}_{G, B_G}(k) + \operatorname{Adv}^{\text{cr}}_{H, B_H}(k)
\]

99
The running time of \( A_G \) is about that of \( A \) and the running time of \( A_H \) is about that of \( A \) plus the time to run \( \text{Ext}_G \). The running time of \( B_G \) is the time to run \( A \) and \( \text{Ext}_G \). The running time of \( B_H \) is the running time of \( A, \text{Ext}_H \) and \( \text{Ext}_G \). The running time of \( \text{Ext} \) is the time to run \( \text{Ext}_G \) and \( \text{Ext}_H \).

Proof. Let \( w \) be the randomness of adversary \( A \) in the game PA0. Let \( K_G \) be the key for the function family \( \mathcal{G} \), \( K_H \) be the key for the function family \( \mathcal{H} \) and \( f \) be the evaluation key for the trapdoor permutation family \( \mathcal{F} \) in the game PA0. We define \( \text{EXT0} \) adversaries \( A_G, A_H \) with randomness \( w \) in Figure 5.21. Let \( v = (K_H, f) \) be the key independent auxiliary input to \( A_G \) and \( u = (K_G, f) \) be the key independent auxiliary input to \( A_H \). Note that auxiliary input \( v \) and \( u \) are independent of key \( K_G \) and \( K_H \), respectively. Let \( \text{Ext}_G \) and \( \text{Ext}_H \) be the corresponding extractor for \( A_G \) and \( A_H \), respectively. We define PA0 extractor \( \text{Ext} \) as shown in Figure 5.22.

Note that for decryption query \( c \) that \( A \) makes, if ciphertext \( c \) is not valid then extractor \( \text{Ext} \) outputs \( \bot \). Thus, adversary \( A \) does not gain any information about \( b \) by
making an invalid decryption query. Hence, we assume wlog that adversary A queries only valid ciphertexts. Let $c$ be the decryption query that A makes and $r_c$ and $s_c$ be the corresponding middle values in the computation of the ciphertext $c$. Let $r$ be the output of $\text{Ext}_G$ and $s_1$ be the output of $\text{Ext}_H$. Wlog, we can assume when $\text{Ext}_G$ and $\text{Ext}_H$ output a non-empty string, they were successful in finding the preimages. Let $W_1$ be the event that $r$ is a non-empty string and $r \neq r_c$ and $W_2$ be the event that $s$ is a non-empty string and $s_1 \neq s_c|\mu$. Let $W = W_1 \lor W_2$. Let $S$ be the event that game $\text{PA0}_{\text{OAEP}, r-\text{clear}}^A(k)$ outputs 1. Note that all of the following probabilities are over
the choice of public key $pk$ and randomness $w$. Then
\[
\text{Adv}_{\text{OAEP-\text{clear},A,Ext}}^{\text{pa0}}(k) = 2 \cdot (\Pr[S \wedge W] + \Pr[S \wedge \neg W]) - 1.
\]

Now consider near collision resistance adversary $B_G$ in Figure 5.23 and collision resistance adversary $B_H$ in Figure 5.24. If event $W$ happens, then either $B_G$ or $B_H$ finds a collision. Thus, we have $\Pr[W] \leq \text{Adv}_{G,B_G}^{\text{ncr}}(k) + \text{Adv}_{H,B_H}^{\text{cr}}(k)$. Then
\[
\Pr[S \wedge W] \leq \text{Adv}_{G,B_G}^{\text{ncr}}(k) + \text{Adv}_{H,B_H}^{\text{cr}}(k).
\]
Let $E$ be the event such that for decryption query $c$ adversary $A$ makes, the outputs of decryption algorithm $\text{Dec}$ and extractor $\text{Ext}$ are equal. Then,

\[ \Pr[ S \land W ] = \Pr[ S \land W \land E ] + \Pr[ S \land W \land \overline{E} ] . \]

Note that $W$ and $E$ are mutually exclusive and when event $W$ happens the output of $\text{Ext}$ is incorrect. Thus, we have $\Pr[ S \land W \land E ] = \Pr[ S \land E ]$. Moreover, we have $\Pr[ S \mid E ] = 1/2$ since for the query made by $A$ the outputs of $\text{Dec}$ and $\text{Ext}$ are equal. Hence,

\[ \Pr[ S \land E ] = \frac{1}{2} \cdot \Pr[ E ] . \]

Consider adversaries $A_G, A_H$ in Figure 5.21. We know adversary $A$ always makes valid decryption query. Thus, when event $\overline{E}$ happens extractor $\text{Ext}$ outputs $\bot$. This implies that either $r \neq r_c$ where $r$ is the output of $\text{Ext}_G$ or $s_1 \neq s_c|\mu$ where $s_1$ is the output of $\text{Ext}_H$. We also know when event $\overline{W}$ happens, extractor $\text{Ext}_G$ either outputs $\bot$ or $r_c$, and extractor $\text{Ext}_H$ outputs either $\bot$ or $s_c|\mu$. Therefore when events $\overline{E}$ and $\overline{W}$ happen, either $\text{Ext}_G$ or $\text{Ext}_H$ fails. Thus,

\[ \Pr[ \overline{W} \land \overline{E} ] \leq \text{Adv}_{\text{OAEP}_{t\text{-clear}}/A,\text{Ext}}^0(k) + \text{Adv}_{\text{ext}0/\delta,\zeta}^0(k) . \]

On the other hand, we know that $E$ and $W$ mutually exclusive. Hence, we get $\Pr[ E ] = \Pr[ W \lor E ] - \Pr[ W ]$. Summing up,

\[
\text{Adv}_{\text{OAEP}_{t\text{-clear}}/A,\text{Ext}}^0(k) \leq 2 \cdot \Pr[ W ] + (\Pr[ W \lor E ] - \Pr[ W ]) + 2 \cdot \Pr[ \overline{W} \land \overline{E} ] - 1 \\
\leq \text{Adv}_{\text{ext}0/\delta,\zeta}^0(k) + \text{Adv}_{\text{ext}0/\delta,\zeta}^0(k) \\
+ \text{Adv}_{\text{cr}0/\delta,\zeta}^0(k) + \text{Adv}_{\text{cr}0/\delta,\zeta}^0(k) .
\]

This completes the proof.

PA1 result. We now show that $t$-clear RSA-OAEP “inherits” the extractability of the underlying padding transform, in the form of PA1 and EXT1, as long as the latter
is also near-collision resistant. Here we state the result for an abstract padding scheme rather than specifically for OAEP. Interestingly, this approach does not seem to work for PA2 and EXT2. We leave PA2 of the encryption scheme as an open problem.

**Theorem 28** Let $\delta, \mu, \zeta, \rho$ be integer parameters. Let $F$ be a family of one-way trapdoor permutations with domain $\{0,1\}^{\mu}$ and $\delta = |\{Kg(1^k)\}|$. Let $\text{PAD}$ be a padding transform from domain $\{0,1\}^{\mu+\rho}$ to range $\{0,1\}^{\mu+\zeta+\rho}$. Suppose $\text{PAD}$ is NCR$_{\zeta+\rho}$ and $\delta$-EXT1$_{\zeta+\rho}$. Then the padding-based encryption scheme $\text{PAD}[F|_{\zeta+\rho}]$ is PA1 secure. In particular, for any PA1 adversary $A$ that makes $q$ queries, there exist an EXT1 adversary $A_{\text{PAD}}$ that makes $q$ queries, an NCR$_{\zeta+\rho}$ adversary $B$ and an extractor $\text{Ext}$ such that for all extractors $\text{Ext}_{\text{PAD}}$

$$\text{Adv}_{\text{PAD}[F|_{\zeta+\rho}],A,\text{Ext}}(k) \leq \text{Adv}_{\text{PAD},A_{\text{PAD}},\text{Ext}_{\text{PAD}}}(k) + \text{Adv}^{n-\text{cr}}_{\text{PAD},B}(k).$$

The running time of $A_{\text{PAD}}$ is that of $A$. The running time of $B$ is about the time to run $A$, $\text{Ext}_{\text{PAD}}$ and $\text{Ext}$. The running time of $\text{Ext}$ is about that of $\text{Ext}_{\text{PAD}}$.

**Proof.** Let $w$ be the randomness of PA1 adversary $A$. We define EXT1 adversary $A_{\text{PAD}}$ with randomness $w$ in Figure 5.25. Note that auxiliary input $f$ is independent of the key $\pi$. Let $\text{Ext}_{\text{PAD}}$ be the corresponding extractor for $A_{\text{PAD}}$. We define PA1 extractor $\text{Ext}$ as shown in Figure 5.26.

Note that for any decryption query $c$ adversary $A$ makes, if $c$ is not valid then extractor $\text{Ext}$ outputs $\bot$. Thus, $A$ does not gain any information about $b$ by making invalid decryption queries. Hence, we assume wlog that adversary $A$ only makes queries for valid ciphertext $c$. Assume $A$ makes $q$ extract queries. Let $c_i$ be the $i$-th query $A$ makes to the extract oracle and $v_i$ be the output of extractor $\text{Ext}_{\text{PAD}}$ on input $c_i|_{\zeta+\rho}$. Let $W$ be the event where there exists a valid ciphertext $c_i$ such that $v_i$ is non-empty and extractor $\text{Ext}$ outputs $\bot$ on input $c_i$. Let $S$ be the event that game
**Algorithm Ext(state, c)**

\[
\begin{align*}
(pk, w, st) &\leftarrow \text{state} ; \ (\pi, f) \leftarrow pk \\
(v, st) &\leftarrow \text{ExtPAD}(st, \pi, f, \perp, c[\zeta+\rho]; w) \\
\text{state} &\leftarrow (pk, w, st) \\
m &\leftarrow v|\mu \ ; \ r \leftarrow v|\rho \\
\text{If } c \neq\text{Enc}(pk, m; r) &\text{ then return } (\perp, \text{state}) \\
\text{Return } (m, \text{state})
\end{align*}
\]

**Figure 5.26: PA1 extractor Ext in the proof of Theorem 28.**

PA1\text{Ext}_{PA1,F|\zeta+\rho}(k) outputs 1. Note that all of the following probabilities are over the choice of public key \( pk \) and randomness \( w \). Then,

\[
\text{Adv}_{PA1,F|\zeta+\rho},A,\text{Ext}(k) = 2 \cdot (\Pr [S \land W] + \Pr [S \land \overline{W}]) - 1.
\]

Note that extractor \text{Ext}_{PAD} either outputs the correct value or \( \perp \). Now consider near-collision resistance adversary \( B \) in Figure 5.27. If event \( W \) happens, then \( B \) finds a collision. Thus, we have \( \Pr [W] \leq \text{Adv}_{PA1,F|\zeta+\rho}(k) \). Then

\[
\Pr [S \land W] \leq \text{Adv}_{PA1,F|\zeta+\rho}(k).
\]

Let \( E \) be the event such that for each decryption query \( c_i \) that adversary \( A \) makes, the output of decryption algorithm \text{Dec} and extractor \text{Ext} are equal. Then,

\[
\Pr [S \land \overline{W}] = \Pr [S \land \overline{W} \land E] + \Pr [S \land \overline{W} \land \overline{E}].
\]
Adversary $B(\pi, \hat{\pi})$
\begin{align*}
\text{out}_1 & \leftarrow \varepsilon; \quad \text{out}_2 \leftarrow \varepsilon \\
 w & \leftarrow \text{Coins}(k); \quad b \leftarrow \{0, 1\} \\
 (f, f^{-1}) & \leftarrow \text{Kg}(1^k) \\
 \text{pk} & \leftarrow (\pi, f); \quad \text{sk} \leftarrow (\hat{\pi}, f^{-1}) \\
 \text{st} & \leftarrow (\pi, w); \quad \text{state} \leftarrow (\text{pk}, w) \\
 \text{Run } A^{\text{DSim}(1)}(\text{pk}; w) & \\
 \text{Return } (\text{out}_1, \text{out}_2)
\end{align*}

Procedure $\text{DSim}(c)$
\begin{align*}
 y_2 & \leftarrow c_{\zeta + \rho}; \quad y_1 \leftarrow f^{-1}(c|\mu) \\
 y & \leftarrow y_1||y_2; \quad \text{out}_1 \leftarrow \pi(y) \\
 (\text{out}_2, \text{st}) & \leftarrow \text{Ext}_{\text{PAD}}(\text{st}, \pi, f, \perp, c|_{\zeta + \rho}; w) \\
 \text{If } (\pi(\text{out}_1)|_{\zeta + \rho} = \pi(\text{out}_2)|_{\zeta + \rho} \land \text{out}_1 \neq \text{out}_2) & \text{ Halt } A \\
 m_0 & \leftarrow \text{Dec}(\text{sk}, c) \\
 (m_1, \text{state}) & \leftarrow \text{Ext}(\text{state}, c) \\
 \text{Return } m_b
\end{align*}

Figure 5.27: NCR adversary $B$ in the proof of Theorem 28.

Note that $W$ and $E$ are mutually exclusive since when event $W$ happens the output of the extractor $\text{Ext}$ would be incorrect for at least one of the extract queries. Thus, we have $\Pr [ S \land \overline{W} \land E ] = \Pr [ S \land E ]$. Moreover, we have $\Pr [ S \mid E ] = 1/2$, since for all queries made by $A$ the output of decryption oracle $\text{Dec}$ and extractor $\text{Ext}$ are equal. Hence,
\[
\Pr [ S \land E ] = \frac{1}{2} \cdot \Pr [ E ] .
\]

Consider the adversary $A_{\text{PAD}}$ in Figure 5.25. We know adversary $A$ always makes valid decryption query. Thus, when event $\overline{E}$ happens extractor $\text{Ext}$ outputs $\perp$ for at least one of the extract queries. Moreover, if $\overline{W}$ happens then for all extract queries either $\text{Ext}_{\text{PAD}}$ outputs $\perp$ or $\text{Ext}$ outputs non-empty string. Therefore, when $\overline{E}$ and $\overline{W}$ happen, extractor $\text{Ext}_{\text{PAD}}$ fails and outputs $\perp$ for at least one of the extract queries. Thus,
\[
\Pr [ \overline{W} \land \overline{E} ] \leq \text{Adv}^{\text{Ext}_{\zeta + \rho}}_{\text{PAD}, A_{\text{PAD}}, \text{Ext}_{\text{PAD}}}(k) .
\]

On the other hand, we know that $E$ and $W$ mutually exclusive. Hence, we get $\Pr [ E ] = \Pr [ W \lor E ] - \Pr [ W ]$. Summing up,
\[
\text{Adv}^{\text{pa1}}_{\text{PAD}[\mathcal{F}_{\zeta + \rho}], A, \text{Ext}}(k) \leq \text{Adv}^{\text{Ext}_{\zeta + \rho}}_{\text{PAD}, A_{\text{PAD}}, \text{Ext}_{\text{PAD}}}(k) + \text{Adv}^{\text{er}_{\zeta + \rho}}_{\text{PAD}, B}(k) .
\]
This completes the proof.

5.4.4 Full Instantiation Results for s-Clear RSA-OAEP

In this section, we give full instantiation results for s-clear RSA-OAEP. Note that we are the first to consider this variant. We show that s-clear is IND-CCA2 if $G$ is a pseudorandom generator, near-collision resistant, and “many-times” extractable with dependent auxiliary information, $H$ is collision-resistant, and $F$ meets novel “XOR-nonmalleability” and “XOR-indistinguishability” notions that seem plausible for RSA. Also note that we avoid the several impossibility results here. First, we avoid the impossibility result of [92] by using XOR-non-malleability of $F$. Second, we avoid the impossibility result of [20] since the dependent auxiliary information is bounded.

IND-CPA result. We first show that s-clear RSA-OAEP is IND-CPA secure under suitable assumptions. Then, we show PA0, PA1 and PA2 security depend on the strength of assumptions on $G, H$ and $F$. Interestingly, our IND-CPA result uses an XOR-based assumption on the trapdoor permutation.

**Theorem 29** Let $\mu, \zeta, \rho$ be integer parameters. Let $F$ be a family of trapdoor permutations with domain $\{0, 1\}^\rho$. Suppose $G : K_G \times \{0, 1\}^\rho \rightarrow \{0, 1\}^{\mu + \zeta}$ is a pseudorandom generator and $H : K_H \times \{0, 1\}^{\mu + \zeta} \rightarrow \{0, 1\}^\rho$ is a family of hash function. Suppose $F$ is XOR-IND0 with respect to hint function family $G$. Then $\text{OAEP}_{s\text{-clear}}[G, H, F|^{\mu + \zeta}]$ is IND-CPA secure. In particular, for any adversary $A$, there are adversaries $B, D$ such that

$$\text{Adv}^{\text{ind-cpa}}_{\text{OAEP}_{s\text{-clear}}, A}(k) \leq 6 \cdot \text{Adv}^{\text{xor-ind0}}_{F, G, B}(k) + 4 \cdot \text{Adv}^{\text{prg}}_{G, D}(k).$$

The running time of $B$ and $D$ are about that of $A$. 

107
Game $G_1(k)$, $G_2(k)$

- $b \leftarrow \{0, 1\}$
- $K_G \leftarrow K_G(1^k)$ ; $K_H \leftarrow K_H(1^k)$
- $(f, f^{-1}) \leftarrow Kg(1^k)$
- $pk \leftarrow (K_G, K_H, f)$
- $(M_0, M_1, state) \leftarrow A_1(1^k, pk)$
- $m_b \leftarrow M_b(1^k, pk)$ ; $r \leftarrow \{0, 1\}^\rho$
- $x \leftarrow G(K_G, r)$ ; $s \leftarrow x \oplus (m_b || 0^\zeta)$
- $z \leftarrow H(K_H, s)$ ; $t \leftarrow z \oplus r$
- $y \leftarrow f(t)$ ; $y' \leftarrow f(r)$
- $c \leftarrow (s, y)$ ; $b' \leftarrow A_2(state, c)$
- Return $(b = b')$

Game $G_3(k)$

- $b \leftarrow \{0, 1\}$
- $K_G \leftarrow K_G(1^k)$ ; $K_H \leftarrow K_H(1^k)$
- $f \leftarrow Kg(1^k)$
- $pk \leftarrow (K_G, K_H, f)$
- $(M_0, M_1, state) \leftarrow A_1(1^k, pk)$
- $m_b \leftarrow M_b(1^k, pk)$ ; $r \leftarrow \{0, 1\}^\rho$
- $x \leftarrow \{0, 1\}^{\mu + \zeta}$ ; $s \leftarrow x \oplus (m_b || 0^\zeta)$
- $y \leftarrow f(r)$ ; $c \leftarrow (s, y)$
- $b' \leftarrow A_2(state, c)$
- Return $(b = b')$

**Figure 5.28:** Games $G_1$–$G_3$ in the proof of Theorem 29.

Proof. Consider games $G_1$–$G_3$ in Figure 5.28. We now explain the game chain.

Game $G_1$ corresponds to game IND-CPA$_{\text{OAEP}}^\text{clear}$. Game $G_2$ is identical to game $G_1$, except that instead of evaluating trapdoor permutation $f$ on input $t$, we are evaluating it on input $r$. Consider adversary $B$ as shown in Figure 5.29. Note that adversary $B$ simulates game $G_1$, $G_2$ with respect to it’s inputs. It returns 1 if adversary $A$ can correctly guess the simulated challenge bit $b$, and returns 0 otherwise. Hence,

$$\Pr[G_1 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1] \leq \text{Adv}_{\text{xor-ind0}}^{\text{var}}(k).$$

Next, game $G_3$ is identical to game $G_2$, except we are using completely random $x$ in the encryption phase instead of using pseudorandom value $G(K_G, r)$. Consider adversary $C$ as shown in Figure 5.30. Hence,

$$\Pr[G_2 \Rightarrow 1] - \Pr[G_3 \Rightarrow 1] \leq \text{Adv}_{\text{hcf}}^{\text{var}}(k).$$

From Theorem 17, there are adversaries $D, E$ such that

$$\Pr[G_2 \Rightarrow 1] - \Pr[G_3 \Rightarrow 1] \leq 2 \cdot \text{Adv}_{\text{xor-ind0}}^{\text{var}}(k) + 2 \cdot \text{Adv}_{\text{prg}}^{\text{var}}(k).$$

108
We assume wlog that advantage of adversary $B$ is greater than adversary $E$. Note that $\Pr[G_3(k) \Rightarrow 1] = 1/2$, since the distribution of the ciphertexts is completely independent of bit $b$. Summing up,

$$\text{Adv}^{\text{ind-cca}}_{\text{OAEP}_{\text{clear}}^s, A}(k) \leq 6 \cdot \text{Adv}^{\text{xor-ind0}}_{F, G, B} (k) + 4 \cdot \text{Adv}^{\text{prg}}_{G, D}(k) .$$

This completes the proof.

PA0 and PA1 results. We give a full instantiation result for $s$-clear RSA-OAEP and show that it is PA0 and PA1 under suitable assumptions. We show that $s$-clear RSA-OAEP “inherits” the extractability of the underlying padding transform, in the form of PA0 and PA1, as long as the latter is also near-collision resistant. Here we
state the result for an abstract padding scheme rather than specifically for OAEP. Note that results for OAEP then follow from the round functions in Section 4.

**Theorem 30** Let $\eta, \mu, \zeta, \rho$ be integer parameters. Let $\mathcal{F}$ be a family of one-way trapdoor permutations with domain $\{0, 1\}^\rho$ and $\eta = ||Kg(1^k)||$. Let PAD be a padding transform from domain $\{0, 1\}^{\mu+\rho}$ to range $\{0, 1\}^{\mu+\zeta+\rho}$. Suppose PAD is NCR$^{\mu+\zeta}$ and $\eta$-EXT0$^{\mu+\zeta}$. Then PAD[$\mathcal{F}^{\mu+\zeta}$] is PA0 secure. In particular, for any PA0 adversary $A$, there are adversaries $A_{PAD}, B$ and extractor Ext such that for all extractors $Ext_{PAD}$

$$Adv^{pa0}_{PAD[\mathcal{F}^{\mu+\zeta}], A, Ext}(k) \leq Adv^{\eta\text{-ext0}^{\mu+\zeta}}_{PA0, A_{PAD, Ext}_{PAD}}(k) + Adv^{\eta\text{-ext}^{\mu+\zeta}}_{PA0, B}(k).$$

The running time of $A_{PAD}$ and Ext are about that of $A$ and $Ext_{PAD}$, respectively. Furthermore, the running time of $B$ is about that of $A$ plus the time to run $Ext_{PAD}$.

The proof of Theorem 30 is very similar to the proof of Theorem 31.

**Theorem 31** Let $\eta, \mu, \zeta, \rho$ be integer parameters. Let $\mathcal{F}$ be a family of one-way trapdoor permutations with domain $\{0, 1\}^\rho$ and $\eta = ||Kg(1^k)||$. Let PAD be a padding transform from domain $\{0, 1\}^{\mu+\rho}$ to range $\{0, 1\}^{\mu+\zeta+\rho}$. Suppose PAD is NCR$^{\mu+\zeta}$ and $\eta$-EXT1$^{\mu+\zeta}$. Then PAD[$\mathcal{F}^{\mu+\zeta}$] is PA1 secure. In particular, for any PA1 adversary $A$ that makes at most $q$ decryption queries, there are adversaries $B, A_{PAD}$ that makes at most $q$ extract queries and extractor Ext such that for all extractors $Ext_{PAD}$

$$Adv^{pa1}_{PAD[\mathcal{F}^{\mu+\zeta}], A, Ext}(k) \leq Adv^{\eta\text{-ext1}^{\mu+\zeta}}_{PA1, A_{PAD, Ext}_{PAD}}(k) + Adv^{\eta\text{-ext}^{\mu+\zeta}}_{PA1, B}(k).$$

The running time of $A_{PAD}$ and Ext are about that of $A$ and $Ext_{PAD}$, respectively. Furthermore, the running time of $B$ is about that of $A$ plus the time to run $Ext_{PAD}$.

Proof. Let $w$ be the randomness of PA1 adversary $A$. We define EXT1 adversary $A_{PAD}$ with randomness $w$ in Figure 5.31. Note that the auxiliary input $f$ is independent.
of the key $\pi$. Let $\text{Ext}_{PAD}$ be the corresponding extractor for $A_{PAD}$. We define PA1 extractor $\text{Ext}$ as shown in Figure 5.32.

Note that for any decryption query $c$ that adversary $A$ makes, if $c$ is not a valid ciphertext then extractor $\text{Ext}$ outputs $\perp$. Thus, adversary $A$ does not gain any information about $b$ by making invalid decryption queries. Hence, we assume wlog that adversary $A$ only makes queries for valid ciphertext $c$.

Assume $A$ makes $q$ extract queries. Let $c_i$ be the $i$-th query $A$ makes to the extract oracle and $v_i$ be the output of extractor $\text{Ext}_{PAD}$ on input $c_i^{\mu+\zeta}$. Let $W$ be the event where there exists a valid ciphertext $c_i$ such that $v_i$ is non-empty and extractor $\text{Ext}$ outputs $\perp$ on input $c_i$. Let $S$ be the event that game $\text{PA1}_{PAD,F^{\mu+\zeta}}(k)$ outputs 1. Note that all of the following probabilities are over the choice of public key $pk$ and
Randomness \( w \). Then,

\[
\text{Adv}^{\text{ncr}}_{\text{PAD}[F|\mu+\zeta],A,\text{Ext}}(k) = 2 \cdot (Pr[S \wedge W] + Pr[S \wedge \overline{W}]) - 1.
\]

Note that extractor \( \text{Ext}_{\text{PAD}} \) either outputs the correct value or \( \perp \). Now consider near-collision resistance adversary \( B \) in Figure 5.33. If event \( W \) happens, then \( B \) finds a collision. Thus, we have \( Pr[W] = \text{Adv}^{\text{ncr}}_{\text{PAD},B}(k) \). Then

\[
Pr[S \wedge W] \leq \text{Adv}^{\text{ncr}}_{\text{PAD},B}(k).
\]

Let \( E \) be the event such that for each decryption query \( c \), that adversary \( A \) makes, the output of decryption algorithm \( \text{Dec} \) and extractor \( \text{Ext} \) are equal. Then, we obtain

\[
Pr[S \wedge \overline{W}] = Pr[S \wedge \overline{W} \wedge E] + Pr[S \wedge \overline{W} \wedge \overline{E}].
\]

Note that \( W \) and \( \overline{E} \) are mutually exclusive since when \( W \) happens the output of extractor \( \text{Ext} \) would be incorrect for at least one of the extract queries. Thus, we have \( Pr[S \wedge \overline{W} \wedge E] = Pr[S \wedge \overline{E}] \).

Moreover, we have \( Pr[S \wedge E] = 1/2 \), since for all queries made by \( A \), the outputs of decryption oracle \( \text{Dec} \) and extractor \( \text{Ext} \) are equal. Hence, we obtain \( Pr[S \wedge E] = 1/2 \cdot Pr[E] \).

Consider EXT1 adversary \( A_{\text{PAD}} \) in Figure 5.31. We know \( A \) always makes valid decryption query. Thus, when event \( \overline{E} \) happens \( \text{Ext} \) outputs \( \perp \) for at least one of the

---

**Figure 5.33**: NCR adversary \( B \) in the proof of Theorem 31.
extract queries. Moreover, if \( W \) happens then for all extract queries either \( \text{Ext}_{\text{PAD}} \perp \) or \( \text{Ext} \) outputs non-empty string. Therefore, when \( E \) and \( W \) happen, extractor \( \text{Ext}_{\text{PAD}} \) fails and outputs \( \perp \) for at least one of the extract queries. Thus,

\[
\Pr [W \land E] \leq \text{Adv}_{\text{PAD}, \text{Ext}_{\text{PAD}}}(k).
\]

On the other hand, we know that \( E \) and \( W \) mutually exclusive. Hence, we get

\[
\Pr [E] = \Pr [W \lor E] - \Pr [W].
\]

Summing up,

\[
\text{Adv}_{\text{PAD}, \text{Ext}_{\text{PAD}}}(k) \leq \text{Adv}_{\text{PAD}, \text{Ext}_{\text{PAD}}}(k) + \text{Adv}_{\text{PAD}, \text{H}}(k).
\]

This completes the proof.

**PA2 Result.** We give a full instantiation result for \( s \)-clear RSA-OAEP and show that it is PA2 under stronger assumptions on \( G, H \) and \( F \). We note that we can reduce assumptions as per Theorem 18.

**Theorem 32** Let \( \eta, \mu, \zeta, \rho \) be integer parameters. Let \( F \) be a family of trapdoor permutations with domain \( \{0,1\}^\rho \). Let \( G : \mathcal{K}_G \times \{0,1\}^\rho \to \{0,1\}^{\mu + \zeta} \) and \( H : \mathcal{K}_H \times \{0,1\}^{\mu + \zeta} \to \{0,1\}^\rho \) be hash function families. Let \( \eta = ||\mathcal{K}_G(1^k)|| + ||\mathcal{K}_H(1^k)|| \).

Suppose \( G \) is VPRG\(_\zeta\), NCR\(_\zeta\) and \( \eta\text{-EXT}_2\zeta \) with respect to \( F \) and \( H \) is collision-resistant. Suppose \( F \) is XOR-NM1 and XOR-IND2\(_\zeta\) with respect to \( G \). Then \( \text{OAEP}_{s\text{-clear}}[G,H,F]^{\mu + \zeta} \) is PA2 secure. In particular, for any adversary \( A \) that makes at most \( q \) decryption queries and \( p \) encryption queries, there are extractor \( \text{Ext} \), adversaries \( B_F, B_G, B_H, C, D \), adversary \( A_G \) that makes at most \( q \) extract queries and \( p \) image queries such that for all extractor \( \text{Ext}_G \)

\[
\text{Adv}_{\text{OAEP}_{s\text{-clear}}, A, \text{Ext}}(k) \leq 3 \cdot \text{Adv}_{G,F,A,G,\text{Ext}G}(k) + 9p \cdot \text{Adv}_{F,G,C}(k)
++ 6p \cdot \text{Adv}_{G,D}(k) + 5 \cdot \text{Adv}_{G,B_G}(k)
+ 2 \cdot \text{Adv}_{H,B_H}(k) + 2p \cdot \text{Adv}_{F,G,B_F}(k)
\]
Adversary $A_G^{O_G, I}(K_G, v; w)$

$(K_H, f) \leftarrow v$; $pk \leftarrow (K_G, K_H, f)$

Run $A_D^{DSim(), EncSim} (pk; w)$

Procedure $EncSim(M)$

$(x, c) \leftarrow I(M)$

Return $c$

Procedure $DSim(c)$

$(s, y) \leftarrow c$; $r \leftarrow O_G(s|\zeta)$

$x \leftarrow G(K_G, r)$

$m^* \leftarrow s \oplus x$; $m \leftarrow m^*|\mu$

Return $m$

Figure 5.34: EXT2 adversary $A_G$ in the proof of Theorem 32.

The running time of $A_G$ is about that of $A$. The running time of $Ext$ is about that of $Ext_G$. The running time of $C$ and $D$ are about that of $A$ plus the time to run $Ext_G$. The running time of $B_F, B_G$ and $B_H$ are about that of $A$ plus the time to run $Ext$.

**Proof.** We will need Lemma 33 in our proof. We refer to [43, Lemma 7.1] for the proof of Lemma 33.

**Lemma 33** Let $\eta, \delta, \zeta$ be integer parameters. Let $G : K_G \times GDom \rightarrow GRng$ and $H : K_H \times GRng \rightarrow GDom$ be a hash function family. Let $F = (Kg, Eval, Inv)$ be a trapdoor permutation family with domain $GDom$ and $\delta = \eta + |[K_H(1^k)]|$. Suppose $G$ is $VPRG_\zeta$ and $\delta$-$EXT_2^\zeta$ function with respect to $F$. Suppose $F$ is $XOR-IND_2^\zeta$ with respect to $G$. Then $G$ is a $\eta$-$EXT_2^\zeta$ with respect to $OAEP_{s-clear}[G, H, F]^{\mu+\zeta}$. In particular, for any adversary $A$ that makes at most $q$ extract queries and $p$ image queries, there are adversaries $C, D$ and adversary $B$ that makes at most $q$ extract queries and $p$ image queries such that for all extractor $Ext$

$$Adv^{\eta-ext_2^\zeta}_{G, OAEP_{s-clear}, A, Ext}(k) \leq Adv^{\delta-ext_2^\zeta}_{G, F, B, Ext}(k) + 3p \cdot Adv^{XOR-IND_2^\zeta}_{F, G, C}(k) + 2p \cdot Adv^{VPRG_\zeta}_{G, D}(k).$$

The running time of $B$ is about that of $A$. The running time of $C$ and $D$ are about that of $A$ plus the time to run $Ext$. 

114
Let $w$ be the randomness of adversary $A$ in the PA2 game. Let $K_H$ be the key for the hash function family $H$ and $f$ be the evaluation key for the trapdoor permutation family $F$ in the game PA2. We define EXT2 adversary $A_G$ with the hint function $\text{OAEP}_{s\text{-clear}}$ and randomness $w$ in Figure 5.34. Let $v = (K_H, f)$ be the key independent auxiliary input to adversary $A_G$. Note that auxiliary input $v$ is independent of key $K_G$. Let $\text{Ext}_G$ be the corresponding extractor for $A_G$. We define PA2 extractor $\text{Ext}$ as shown in Figure 5.35. Let $c_i = (s_i, y_i)$ be the $i$-th decryption query made by $A$ and $r_i$ be the correspondence randomness. Note that when decryption query $c_i$ is invalid $\text{Ext}$ will always output $\bot$. Hence, the output of $\text{Ext}$ and decryption algorithm are always equal on invalid decryption queries. Thus, we only consider valid queries $c_i$ made by $A$.

We define $C_i$ to be the set of all ciphertext $c$ produced by encryption oracle before adversary $A$ makes the $i$-th decryption query, for all $i \in [q]$. We also define $C_{i,1}$ to be the set of all $c \in C_i$ such that $s_i|_\zeta = s|_\zeta$. Let $S$ be the event that game $\text{PA}^A_{\text{OAEP}_{s\text{-clear}}}(k)$
outputs 1 and $E$ be the event such that for all decryption query $c_i$ made by $A$, $C_{i,1}$ is empty. Then

$$\text{Adv}_\text{OAEP}^{\text{adv2}}_{\text{clear}, A, \text{Ext}}(k) = 2 \cdot (\Pr[S \land E] + \Pr[S \land \overline{E}]) - 1.$$ 

Note that using the same argument made in the proof of Theorem 31, there exists adversary $B$ such that

$$\Pr[S \land E] \leq \frac{1}{2} \cdot (\text{Adv}_\text{OAEP}^{\text{ext2}\zeta}_{g, \text{OAEP}^{\text{adv2}}_{\text{clear}, A, \text{Ext}}}(k) + \text{Adv}_\text{Gc}^{\text{cr}\zeta}_{g, B}(k) + 1).$$

Let $W$ be the event such that for at least one decryption query $c_i$, there exists $c \in C_{i,1}$ where $r_i \neq r$. Then, we obtain $\Pr[E] \leq \Pr[E \land W] + \Pr[E \land \overline{W}]$. Consider near collision resistance adversary $B_G$ in Figure 5.36. Note that, when $W$ and $E$ happen, adversary $B_G$ finds a collision. Thus,

$$\Pr[E \land W] \leq \text{Adv}_\text{Gc}^{\text{cr}\zeta}_{g, B_G}(k).$$

Moreover, let $Q$ be the event such that for at least one decryption query $c_i$, there exists $c \in C_{i,1}$ where $y_i = y$. Then, we get that $\Pr[E \land \overline{W}] \leq \Pr[E \land \overline{W} \land Q] + $
**Figure 5.37:** CR adversary $B_H$ in the proof of Theorem 32.

Consider collision resistance adversary $B_H$ in Figure 5.37. Note that, when events $E, W$ and $Q$ happen, adversary $B_H$ finds a collision. Thus,

\[
\Pr [E \land W \land Q] \leq \text{Adv}_{H, B_H}^{\text{cr}}(k).
\]

Note that when events $E, W$ and $Q$ happen, for all decryption query $c_i$ with non-empty $C_{i,1}$, we have that $r_i = r$ and $y_i \neq y$ for all $c = (s, y) \in C_{i,1}$. Let $c_j$ be the first decryption query with non-empty $C_{j,1}$. Let $R$ be the event such that for all decryption query $c_i$ with $i < j$ the output of Ext and decryption algorithm are equal. Therefore, using the same argument made in the proof of Theorem 31, there exists adversary $B$ such that

\[
\Pr [E \land W \land Q \land R] \leq \text{Adv}_{G, \text{OAEP}_{\text{clear}}, \text{AG}, \text{Ext}_G}(k) + \text{Adv}_{G, B}^{\text{n-cr}}(k).
\]

Consider XOR-NM1 adversary $B_F$ in Figure 5.38. Note that, when events $E, W, Q$ and $R$ happen, adversary $B_F$ can win the XOR-NM1 game. Thus, we obtain
Using Lemma 33,

\[
\Pr \left[ E \land W \land Q \land R \right] \leq p \cdot \text{Adv}_F^{\text{xor-nm1}}(k). \text{ Summing up, }
\]

\[
\text{Adv}_{\text{OAEP}_{\text{clear}},A,\text{Ext}}(k) \leq 3 \cdot \text{Adv}_{G,\text{OAEP}_{\text{clear}},A,G,\text{Ext}}^{\text{ext}2}(k) + 5 \cdot \text{Adv}_{G,B_\text{G}}^{\text{cr}}(k) + 2 \cdot \text{Adv}_{H,B_H}^{\text{cr}}(k) + 2p \cdot \text{Adv}_{F,G,B_F}^{\text{xor-nm1}}(k)
\]

Using Lemma 33,

\[
\text{Adv}_{\text{OAEP}_{\text{clear}},A,\text{Ext}}(k) \leq 3 \cdot \text{Adv}_{G,F,A,G,\text{Ext}}^{\text{ext}2}(k) + 9p \cdot \text{Adv}_{G,B_G}^{\text{cr}}(k) + 6p \cdot \text{Adv}_{G,D}^{\text{vprg}}(k) + 5 \cdot \text{Adv}_{G,B_G}^{\text{cr}}(k) + 2 \cdot \text{Adv}_{H,B_H}^{\text{cr}}(k) + 2p \cdot \text{Adv}_{F,G,B_F}^{\text{xor-nm1}}(k)
\]
Procedure \( K_G(1^k) \)
\[
\begin{aligned}
K_3 &\leftarrow \text{PRF}.Kg(1^k) \\
f &\leftarrow \text{ELF}.IKg(1^k) \\
K_G &\leftarrow \text{iO}(f(\text{PRF}_{K_3}(.))) \\
\text{Return } K_G
\end{aligned}
\]

Procedure \( G(K_G, x) \)
\[
\begin{aligned}
C_G &\leftarrow K_G \\
\text{Return } C_G(x)
\end{aligned}
\]

Figure 5.39: The hash function family \( \mathcal{G} \).

This completes the proof.

5.5 Full Instantiation Results (II)

In this section, we instantiate RSA-OAEP using iO and ELF. Our instantiation is obtained via a new unified paradigm of as obfuscating an extremely lossy function (ELF).\(^2\) We combine this with on prior paradigms of Brzuska and Mittelbach [35] (using point function obfuscation in the proof). Let \( \text{PRF} = (\text{PRF}.Kg, \text{PRF}.\text{Punct}, \text{PRF}.\text{Eval}) \) be a puncturable PRF and iO be an indistinguishability obfuscator for all circuits in \( \mathcal{P}/\text{poly} \). Moreover, let \( \text{ELF} = (\text{ELF}.IKg, \text{ELF}.LKg, \text{ELF}.\text{Eval}) \) be a family of extremely lossy functions. We define our hash function family \( \mathcal{G} = (K_G, G) \) in Figure 5.39.

We show in Theorem 34 that \( \text{OAEP}[\mathcal{G}, \mathcal{H}, \mathcal{F}] \) is IND-CCA secure if \( \mathcal{G} \) is instantiated as in Figure 5.39 for any function \( \mathcal{H} \). We now explain the proof idea. Let circuit \( \mathcal{C} = \text{ELF}(\text{PRF}_K(\cdot)) \). We adapt a new approach by incorporating ELF to the technique in [35] that was used to instantiate UCEs. We use AIPO to alter the circuit \( \mathcal{C} \) on input \( r^* \) to output freshly random value \( z^* \) instead of \( \text{ELF}(\text{PRF}_K(r^*)) \). We show that obfuscation of an alternative circuit is indistinguishable from the obfuscation of the

\(^2\)By obfuscating an ELF, we mean obfuscating the program that evaluates it, with the key hard-coded.
Games $G_1(k), G_2(k)$

<table>
<thead>
<tr>
<th>Games $G_1(k), G_2(k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b \leftarrow {0, 1}; K \leftarrow \text{PRF.Kg}(1^k)$</td>
</tr>
<tr>
<td>$K^* \leftarrow \text{PRF.Punct}(K, r^<em>)$; $p \leftarrow \text{AIPO}(r^</em>)$</td>
</tr>
<tr>
<td>$f \leftarrow \text{ELF.IKg}(1^k)$; $K_G \leftarrow \text{iO}(G_1[K, f])$</td>
</tr>
<tr>
<td>$K_G \leftarrow \text{iO}(C_2[K^<em>, f, p, z^</em>])$</td>
</tr>
<tr>
<td>$K_H \leftarrow K_H(1^k)$; $(F, F^{-1}) \leftarrow \text{Kg}(1^k)$</td>
</tr>
<tr>
<td>$pk \leftarrow (F, K_H, K_G)$</td>
</tr>
<tr>
<td>$(m_0, m_1, st) \leftarrow A_1^{\text{Dec}(\cdot)}(1^k)$</td>
</tr>
<tr>
<td>$r^* \leftarrow {0, 1}^\rho$; $x^* \leftarrow \text{PRF}_K(r^*)$</td>
</tr>
<tr>
<td>$z^* \leftarrow f(x^<em>)$; $s^</em> \leftarrow z^* \oplus 0^k</td>
</tr>
<tr>
<td>$y^* \leftarrow H(K_H, s^<em>)$; $t^</em> \leftarrow r^* \oplus y^*$</td>
</tr>
<tr>
<td>$c^* \leftarrow F(s^*</td>
</tr>
<tr>
<td>Return $(b = b')$</td>
</tr>
</tbody>
</table>

Games $G_3(k), G_4(k)$

<table>
<thead>
<tr>
<th>Games $G_3(k), G_4(k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b \leftarrow {0, 1}$; $K \leftarrow \text{PRF.Kg}(1^k)$</td>
</tr>
<tr>
<td>$K^* \leftarrow \text{PRF.Punct}(K, r^<em>)$; $p \leftarrow \text{AIPO}(r^</em>)$</td>
</tr>
<tr>
<td>$f \leftarrow \text{ELF.IKg}(1^k)$; $K_G \leftarrow \text{iO}(C_2[K^<em>, f, p, z^</em>])$</td>
</tr>
<tr>
<td>$K_H \leftarrow K_H(1^k)$; $(F, F^{-1}) \leftarrow \text{Kg}(1^k)$</td>
</tr>
<tr>
<td>$pk \leftarrow (F, K_H, K_G)$</td>
</tr>
<tr>
<td>$(m_0, m_1, st) \leftarrow A_1^{\text{Dec}(\cdot)}(1^k)$</td>
</tr>
<tr>
<td>$r^* \leftarrow {0, 1}^\rho$; $x^* \leftarrow \text{PRF.Rng}(k)$</td>
</tr>
<tr>
<td>$z^* \leftarrow f(x^<em>)$; $z^</em> \leftarrow \text{GRng}(k)$</td>
</tr>
<tr>
<td>$s^* \leftarrow z^* \oplus 0^k</td>
</tr>
<tr>
<td>$y^* \leftarrow H(K_H, s^<em>)$; $t^</em> \leftarrow r^* \oplus y^*$</td>
</tr>
<tr>
<td>$c^* \leftarrow F(s^*</td>
</tr>
<tr>
<td>Return $(b = b')$</td>
</tr>
</tbody>
</table>

Circuit $C_1[K, f](r)$

Return $f(\text{PRF}_K(r))$

Circuit $C_2[K^*, f, p, z^*](r)$

If $p(r) = \bot$ then return $f(\text{PRF}_{K^*}(r))$

Return $z^*$

**Figure 5.40: Games $G_1$–$G_4$ in the proof of Theorem 34.**

The original circuit, using differing input obfuscation given the auxiliary information of the differing point $r^*$. However, in order to do so adversary attacking diO need to simulate the decryption oracle for IND-CCA adversary. We do this by using the property of ELF. Considering the running time of the IND-CCA adversary, we switch to the proper extremely lossy mode of ELF function that is indistinguishable to the adversary. We note that once we are in lossy mode we can answer decryption queries by going over all possible output of the extremely lossy function. Now that $z^*$ is uniformly random we conclude that ciphertext $c^*$ looks uniformly random.

**Theorem 34** Let $n, \mu, \zeta, \rho$ be integer parameters. Let $\mathcal{G} : \mathcal{K}_G \times \{0, 1\}^\rho \rightarrow \{0, 1\}^{\mu+\zeta}$ and $\mathcal{H} : \mathcal{K}_H \times \{0, 1\}^{\mu+\zeta} \rightarrow \{0, 1\}^\rho$ be a hash function family. Let $\mathcal{F}$ be a family of trapdoor permutations with domain $\{0, 1\}^n$, where $n = \mu + \zeta + \rho$. Assuming ELF is a secure ELF, PRF is a secure PRF, iO is a secure indistinguishability obfuscation, and
assuming $\text{AIPO}$ exists. Moreover, assuming $\mathcal{F}$ is $(\mu + \zeta)$-$\text{OW}$, $\zeta$-$\text{NM}$ and $(\eta, \zeta)$-$\text{NBB-SIE2}$. Then if $\mathcal{G}$ is instantiated as above, $\text{OAEP}[\mathcal{G}, \mathcal{H}, \mathcal{F}]$ is IND-CCA-KI secure.

Proof. Consider games $G_1$–$G_8$ in Figures 5.40 and 5.41. We now explain the game chain.

Game $G_1$: This is the standard indistinguishability chosen ciphertext game. Suppose adversary $A$ runs in time $s$ and wins game $G_1$ with probability $\epsilon$ where $\epsilon$ is non-negligible. Let $\delta$ be an inverse polynomial such that $\epsilon \geq \delta$ infinitely often.

Game $G_2$: Game $G_2$ is similar to game $G_1$ except that we puncture the PRF key $K$ at $r^*$. Moreover, the hash key $K_G$ does not consist of an obfuscation of $C_1[K, f]$, but rather of an obfuscation of the circuit $C_2[K^*, f, p, z^*]$. Note that the two

<table>
<thead>
<tr>
<th>Games $G_5(k), G_6(k)$</th>
<th>Games $G_7(k), G_8(k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b \leftarrow {0, 1}$ ; $K \leftarrow \text{PRF.Kg}(1^k)$</td>
<td>$b \leftarrow {0, 1}$ ; $\text{coin} \leftarrow \text{Coins}(k)$</td>
</tr>
<tr>
<td>$p \leftarrow \text{AIP}(r^*); f \leftarrow \text{ELF.LKg}(1^k)$</td>
<td>$K \leftarrow \text{PRF.Kg}(1^k); f \leftarrow \text{ELF.LKg}(1^k)$</td>
</tr>
<tr>
<td>$f \leftarrow \text{ELF.LKg}(1^k); K_G \leftarrow \text{iO}(C_2[K, f, p, z^*])$</td>
<td>$p \leftarrow \text{AIP}(r^<em>); K_G \leftarrow \text{iO}(C_2[K, f, p, z^</em>])$</td>
</tr>
<tr>
<td>$K_H \leftarrow K_H(1^k); (F, F^{-1}) \leftarrow \text{Kg}(1^k)$</td>
<td>$K_G \leftarrow \text{iO}(C_1[K, f]); K_H \leftarrow K_H(1^k)$</td>
</tr>
<tr>
<td>$pk \leftarrow (F, K_H, K_G)$</td>
<td>$(F, F^{-1}) \leftarrow \text{Kg}(1^k); pk \leftarrow (F, K_H, K_G)$</td>
</tr>
<tr>
<td>$(m_0, m_1, st) \leftarrow A_{1}^{\text{Dec}()}(1^k)$</td>
<td>$(m_0, m_1, st) \leftarrow A_{1}^{\text{Dec}()}(1^k; \text{coin})$</td>
</tr>
<tr>
<td>$r^* \leftarrow {0, 1}^\rho; x^* \leftarrow \text{PRF.Rng}(k)$</td>
<td>$r^* \leftarrow {0, 1}^\rho; x^* \leftarrow \text{PRF.Rng}(k)$</td>
</tr>
<tr>
<td>$z^* \leftarrow \text{GRng}(k); s^* \leftarrow z^* \oplus 0^s \parallel m_b$</td>
<td>$z^* \leftarrow \text{GRng}(k); s^* \leftarrow z^* \oplus 0^s \parallel m_b$</td>
</tr>
<tr>
<td>$y^* \leftarrow H(K_H, s^<em>)$ ; $t^</em> \leftarrow r^* \oplus y^*$</td>
<td>$y^* \leftarrow H(K_H, s^<em>)$ ; $t^</em> \leftarrow r^* \oplus y^*$</td>
</tr>
<tr>
<td>$c^* \leftarrow F(s^* \parallel t^<em>)$ ; $b' \leftarrow A_{2}^{\text{Dec}()}(st, pk, c^</em>)$</td>
<td>$c^* \leftarrow F(s^* \parallel t^<em>)$ ; $b' \leftarrow A_{2}^{\text{Dec}()}(st, pk, c^</em>; \text{coin})$</td>
</tr>
<tr>
<td>Return $(b = b')$</td>
<td>Return $(b = b')$</td>
</tr>
</tbody>
</table>
circuits are functionally equivalent. Therefore, considering the adversary $D_1$ in Figure 5.42, we get that $\Pr[G_1 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1] \leq \text{Adv}_{io,D_1,C}^\text{io}(k)$.

**Game $G_3$:** Game $G_3$ is similar to game $G_2$ except that $x^*$ is chosen randomly in PRF.Rng($k$). Considering the adversary $D_2$ attacking pseudorandom function PRF at the punctured points, we get that $\Pr[G_2 \Rightarrow 1] - \Pr[G_3 \Rightarrow 1] \leq \text{Adv}_{PRF,D_2}^\text{pprf}(k)$. We skip the code of adversary $D_2$ for simplicity.

**Game $G_4$:** Game $G_4$ is similar to game $G_3$ except that $z^*$ is chosen randomly in GRng($k$). Considering the adversary $D_3$ distinguishing the output of ELF from uniformly random, we get that $\Pr[G_3 \Rightarrow 1] - \Pr[G_4 \Rightarrow 1] \leq \text{Adv}_{ELF,D_3}^\text{elf}(k)$. We skip the code of adversary $D_3$ for simplicity.

**Game $G_5$:** Game $G_5$ is similar to game $G_4$ except that an obfuscation of circuit $C_2[K,f,p,z^*]$ is used as the hash key $K_G$. Note that circuit $C_2[K,f,p,z^*]$ is identical to circuit $C_2[K^*,f,p,z^*]$, except that it uses the original PRF key $K$ instead of the punctured key $K^*$. The two circuits are functionally equivalent.
Procedure $\text{Dec}(c)$

For all $z \in [f(\cdot)]$ do

$s\|t \leftarrow \mathcal{O}(z; c)$

$r \leftarrow t \oplus H(K_H, s)$

$\overline{m} \leftarrow G(K_G, r) \oplus s$ ; $m \leftarrow \overline{m}|\mu$

If $\text{Enc}(pk, m; r) = c$ then return $m$

Return $\bot$

Therefore, considering the adversary $D_4$ attacking $\text{iO}$, we get that $\Pr [ G_4 \Rightarrow 1 ] - \Pr [ G_5 \Rightarrow 1 ] \leq \text{Adv}_{\text{iO}, D_4, \mathcal{C}}(k)$. We skip the code of adversary $D_4$ and note that it is very similar to the adversary $D_1$.

**Game $G_6$:** Game $G_6$ is similar to game $G_5$ except that we change $\text{ELF}$ to be lossy mode. That is, we generate $f \leftarrow \text{ELF.LKg}(1^k, \lambda(s, 2/\delta))$. Considering the adversary $D_5$ attacking lossiness of $\text{ELF}$ running in time $s$, we get that $\Pr [ G_5 \Rightarrow 1 ] - \Pr [ G_6 \Rightarrow 1 ] \leq \delta/2$.

**Game $G_7$:** Game $G_7$ is similar to game $G_6$ except that in the decryption oracle we run the extractor for TDP $\mathcal{F}$ with all point in the range of the ELF function $f$ to decrypt the challenge ciphertexts. Let $c_i$ be the decryption query that adversary $A$ makes. We note that either $F^{-1}(c_i)|\zeta = s^*|\zeta$ or there exists $z \in [f(\cdot)]$ such that $F^{-1}(c_i)|\zeta = z|\zeta$. Consider the adversary $D_6$ attacking non-malleability of $\mathcal{F}$, and the fact that $(i\mathcal{O}(C_2[K, f, p, z^*], \alpha))$ and $(i\mathcal{O}(C_1[K, f], \alpha))$
are indistinguishable (shown in game $G_8$) where $\alpha = (\text{coin}, c^*, K_H, F, b)$, we obtain that $\Pr\left[ F^{-1}(c_1) | ^c = s^* | ^c \right] \leq \text{Adv}_{F,D_6}^{ur-mm}(k) + \text{Adv}_{iO,D_7,Samp}^{dio}(k)$.

Moreover, when there exists $z \in [f(\cdot)]$ such that $F^{-1}(c_1) | ^c = z | ^c$, we show that output of $\text{Dec}$ and $\overline{\text{Dec}}$ are equal with all but negligible probability. Let $aux = (m_0, m_1, K_H, b, f, K, \text{coin})$. Considering EXT2 adversary $B$ and distribution $D$ in Figure 5.43, we obtain that output of $\text{Dec}$ and $\overline{\text{Dec}}$ are equal except with probability $\text{Adv}_{F,D,B,\text{Ext}}^{(\eta,\mu)-ext2}(k)$. Therefore, we obtain that $\Pr [ G_6 \Rightarrow 1 ] - \Pr [ G_7 \Rightarrow 1 ] \leq \text{Adv}_{F,D,B,\text{Ext}}^{(\eta,\mu)-ext2}(k) + \text{Adv}_{F,D_6}^{ur-mm}(k) + \text{Adv}_{iO,D_7,Samp}^{dio}(k)$.

**Game $G_8$:** Game $G_8$ is similar to game $G_7$ except that the hash key $K_G$ is consist of an obfuscation of circuit $C_1[K,f]$. Note that Circuit $C_1[K,f]$ is the original circuit. Observe that the circuits $C_2[K,f,p,z^*]$ and $C_1[K,f]$ only differ on points where $p(r)$ is not equal to $\perp$, that is, they differ on a single point, which is the query point $r^*$. We will bound the difference between games $G_7$ and $G_8$ by the differing-inputs security of the indistinguishability obfuscator $iO$. Now, consider the differing-inputs circuit sampler $\text{Samp}$ in Figure 5.44. We show in Lemma 35 that family of circuit pairs $(C_2[K,f,p,z^*],C_1[K,f],\text{Samp})$ is differing-inputs. Considering the adversary $D_7$ attacking differing-inputs obfuscator $iO$ in Figure 5.44, we get that $\Pr [ G_7 \Rightarrow 1 ] - \Pr [ G_8 \Rightarrow 1 ] \leq \text{Adv}_{iO,D_7,Samp}^{dio}(k)$.

Observe that adversary $A$ running in time $s$ wins in game $G_8$ with probability at least $\delta/2 - \neg(k)$. Note that this quantity is at least $\delta/3$ infinitely often, and is therefore non-negligible. However, we know that ciphertext $c^*$ in game $G_8$ is independent of bit $b$. Therefore, advantage of adversary $A$ winning in game $G_8$ is zero, which contradicts. Hence, there are no PPT adversary that can win game $G_1$ with non-negligible probability. This completes the proof of Theorem 34.
Distribution $\text{Samp}(1^k)$
\[ r^* \leftarrow \{ 0, 1 \}^\rho \quad z^* \leftarrow \text{GRng}(k) \]
coin $\leftarrow \text{Coins}(k) \quad K \leftarrow \text{PRF.Kg}(1^k)$
p $\leftarrow \text{AIPO}(r^*) \quad f \leftarrow \text{ELF.LKg}(1^k)$
\[ C^1 \leftarrow C_1[K, f] \quad C^2 \leftarrow C_2[K, f, p, z^*] \]
\[ K_H \leftarrow K_H(1^k) \quad (F, F^{-1}) \leftarrow \text{Kg}(1^k) \]
b $\leftarrow \{ 0, 1 \} \quad (m_0, m_1) \leftarrow A_1^{\text{Dec}'}(1^k; \text{coin})$
\[ s^* \leftarrow z^* \oplus 0^c \parallel m_0 \quad y^* \leftarrow H(K_H, s^*) \]
\[ t^* \leftarrow r^* \oplus y^* \quad c^* \leftarrow F(s^* \parallel t^*) \]
Return $(C^1, C^2, (\text{coin}, c^*, K_H, F, b))$

Adversary $D_7(C^*, (\text{coin}, c^*, K_H, F, b))$
\[ K_G \leftarrow C^* \quad pk \leftarrow (F, K_H, K_G) \]
b' $\leftarrow A_2^{\text{Dec}(\cdot)}(pk, c^*; \text{coin})$
Return ($b = b'$)

Procedure $\text{Dec}(c)$
\[ \text{aux} \leftarrow (m_0, m_1, K_H, b, f, K, \text{coin}) \]
For all $z \in [f(\cdot)]$ do
\[ s \parallel t \leftarrow \text{Ext}(z|c, \text{aux}, K_G, c^*) \]
r $\leftarrow t \oplus H(K_H, s)$
\[ \overline{m} \leftarrow G(K_G, r) \oplus s \quad m \leftarrow \overline{m}|_\mu \]
If $\text{Enc}(pk, m; r) = c$ then return $m$
Return $\bot$

Figure 5.44: Circuit sample $\text{Samp}$ (left) and adversary $D_7$ (right) in the proof of Theorem 34.

Lemma 35 If AIPO is a secure AIPO obfuscator then the family of circuit pairs $(C_2[K, f, p, z^*], C_1[K, f], \text{Samp})$ is differing-inputs.

Proof. Let $B$ be an adversary against the differing-inputs of the above circuit family which receives as input $(C_2[K, f, p, z^*], C_1[K, f], \text{aux})$ and outputs a value $\alpha$ such that $C_2[K, f, p, z^*](\alpha) = C_1[K, f](\alpha)$, where $\text{aux} = (\text{coin}, c^*, K_H, F, b)$. Then, we show that we can build an adversary against AIPO using adversary $B$. Consider following distribution $D$ and adversary $A$ attacking AIPO in Figure 5.45.

We start by showing that $D$ is an unpredictable distribution. Note that if there exist an adversary that outputs $r^*$ on input $L$, we can build an LROW adversary against trapdoor permutation $F$ that on input $s^*, c^*$ outputs $t^*$. Thus, distribution $D$ is unpredictable. Next, we note that the probability of adversary $B$ succeeds against differing-inputs distribution $\text{Samp}$ is bounded by $\text{Adv}_{\text{AIPO}, A, D}(k)$. We skip the details and note that it is similar to the proof of Claim 3.4 of [35]. This completes the proof from Lemma 35.
5.6 Discussion and Perspective

We summarize and compare our results to prior work in Figure 5.46. Note that we get a lot of mileage from assuming the trapdoor permutation is specifically RSA, whereas prior work, which has mostly shown negative results CCA-style security notions, went for a general approach. We also highlight that while our assumptions on both RSA and the round functions for our full instantiability results are expectedly stronger than what we need for partial instantiations, they still compare favorably to prior work. In particular, while our assumption of EXT2 for \( G \) in our s-clear result is already “PA2-flavored,” prior work such as [26] made CCA-style assumptions on the round functions even to obtain relatively weak notions of non-malleability. It can be viewed as a strengthening of “adaptive” (CCA-style) security notions on one-way functions [75, 83]. Indeed, [83] already advocated of making strong but standard-model assumptions satisfied by a RO to resolve very different problems, and in a way

\[ \text{Figure 5.45: Distribution } D(1^k) \text{ (left) and adversary } A \text{ (right) in the proof of Lemma 35.} \]

\[ \text{Adversary } A(1^k, L, p) \]
\[ (t, d, K, f, z^*, \text{coin}, c^*, K_H, F, b) \leftarrow L \]
\[ \text{aux } \leftarrow (\text{coin}, c^*, K_H, F, b) \]
\[ \alpha \leftarrow B(C_2[K, f, p, z^*], C_1[K, f], \text{aux}) \]
\[ \text{If } \alpha = \bot \text{ then } d' \leftarrow \{0, 1\} \]
\[ \text{Else If } \langle t, \alpha \rangle = d \text{ then } d' \leftarrow 1 \]
\[ \text{Else } d' \leftarrow 0 \]
\[ \text{Return } d' \]

\[ \text{Distribution } D(1^k) \]
\[ r^* \leftarrow \{0, 1\}^\rho; \quad z^* \leftarrow \text{GRng}(k) \]
\[ \text{coin } \leftarrow \text{Coins}(k) \]
\[ f \leftarrow \text{ELF.LKg}(1^k); \quad K \leftarrow \text{PRF.Kg}(1^k) \]
\[ K_H \leftarrow K_H(1^k); \quad (F, F^{-1}) \leftarrow \text{Kg}(1^k) \]
\[ b \leftarrow \{0, 1\}; \quad (m_0, m_1) \leftarrow A_{\text{Dec}}(1^k; \text{coin}) \]
\[ s^* \leftarrow z^* \oplus 0^\rho \parallel m_0; \quad y^* \leftarrow H(K_H, s^*) \]
\[ t^* \leftarrow r^* \parallel y^*; \quad c^* \leftarrow F(s^* \parallel t^*) \]
\[ t \leftarrow \{0, 1\}^\rho; \quad d \leftarrow \langle t, r^* \rangle \]
\[ L \leftarrow (t, d, K, f, z^*, \text{coin}, c^*, K_H, F, b) \]
\[ \text{Return } (L, r^*) \]
### Scheme Assumptions on OAEP Assumptions on $F$ Security Size Ref

<table>
<thead>
<tr>
<th></th>
<th>G : PRG and $H$ : RO</th>
<th>OW, SIE and CIE</th>
<th>IND-CCA2</th>
<th>$n$</th>
<th>Section 5.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA-OAEP</td>
<td>G : RO and $H$ : PHCF</td>
<td>OW, SIE and CIE</td>
<td>IND-CCA2</td>
<td>$n$</td>
<td>Section 5.2</td>
</tr>
<tr>
<td>RSA-OAEP</td>
<td>$G$ : t-wise independent</td>
<td>Lossy TDP</td>
<td>IND-CPA</td>
<td>$n$</td>
<td>[76]</td>
</tr>
<tr>
<td>RSA-OAEP</td>
<td>$G$ : PRG, EXT0 and NCR</td>
<td>$H$ : HCF, EXT0 and CR</td>
<td>$\text{IND-CCA0-KI}$</td>
<td>$3n + 3k$</td>
<td>Section 5.4.3</td>
</tr>
<tr>
<td>RSA-OAEP</td>
<td>G : EXT1 and NCR</td>
<td>$G$ : PRG and $H$ : HCF</td>
<td>$\text{IND-CCA1-KI}$</td>
<td>$3n + 3k$</td>
<td>Section 5.4.3</td>
</tr>
<tr>
<td>RSA-OAEP</td>
<td>$G$ : PRG and NCR</td>
<td>$H$ : RO</td>
<td>IND-CCA2</td>
<td>$n + k$</td>
<td>[26]</td>
</tr>
<tr>
<td>RSA-OAEP</td>
<td>$G$ : RO</td>
<td>OW</td>
<td>IND-CCA2</td>
<td>$n + k$</td>
<td>[26]</td>
</tr>
<tr>
<td>RSA-OAEP</td>
<td>$G$ : PRG and NCR</td>
<td>$H$ : NM PRG with hint</td>
<td>$\text{NM-CPA}$</td>
<td>$n + k$</td>
<td>[26]</td>
</tr>
<tr>
<td>RSA-OAEP</td>
<td>$G$ : PRG, EXT1 and NCR</td>
<td>$XOR-IND0$</td>
<td>IND-CCA1</td>
<td>$2n + k + \mu$</td>
<td>Section 5.4.4</td>
</tr>
<tr>
<td>RSA-OAEP</td>
<td>$G$ : PRG, EXT2 and NCR</td>
<td>$H$ : CR and XOR-NM0</td>
<td>IND-CCA2</td>
<td>$2n + k + \mu$</td>
<td>Section 5.4.4</td>
</tr>
</tbody>
</table>

**Figure 5.46:** Instantiability results for RSA-OAEP sorted by scheme variant, where $n$ is modulus length, $k$ is the security parameter, and $\mu$ is message length. Typically $n = 2048$, $k = 128$, and $\mu = 128$.

we follow in their footsteps. Plus, it is not clear how to get an IND-CCA2 encryption scheme from EXT2 functions in a simpler way.
In this chapter, we show new instantiation results under chosen-ciphertext security for slightly modified Fujisaki-Okamoto transform. Fujisaki-Okamoto transform takes a public-key encryption scheme and a symmetric-key encryption scheme, and produces a new public-key encryption scheme. The encryption algorithm has the form

\[ E_{pk}^{hy}(m; r) = E_{pk}^{asy}(r; H(r)) || E_{K}^{sy}(m) \quad \text{where} \quad K = G(r). \]

Unfortunately, FO transform was shown uninstantiable by Brzuska et al. [35]. More generally they showed uninstantiability of all “admissible” encryption transforms. In particular, when the public-key scheme is allowed arbitrary (but IND-CCA), FO is admissible regardless of the class of symmetric-key schemes considered.

6.1 Our Technique

We consider a slightly modified FO transform and show IND-CCA security in standard model. We give two separate instantiations for two different slightly tweaked FO transforms. We note that the changes that we make to the FO transform are conservative. For our first instantiation, we make use of the extractable functions to instantiate \( H \) while we model \( G \) as one-wayness extractor. In our second instantiation result, we use an indistinguishably obfuscator, puncturable PRF \( \text{PRF} \), as well as an extremely lossy function \( \text{ELF} \). Additionally, in the proof, we use an auxiliary-input
point-function obfuscator. At a high-level, to instantiate \( H \) we use the composite function \( \text{PRF}_K(\text{PRG}(\cdot)) \) and to instantiate \( G \) we use the composite function \( \text{ELF}(\text{PRF}_{K'}(\cdot)) \).

### 6.2 Fujisaki-Okamoto Transform Instantiation (I)

In this section, we slightly change the original FO transform and give a new transform which we call \( \overline{\text{FO}} \). Next we instantiate the new transform \( \overline{\text{FO}} \) using extractable functions. Let \( \text{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D}) \) be a private-key encryption, \( \text{PKE} = (\mathcal{K}_g, \mathcal{E}, \mathcal{D}) \) be a public-key encryption schemes, and \( \mathcal{F} = (\mathcal{K}_g, \mathcal{E}, \mathcal{I}_n) \) be a trapdoor permutation family. Moreover, let \( \mathcal{H} : \mathcal{K}_H \times \text{HDom} \to \text{HRng} \) and \( \mathcal{G} : \mathcal{K}_G \times \text{GDom} \to \text{GRng} \) be function families. We define \( \overline{\text{FO}}_{F,H,G}[\text{PKE}, \text{SE}] = (\overline{\text{FO}}.\text{Dec}, \overline{\text{FO}}.\text{Enc}, \overline{\text{FO}}.\text{Dec}) \) in Figure 6.1.

We now instantiate \( \overline{\text{FO}}_{F,H,G}[\text{PKE}, \text{SE}] \) transform using extractable functions as follows.

**Theorem 36** Assuming \( \mathcal{F} \) is a one-way trapdoor permutation family, \( \mathcal{H} \) is a hardcore function for \( \mathcal{F} \) and \( \eta\)-\text{EXT2}, and \( \mathcal{G} \) is one-wayness extractor. Moreover, assuming \( \text{PKE} \) is uniquely randomness recovering, and \( \text{SE} \) is IND-CPA and INT-CTXT secure. Then \( \overline{\text{FO}} \) defined as above is IND-CCA secure.
Games $G_1(k)$

\[
\begin{align*}
(pk', sk') & \leftarrow Kg(1^k) ; (f, f^{-1}) \leftarrow Kg(1^k) \\
K_H & \leftarrow Kg_H(1^k) ; K_G \leftarrow Kg_G(1^k) \\
pk & \leftarrow (pk', f, K_H, K_G) \\
(M_0, M_1, st) & \leftarrow A_1^{\text{Dec}}(1^k, pk) \\
b & \leftarrow \{0, 1\} ; m_b & \leftarrow M_0(1^k, pk) \\
r^* & \leftarrow \text{HDom}(k) ; c_1^* & \leftarrow \text{Enc}(pk', f(r^*) ; h^*) \\
K^* & \leftarrow G(K_G, r^*) ; c_2^* & \leftarrow \mathcal{E}_{K^*}(m_b) \\
d & \leftarrow A_2^{\text{Dec}}(st, (c_1^*, c_2^*)) \\
\text{Return } (b = d)
\end{align*}
\]

Procedure $\text{Dec}(c)$

\[
\begin{align*}
hint & \leftarrow (f, K_G, f(r^*), K^*) ; \text{aux} & \leftarrow (b, pk', sk') \\
(c_1, c_2) & \leftarrow c ; h & \leftarrow \text{Rec}(sk', c_1) \\
r & \leftarrow \text{Ext}(K_H, \text{aux}, \text{coin}, \text{hint}, h^*, h) \\
K & \leftarrow G(K_G, r) ; m & \leftarrow D_K(c_2) \\
\text{Return } m
\end{align*}
\]

Games $G_2(k), G_3(k)$

\[
\begin{align*}
(pk', sk') & \leftarrow Kg(1^k) ; (f, f^{-1}) \leftarrow Kg(1^k) \\
K_H & \leftarrow Kg_H(1^k) ; K_G \leftarrow Kg_G(1^k) \\
pk & \leftarrow (pk', f, K_H, K_G) ; \text{coin} & \leftarrow \text{Coins} \\
(M_0, M_1, st) & \leftarrow A_1^{\text{Dec}}(1^k, pk; \text{coin}) \\
b & \leftarrow \{0, 1\} ; m_b & \leftarrow M_0(1^k, pk; \text{coin}) \\
r^* & \leftarrow \text{HDom}(k) ; c_1^* & \leftarrow \text{Enc}(pk', f(r^*) ; h^*) \\
K^* & \leftarrow G(K_G, r^*) ; K^* & \leftarrow \text{GRng}(k) \\
c_2^* & \leftarrow \mathcal{E}_{K^*}(m_b) ; d & \leftarrow A_2^{\text{Dec}}(st, (c_1^*, c_2^*); \text{coin}) \\
\text{Return } (b = d)
\end{align*}
\]

Figure 6.2: Games $G_1$–$G_3$ in the proof of Theorem 36.

\[\]

**Proof.** We prove security through a sequence of games. Consider games $G_1$–$G_3$ in Figure 6.2. We now explain the game chain. Game $G_1$ is the standard indistinguishability chosen ciphertext (IND-CCA) game. Thus, we have that $\text{Adv}^{\text{ind-cca}}_{\text{FO}, A}(k) = 2 \cdot \text{Pr}[G_1 \Rightarrow 1] - 1$ for any PPT adversary $A$.

Game $G_2$ is similar to game $G_1$ except that we change the decryption oracle as follows. We first run randomness recovery algorithm $\text{Rec}$ on inputs $c_1$ and secret key $sk'$ to obtain $h$. Then we use the extractor for the hash function family $\mathcal{H}$ to extract the randomness $r$ and decrypt $c_2$ using symmetric key $G(K_G, r)$. Consider EXT2 adversary $B$ in Figure 6.3. Let $\text{Ext}$ be an extractor for adversary $B$. We note that $\text{hint}$ given to adversary $B$ by image oracle $\mathcal{I}$ is uninvertible, since $\mathcal{G}$ is a one-wayness extractor and $\mathcal{F}$ is one-way.
<table>
<thead>
<tr>
<th><strong>Adversary</strong> $B^O I (K_H, aux; coin)$</th>
<th><strong>Procedure</strong> Dec$(c)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(b, pk', sk') \leftarrow aux; (\text{hint}, h^*) \leftarrow I(1^k)$</td>
<td>$(c_1, c_2) \leftarrow c$</td>
</tr>
<tr>
<td>$(f, K_G, f(r^*)) \leftarrow \text{hint}$</td>
<td>$h \leftarrow \text{Rec}(sk', c_1)$</td>
</tr>
<tr>
<td>$pk \leftarrow (pk', f, K_H, K_G)$</td>
<td>$r \leftarrow O(h)$</td>
</tr>
<tr>
<td>$(M_0, M_1, st) \leftarrow A_1^{\text{Dec}(i)}(pk; coin)$</td>
<td>$K \leftarrow G(K_G, r)$</td>
</tr>
<tr>
<td>$m_b \leftarrow M_0(1^k; pk; coin)$</td>
<td>$m \leftarrow D_K(c_2)$</td>
</tr>
<tr>
<td>$c_1^* \leftarrow \text{Enc}(pk', f(r^<em>); h^</em>)$</td>
<td>Return $m$</td>
</tr>
<tr>
<td>$K^* \leftarrow G(K_G, r^<em>)$; $c_2^</em> \leftarrow \mathcal{E}_K^r(m_b)$</td>
<td></td>
</tr>
<tr>
<td>Run $A_2^{\text{Dec}(i)}(st, (c_1^<em>, c_2^</em>); coin)$</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 6.3:** EXT2 adversary $B$ in the proof of Theorem 36.

Let $c_i = (c_{i,1}, c_{i,2})$ be the $i$-th decryption query that adversary $A$ makes, where $c_{i,1} = \text{Enc}(pk', f(r_i); h_i)$ and $c_{i,2} = \mathcal{E}_{G(K_G, r_i)}(m)$. Note that for all queries $c_i \neq c^*$ if we have $h_i = h^*$ then extractor $\text{Ext}$ fails. Otherwise extractor $\text{Ext}$ can successfully extract $r_i$. Thus, we need to bound the probability of $h_i = h^*$ for any $i \in [p]$ where $p$ is the number of decryption queries that adversary $A$ makes. Note that we have

$$\Pr[h_i = h^*] = \Pr[h_i = h^* \land f(r_i) = f(r^*)] + \Pr[h_i = h^* \land f(r_i) \neq f(r^*)] .$$

Observe that when adversary $A$ makes a decryption query $c_i \neq c^*$ such that $h_i = h^*$ and $f(r_i) = f(r^*)$, we are able to construct INT-CTX adversary $B_1$ attacking symmetric key encryption $\text{SE}$. Thus, we obtain $\Pr[h_i = h^* \land f(r_i) = f(r^*)] \leq p \cdot \text{Adv}_{\text{SE}, B_1}^{\text{INT-CTX}}(k)$. On the other hand, when adversary $A$ makes a decryption query $c_i \neq c^*$ such that $h_i = h^*$ and $f(r_i) \neq f(r^*)$, we are able to construct adversary $B_2$ attacking function family $\mathcal{H}$ that can successfully find collisions. Hence, we obtain that $\Pr[h_i = h^* \land f(r_i) \neq f(r^*)] \leq \text{Adv}_{\mathcal{H}, B_2}^{\text{EXT}}(k)$. Summing up, we obtain that $\Pr[G_1 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1] \leq p \cdot \text{Adv}_{\text{SE}, B_1}^{\text{INT-CTX}}(k) + \text{Adv}_{\mathcal{H}, B_2}^{\text{CR}}(k) + \text{Adv}_{\mathcal{H}, \mathcal{B}, \text{Ext}}^{\text{EXT}}(k)$.

Game $G_3$ is similar to game $G_2$ except that $K^*$ is chosen at random in $\text{GRng}(k)$. Consider distribution $D^1 = \{D^1_k\}_{k \in \mathbb{N}}$ such that $D^1_k$ outputs $(z, r^*)$ where $r^*$ is chosen
uniformly random from domain $G\text{Dom}(k)$ and $z = (f, K_H, f(r^*), h^*)$ for $f \leftarrow \mathcal{K}g(1^k)$, $K_H \leftarrow \mathcal{K}_H(1^k)$, and $h^* = H(K_H, r^*)$. We note that $D^1$ is unpredictable since $F$ is one-way and $H$ is a hardcore function for $F$. Now, consider adversary $C$ attacking one-wayness extractor $\mathcal{G}$ in Figure 6.4. Then, we obtain that $\Pr[\mathcal{G}_2 \Rightarrow 1] - \Pr[\mathcal{G}_3 \Rightarrow 1] \leq 2 \cdot \text{Adv}_{\text{cdist}}^{\mathcal{G}, C, D^1}$. Finally in game $G_3$, we have that $K^*$ is uniformly random and independent of $c^*_1$. Then, considering IND-CPA adversary $D$ attacking $\text{SE}$, we can show that $\text{Adv}_{\text{SE}, D}^{\text{ind-cpa}}(k) = 2 \cdot \Pr[\mathcal{G}_3 \Rightarrow 1] - 1$. We omit the code of adversary $D$ for simplicity.

Summing up,

$$\text{Adv}_{\text{FO}, A}^{\text{ind-cpa}}(k) \leq 2p \cdot \text{Adv}_{\text{SE}, B_1}^{\text{int-ctxt}}(k) + 2 \cdot \text{Adv}_{H, B_2}^{\text{cr}}(k)$$

$$+ 2 \cdot \text{Adv}_{H, B, \text{Ext}}^{\text{ext2}}(k) + 2 \cdot \text{Adv}_{\mathcal{G}, C, D_1}^{\text{dist}} + \text{Adv}_{\text{SE}, D}^{\text{ind-cpa}}(k).$$

This completes the proof of Theorem 36.

### 6.3 Fujisaki-Okamoto Transform Instantiation (II)

In this section, we slightly modify the original FO transform and give a new transform which we call $\overline{\text{FO}}$. Next we instantiate the new transform $\overline{\text{FO}}$ using iO and
Figure 6.5: Modified FO transform $\mathcal{FO}_{H,G}[\text{PKE, SE}] = (\mathcal{FO}.\text{Dec, FO.Enc, FO.Dec})$.

ELF. Let $\text{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a private-key encryption, $\text{PKE} = (\mathcal{K}_g, \mathcal{E}_g, \mathcal{D}_g)$ be a public-key encryption schemes, and $\text{PRG}$ be a pseudorandom generator. Moreover, let $\mathcal{H} : \mathcal{K}_H \times \text{HDom} \rightarrow \text{HRng}$ and $\mathcal{G} : \mathcal{K}_G \times \text{GDom} \rightarrow \text{GRng}$ be function families. We define $\mathcal{FO}_{H,G}[\text{PKE, SE}] = (\mathcal{FO}.\text{Dec, FO.Enc, FO.Dec})$ in Figure 6.5.

Note that in practice we use pseudorandom generator to transforming a short, uniform string called the seed $r$ into a longer, pseudorandom output string which we usually use as randomness in the encryption schemes. Thus, the changes are conservative. Let $\text{PRF} = (\text{PRF}.\mathcal{K}_g, \text{PRF}.\mathcal{Punct}, \text{PRF}.\mathcal{Eval})$ be a puncturable PRF and $\text{iO}$ be an indistinguishability obfuscator for all circuits in $\mathcal{P}/\text{poly}$. Moreover, let $\text{ELF} = (\text{ELF}.\text{IK}_g, \text{ELF}.\text{LK}_g, \text{ELF}.\mathcal{Eval})$ be a family of extremely lossy functions. We define our hash function families $\mathcal{H} : \mathcal{K}_H \times \text{HDom} \rightarrow \text{HRng}$ and $\mathcal{G} : \mathcal{K}_G \times \text{GDom} \rightarrow \text{GRng}$ in Figure 6.6.

We show in Theorem 37 that $\mathcal{FO}_{H,G}[\text{PKE, SE}]$ is IND-CCA secure if $\mathcal{H}$ and $\mathcal{G}$ are instantiated as in Figure 6.6. Next, we explain the proof idea. Let circuit $C_1 = \text{PRF}_K(\text{PRG}())$. First we show that $c_1^* = \text{Enc}(\text{PRG}(r^*); C_1(r^*))$ is indistinguishable from $c_1^* = \text{Enc}(\text{PRG}(r^*); y^*)$ for freshly chosen random $y^*$ to any PPT adversary. To do so, we use a technique similar to the one given in [90]. At a very high-level, the idea of the technique is to alter a program $C_1$ (which is to be obfuscated) by
surgically removing a key element of the program (in a way that does not alter the functionality of the program), without which the adversary cannot distinguish between \( C_1(r^*) \) and freshly chosen random \( y^* \), given \( \text{PRG}(r^*) \). We first argue that since \( \text{PRG} \) is secure, the adversary cannot distinguish the original security game in which the challenge ciphertext was created as \( c_1^* = \text{Enc}(\text{PRG}(r^*); \text{PRF}_K(\text{PRG}(r^*))) \), and a hybrid experiment where the challenge ciphertext is created with a freshly chosen random \( x^* \) as \( c_1^* = \text{Enc}(x^*; \text{PRF}_K(x^*)) \). Note that the point \( x^* \) is not functionally accessible by the circuit with high probability (for significantly expanding \( \text{PRG} \) we have w.h.p that \( x^* \notin \text{PRGRng} \)). Thus we can puncture the PRF key \( K \) on \( x^* \) without effecting the functionality of circuit \( C_1 \).

Thus, indistinguishability obfuscation guarantees that an obfuscation of an alternative circuit that uses a punctured PRF key that carves out \( x^* \) is indistinguishable from the obfuscation of the original circuit, because these two circuits are functionally equivalent. Now, due to the puncturing, the adversary simply does not have enough information to distinguish \( c_1^* = \text{Enc}(x^*; \text{PRF}_K(x^*)) \) from \( c_1^* = \text{Enc}(x^*; y^*) \).

Next we need to show that \( c_2^* = \mathcal{E}_{K^*}(m) \) looks uniformly random to any PPT adversary given \( c_1^* \), where \( K^* = C_2(x^*) \) and circuit \( C_2 = \text{ELF}(\text{PRF}_K(\cdot)) \). To do so,
we adapt a new approach by incorporating ELF to the technique in [35] that was used to instantiate UCEs. We use AIPO to alter the circuit \( \mathcal{C}_2 \) on input \( x^* \) to output freshly random value \( K^* \) instead of \( \text{ELF}(\text{PRF}_K(x^*)) \). We show that obfuscation of an alternative circuit is indistinguishable from the obfuscation of the original circuit, using differing input obfuscation given the auxiliary information \( c_1 \) of the differing point \( x^* \). However, in order to do so adversary attacking diO need to simulate the decryption oracle for IND-CCA adversary. We do this by using the property of ELF. Considering the running time of the IND-CCA adversary, we switch to the proper extremely lossy mode of ELF function that is indistinguishable to the adversary. We note that once we are in lossy mode we can answer decryption queries by going over all possible output of the extremely lossy function. Now that once \( K^* \) is uniformly

---

**Figure 6.7: Games \( G_1-G_4 \) in the proof of Theorem 37.**
\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Games $G_5(k)$, $G_6(k)$ & Games $G_7(k)$, $G_8(k)$ \\
\hline
$K_1 \leftarrow \text{PRF.} \text{Kg}(1^k)$ ; $K_2 \leftarrow \text{PRF.} \text{Kg}(1^k)$ & $K_1 \leftarrow \text{PRF.} \text{Kg}(1^k)$ ; $K_2 \leftarrow \text{PRF.} \text{Kg}(1^k)$ \\
$K'_2 \leftarrow \text{PRF.} \text{Punct} (K_2, x^*)$ ; $p \leftarrow \text{AIPO}(x^*)$ & $K'_2 \leftarrow \text{PRF.} \text{Punct} (K_2, x^*)$ ; $p \leftarrow \text{AIPO}(x^*)$ \\
f $\leftarrow \text{ELF.} \text{IKg}(1^k)$ ; $K_H \leftarrow \text{iO}(C_1[K])$ & f $\leftarrow \text{ELF.} \text{IKg}(1^k)$ ; $K_H \leftarrow \text{iO}(C_1[K])$ \\
$K_G \leftarrow \text{iO}(C_2[K_2, f])$ ; $(p k', sk') \leftarrow \text{Kg}(1^k)$ & $(p k', sk') \leftarrow \text{Kg}(1^k)$ ; $p k \leftarrow (p k', K_H, K_G)$ \\
$K_G \leftarrow \text{iO}(C_3[K_2^*, f, p, K^*])$ & b $\leftarrow \{0, 1\}$ ; $(m_0, m_1, st) \leftarrow A_{1}^{\text{Dec}}(p k)$ \\
$\text{Circuit } C_3[K_2^*, f, p, K^*](x)$ & $x^* \leftarrow \text{PRG.} \text{Rng}(k)$ ; $y^* \leftarrow \text{HRng}(k)$ \\
If $p(x) = \perp$ then return $f(\text{PRF}_K^*(x))$ & $z^* \leftarrow \text{PRF.} \text{Rng}(k)$ ; $K^* \leftarrow f(z^*)$ \\
Return $K^*$ & $K^* \leftarrow \text{GRng}(k)$ \\
\hline
\end{tabular}
\caption{Games $G_5$–$G_8$ in the proof of Theorem 37.}
\end{table}

Random we conclude that $c_2$ looks uniformly random using IND-CCA security of symmetric encryption SE.

**Theorem 37** Assuming that PRG is a secure PRG with the range size of at least $2^k$ times greater than size of the domain, ELF is a secure ELF, PRF is a secure PRF, iO is a secure indistinguishability obfuscation, and assuming AIPO exists. Moreover, assuming PKE is OW-CPA secure and SE is IND-CPA and INT-CTXT secure and robust private-key encryption. Then if $\mathcal{H}$ and $G$ are instantiated as above, $\mathcal{FO}$ defined as above is IND-CCA secure.

**Proof.** Consider games $G_1$–$G_{12}$ in Figures 6.7, 6.8 and 6.9. We now explain the game chain.
Games $G_9(k), G_{10}(k)$

$K_1 \leftarrow \text{PRF}.Kg(1^k)$; $K_2 \leftarrow \text{PRF}.Kg(1^k)$
$p \leftarrow \text{AIO}(x^*)$; $f \leftarrow \text{ELF}.Lkg(1^k)$
$f \leftarrow \text{ELF}.Lkg(1^k)$; $K_H \leftarrow \text{iO}(C_1[K_1])$
$K_G \leftarrow \text{iO}(C_3[K_2,f,p,K^*])$
$(pk', sk') \leftarrow Kg(1^k)$; $pk \leftarrow (pk', K_H, K_G)$
$b \leftarrow \{0, 1\}$; $(m_0, m_1, st) \leftarrow A_1^{\text{Dec}()}(pk)$
$x^* \leftarrow \text{PRG}Rng(k)$; $y^* \leftarrow \text{HRng}(k)$
$z^* \leftarrow \text{PRF}.Rng(k)$; $K^* \leftarrow \text{GRng}(k)$
$c_1 \leftarrow Enc(pk', x^*; y^*)$; $c_2 \leftarrow \mathcal{E}_{K^*}^{\text{cy}}(m_b)$
$c' \leftarrow (c_1, c_2)$; $b' \leftarrow A_2^{\text{Dec}()}(st, c)$
Return $(b = b')$

Procedure $\overline{\text{Dec}}(c = (c_1, c_2))$ // of games $G_{11}, G_{12}$

For all $K \in [f(\cdot)]$ do
  $m \leftarrow D_K^{\text{cy}}(c_2)$
  If $m \neq \perp$ then return $m$
Return $\perp$

Games $G_{11}(k), G_{12}(k)$

$K_1 \leftarrow \text{PRF}.Kg(1^k)$; $K_2 \leftarrow \text{PRF}.Kg(1^k)$
$f \leftarrow \text{ELF}.Lkg(1^k)$; $K_H \leftarrow \text{iO}(C_1[K_1])$
$K_G \leftarrow \text{iO}(C_3[K_2, f, p, K^*])$
$(pk', sk') \leftarrow Kg(1^k)$; $pk \leftarrow (pk', K_H, K_G)$
$b \leftarrow \{0, 1\}$; $(m_0, m_1, st) \leftarrow A_1^{\text{Dec}()}(pk)$
$x^* \leftarrow \text{PRG}Rng(k)$; $y^* \leftarrow \text{HRng}(k)$
$z^* \leftarrow \text{PRF}.Rng(k)$; $K^* \leftarrow \text{GRng}(k)$
$c_1 \leftarrow Enc(pk', x^*; y^*)$; $c_2 \leftarrow \mathcal{E}_{K^*}^{\text{cy}}(m_b)$
$c' \leftarrow (c_1, c_2)$; $b' \leftarrow A_2^{\text{Dec}()}(st, c)$
Return $(b = b')$

Figure 6.9: Games $G_9$–$G_{12}$ in the proof of Theorem 37.

Game $G_1$: This is the standard indistinguishability chosen ciphertext game. Suppose adversary $A$ runs in time $s$ and wins game $G_1$ with probability $\epsilon$ where $\epsilon$ is non-negligible. Let $\delta$ be an inverse polynomial such that $\epsilon \geq \delta$ infinitely often.

Game $G_2$: Game $G_2$ is similar to game $G_1$ except that $x^*$ is chosen randomly in $\text{PRG}Rng(k)$ (i.e., we switch the output of $\text{PRG}$ to uniformly random value). Note that $r^*$ is no longer in the attacker’s view and does not need to be generated. Considering the adversary $D_1$ attacking $\text{PRG}$ in Figure 6.10, we get that $\Pr [G_1 \Rightarrow 1] - \Pr [G_2 \Rightarrow 1] \leq \text{Adv}_{\text{PRG}, D_1}^{\text{PRG}}(k)$. Thus, adversary $A$ wins in game $G_2$ with probability at least $\epsilon - \text{Adv}_{\text{PRG}, D_1}^{\text{PRG}}(k)$.

Game $G_3$: Game $G_3$ is similar to game $G_2$ except that the hash key $K_H$ is created as an obfuscation of the circuit $C_1[K^*]$, instead of $C_1[K_1]$. Note that with all but
negligible probability \( x^* \) is not in the image of the PRG. We first note that with all but negligible probability, the two circuits are functionally equivalent when \( x^* \) is chosen at random. The reason is that with probability \( 1 - 2^{-k} \), \( x^* \) is not in the image on the PRG, since size of the range is at least \( 2^k \) times larger than size of the domain. Hence, puncturing \( x^* \) from the key \( K_1 \) will not effect input/output behavior. Therefore, considering the adversary \( D_2 \) attacking iO in Figure 6.10, we get that \( \Pr \left[ G_2 \Rightarrow 1 \right] - \Pr \left[ G_3 \Rightarrow 1 \right] \leq \text{Adv}^{\text{iO}}_{\text{iO},D_2,C}(k) + 2^{-k} \). Thus, adversary \( A \) wins game \( G_3 \) with probability at least \( \epsilon - \text{Adv}^{\text{PRG}}_{\text{PRG},D_1}(k) - \text{Adv}^{\text{iO}}_{\text{iO},D_2,C}(k) - 2^{-k} \).
**Procedure Dec(c)**

\[ m \leftarrow \text{FO. Dec}(sk, c) \]

Return \( m \)

**Circuit \( C_1[K_1](r) \)**

Return \( PRF_{K_1}(\text{PRG}(r)) \)

**Circuit \( C_2[K_2, f](x) \)**

Return \( f(PRF_{K_2}(x)) \)

---

**Figure 6.11: Adversary \( D_3 \) in the proof of Theorem 37.**

---

**Game \( G_4 \):** Game \( G_4 \) is similar to game \( G_3 \) the exception that \( y^* \) is chosen randomly in \( HRng(k) \) (i.e., we switch the output of \( PRF \) to uniformly random value).

Considering the adversary \( D_3 \) attacking pseudorandom function \( PRF \) at the punctured points in Figure 6.11, we get that

\[ \Pr[G_3 \Rightarrow 1] - \Pr[G_4 \Rightarrow 1] \leq \text{Adv}_{\text{pprf}, D_3}^\text{pprf}(k). \]

**Game \( G_5 \):** Game \( G_5 \) is similar to game \( G_3 \) except that the hash key \( K_H \) is created as an obfuscation of the circuit \( C_1[K_1] \), instead of \( C_1[K_1^*] \). Using the same argument in game \( G_3 \) we obtain that

\[ \Pr[G_4 \Rightarrow 1] - \Pr[G_5 \Rightarrow 1] \leq \text{Adv}_{\text{iO}, D_3, C}^\text{iO}(k). \]

**Game \( G_6 \):** Game \( G_6 \) is similar to game \( G_5 \) except that we puncture the PRF key \( K_2 \) at \( x^* \). Moreover, the hash key \( K_G \) does not consist of an obfuscation of \( C_2[K_2, f] \), but rather of an obfuscation of the circuit \( C_3[K_2^*, f, p, K^*] \). Here, \( p \) is the AIPO obfuscation of the point function \( p_{x^*} \) and thus, \( p(x) \) outputs 1 if and only if \( x \) is equal to \( x^* \). Note that the two circuits are functionally equivalent. Therefore, considering the adversary \( D_3 \) in Figure 6.12, we get that

\[ \Pr[G_5 \Rightarrow 1] - \Pr[G_6 \Rightarrow 1] \leq \text{Adv}_{\text{iO}, D_3, C}^\text{iO}(k). \]
Considering the adversary punctured points, we get that identical to circuit Fig. 6.13, we get that.

\[
\text{Algorithm 6.12: Adversary } D_5 \text{ in the proof of Theorem 37.}
\]

**Game G\textsubscript{7}:** Game G\textsubscript{7} is similar to game G\textsubscript{6} except that \( z^* \) is chosen randomly in \( \text{PRF.Rng}(k) \) \( i.e., \) we switch the output of \( \text{PRF} \) to uniformly random value. Considering the adversary \( D_6 \) attacking pseudorandom function \( \text{PRF} \) at the punctured points, we get that \( \Pr [G_6 \Rightarrow 1] - \Pr [G_7 \Rightarrow 1] \leq \text{Adv}^{\text{prf}}_{\text{PRF}, D_6}(k) \). We skip the code of adversary \( D_6 \) and note that it is very similar to the code of adversary \( D_3 \).

**Game G\textsubscript{8}:** Game G\textsubscript{8} is similar to game G\textsubscript{7} except that \( K^* \) is chosen randomly in \( \text{GRng}(k) \) \( i.e., \) we switch the output of \( \text{ELF} \) to uniformly random value. Considering the adversary \( D_7 \) distinguishing the output of \( \text{ELF} \) from uniformly random in Fig. 6.13, we get that \( \Pr [G_7 \Rightarrow 1] - \Pr [G_8 \Rightarrow 1] \leq \text{Adv}^{\text{elf}}_{\text{ELF}, D_7}(k) \).

**Game G\textsubscript{9}:** Game G\textsubscript{9} is similar to game G\textsubscript{8} except that an obfuscation of circuit \( C_3[K_2, f, p, K^*] \) is used as the hash key \( K_G \). Note that circuit \( C_3[K_2^*, f, p, K^*] \) is identical to circuit \( C_3[K_2, f, p, K^*] \), except that it uses the original PRF key \( K_2 \).
Procedure Dec(c)

\[ m \leftarrow \text{FO. Dec}(sk, c) \]
Return \( m \)

Circuit \( C_1[K_1](r) \)
Return \( \text{PRF}_{K_1}(\text{PRG}(r)) \)

Circuit \( C_3[K_2^*, f, p, K^*](x) \)
If \( p(x) = \perp \)
Return \( f(\text{PRF}_{K_2^*}(x)) \)
Return \( K^* \)

<table>
<thead>
<tr>
<th>Adversary ( D_7(K^*, f) )</th>
<th>Procedure Dec(c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x^* \leftarrow \text{PRG}^r(k) ); ( y^* \leftarrow \text{HRng}(k) )</td>
<td>( m \leftarrow \text{FO. Dec}(sk, c) )</td>
</tr>
<tr>
<td>( K_1 \leftarrow \text{PRF. Kg}(1^k) ); ( K_2 \leftarrow \text{PRF. Kg}(1^k) )</td>
<td>Return ( m )</td>
</tr>
<tr>
<td>( K_2^* \leftarrow \text{PRF. Punct}(K_2, x^<em>) ); ( p \leftarrow \text{AIPO}(x^</em>) )</td>
<td>Circuit ( C_1<a href="r">K_1</a> )</td>
</tr>
<tr>
<td>( (pk', sk') \leftarrow \text{Kg}(1^k) ); ( K_H \leftarrow \text{iO}(C_1[K_1]) )</td>
<td>Return ( \text{PRF}_{K_1}(\text{PRG}(r)) )</td>
</tr>
<tr>
<td>( K_G \leftarrow \text{iO}(C_3[K_2^<em>, f, p, K^</em>]) )</td>
<td>Circuit ( C_3<a href="x">K_2^<em>, f, p, K^</em></a> )</td>
</tr>
<tr>
<td>( pk \leftarrow (pk', K_H, K_G) ); ( sk \leftarrow (pk, K_H, K_G) )</td>
<td>If ( p(x) = \perp )</td>
</tr>
<tr>
<td>( b \leftarrow {0, 1} ); ( (m_0, m_1, st) \leftarrow A_{1}^\text{Dec}(pk) )</td>
<td>Return ( f(\text{PRF}_{K_2^*}(x)) )</td>
</tr>
<tr>
<td>( c_1 \leftarrow \text{Enc}(pk', x^<em>; y^</em>) ); ( c_2 \leftarrow e_{K^*}^{g}(m_b) )</td>
<td>Return ( K^* )</td>
</tr>
<tr>
<td>( c \leftarrow (c_1, c_2) ); ( b' \leftarrow A_{2}^\text{Dec}(st, c) )</td>
<td></td>
</tr>
<tr>
<td>Return ( (b = b') )</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 6.13:** Adversary \( D_7 \) in the proof of Theorem 37.

instead of the punctured key \( K_2^* \). The two circuits are functionally equivalent. Therefore, considering the adversary \( D_8 \) attacking iO, we get that \( \Pr[G_8 \Rightarrow 1] - \Pr[G_9 \Rightarrow 1] \leq \text{Adv}^\text{iO}_{D_8}(k) \). We note that adversary \( D_8 \) is very similar to adversary \( D_2 \).

**Game \( G_{10} \):** Game \( G_{10} \) is similar to game \( G_9 \) except that we change ELF to be lossy mode. That is, we generate \( f \leftarrow \text{ELF. LKg}(1^k, p(s, 2/\delta)) \). Considering the adversary \( D_9 \) attacking lossiness of ELF running in time \( s \), we get that \( \Pr[G_9 \Rightarrow 1] - \Pr[G_{10} \Rightarrow 1] \leq \delta/2 \).

**Game \( G_{11} \):** Game \( G_{11} \) is similar to game \( G_{10} \) except that in the decryption oracle we examine all possible values of symmetric key \( K \) in the range of the ELF function \( f \) to decrypt the adversary queries. Let \( \tau_i = (\tau_{i,1}, \tau_{i,2}) \) be the \( i \)-th decryption query that adversary \( A \) makes. Moreover, let \( \overline{K}_i \) be the symmetric key that was used in the encryption process of \( \overline{\tau}_i \). Note that, either \( \overline{K}_i = K^* \) or \( \overline{K}_i \in [f(\cdot)] \).

Consider the adversary \( B_i \) in Figure 6.14 attacking integrity of symmetric
Adversary $B_i^{\ell_{K^*}(\cdot)}(1^k)$

$K_1 \leftarrow$ PRF.$\text{Kg}(1^k)$; $j \leftarrow 1$

$f \leftarrow \text{ELF.}\text{Lkg}(1^k)$; $K_2 \leftarrow$ PRF.$\text{Kg}(1^k)$

$K_H \leftarrow \text{iO}(C_1[K_1])$; $K_G \leftarrow \text{iO}(C_2[K_2,f])$

$(pk',sk') \leftarrow \text{Kg}(1^k)$; $pk \leftarrow (pk',K_H,K_G)$

$b \leftarrow \{0,1\}$; $(m_0,m_1,st) \leftarrow A^{\text{Dec}(\cdot)}_1(pk)$

$x* \leftarrow \text{PRG}^{\text{rng}}(k)$; $y* \leftarrow \text{HRng}(k)$

$c_1 \leftarrow \text{Enc}(pk',x^*;y^*)$; $c_2 \leftarrow \mathcal{E}_{K^*}(m_b)$

$c \leftarrow (c_1,c_2)$; Run $A^{\text{Dec}(\cdot)}_2(st,c)$

Return $\overline{c}$

Procedure $\text{Dec}(c = (c_1,c_2))$

If $i = j$ then $c_2 \leftarrow c_2$

$j \leftarrow j + 1$

For all $K \in [f(\cdot)]$ do

$m \leftarrow D^y_K(c_2)$

If $m \neq \bot$ then return $m$

Return $\bot$

Circuit $C_1[K_1](r)$

Return PRF$_{K_1}(\text{PRG}(r))$

Circuit $C_2[K_2,f](x)$

Return $f(\text{PRF}_{K_2}(x))$

| Figure 6.14: Adversary $B_i$ in the proof of Theorem 37. |

encryption SE, and the fact that $(\text{iO}(C_3[K_2,f,p,K^*],z)$ and $(\text{iO}(C_2[K_2,f]),z)$ are indistinguishable (shown in game $G_{12}$) where $z = (c_1,pk',K_H,f,K^*)$, we obtain that $\Pr[\overline{K}_i = K^*] \leq \text{Adv}_{\text{SE},B_i}^{\text{int ctxt}}(k) + \text{Adv}_{\text{iO},D_{11},\text{Samp}}^{\text{dio}}(k)$. On the other hand, we know that the image size of $f$ is polynomial. Moreover, we know that private-key encryption SE is robust, thus with probability at least $1 - \text{Adv}_{\text{SE},B}(k)$, there is a unique $\overline{K}_i \in [f(\cdot)]$ that successfully decrypts $\mathcal{C}$. Therefore, we can try all possible $\overline{K}_i$ to correctly decrypt the decryption queries. Hence, we obtain that $\Pr[G_{10} \Rightarrow 1] \leq \Pr[G_{11} \Rightarrow 1] + q \cdot \text{Adv}_{\text{SE},B_i}^{\text{int ctxt}}(k) + q \cdot \text{Adv}_{\text{iO},D_{11},\text{Samp}}^{\text{dio}}(k) + \text{Adv}_{\text{SE},B}(k)$, where $q$ is the number of decryption queries that adversary $A$ makes.

Game $G_{12}$: Game $G_{12}$ is similar to game $G_{11}$ except that the hash key $K_G$ is consistent of an obfuscation of circuit $C_2[K_2,f]$. Note that Circuit $C_2[K_2,f]$ is the original circuit. Observe that the circuits $C_3[K_2,f,p,K^*]$ and $C_2[K_2,f]$ only differ on points where $p(x)$ is not equal to $\bot$, that is, they differ on a single
point, which is the point $x^*$. We will bound the difference between games $G_{11}$ and $G_{12}$ by the differing-inputs security of the indistinguishability obfuscator iO. We note that Boyle et al. in [32] show that any indistinguishability obfuscator is also a differing-inputs obfuscator for differing-inputs circuits which differ on at most polynomial many points. Now, consider the differing-inputs circuit sampler $\text{Samp}$ in Figure 6.15. We show in Lemma 38 that family of circuit pairs $(C_3[K_2, f, p, K^*], C_2[K_2, f], \text{Samp})$ is differing-inputs. Considering the adversary $D_{11}$ attacking differing-inputs obfuscator iO in Figure 6.15, we get that $\Pr[G_{11} \Rightarrow 1] - \Pr[G_{12} \Rightarrow 1] \leq \text{Adv}_{iO,D_{11},\text{Samp}}^{\text{dio}}(k)$.

Observe that adversary $A$ running in time $s$ wins in game $G_{12}$ with probability at least $\delta/2 - \text{neg}(k)$. Note that this quantity is at least $\delta/3$ infinitely often, and is therefore non-negligible. However, considering adversary $D_{12}$ attacking the private-

---

**Figure 6.15**: Adversary $D_{11}$ and circuit sample $\text{Samp}$ in the proof of Theorem 37.
key encryption \(SE\), we obtain that \(\Pr[G_{12} \Rightarrow 1] \leq \text{Adv}^{\text{ind-cpa}}_{SE,D_{12}}(k)\). Therefore, we get that \(\delta/3 \leq \text{Adv}^{\text{ind-cpa}}_{SE,D_{12}}(k)\) which contradicts. Hence, there are no PPT adversaries that can win game \(G_1\) with non-negligible probability. This completes the proof of Theorem 37.

**Lemma 38** If \(\text{AIPO}\) is a secure AIPO obfuscator then the family of circuit pairs \((C_3[K_2, f, p, K^*], C_2[K_2, f], \text{Samp})\) is differing-inputs.

**Proof.** Let \(B\) be an adversary against the differing-inputs of the above circuit family which receives as input \((C_3[K_2, f, p, K^*], C_2[K_2, f], z)\) and outputs a value \(\alpha\) such that \(C_3[K_2, f, p, K^*](\alpha) = C_2[K_2, f](\alpha)\), where \(z = (c_1, pk', K_H, f, K^*)\). Then, we show that we can build an adversary against \(\text{AIPO}\) using adversary \(B\). Consider distribution \(D\) and adversary \(A\) attacking \(\text{AIPO}\) in Figure 6.16.

We start by showing that \(D\) is an unpredictable distribution. Note that if there exist an adversary that outputs \(x^*\) on input \((c_1, pk', K_H, t, b)\), we can build a OW-CPA adversary against public key encryption \(PKE\). Thus, distribution \(D\) is unpredictable.
Next, we note that probability of adversary $B$ succeed against differing-inputs distribution $\text{Samp}$ is bounded by $\text{Adv}^{\text{aiPo}, A, D}(k)$. We skip the details and note that it is similar to the proof of Claim 3.4 of [35]. This completes the proof from Lemma 38.
Chapter 7

Selective-Opening Security of Deterministic Primitives

In this chapter, we revisit the problem of selective-opening security for deterministic (public-key) encryption, whose study was recently initiated by Bellare, Dowsley, and Keelveedhi (BDK) at PKC 2015. While BDK showed that a “simulation-based” semantic-security style definition of selective-opening security is unachievable for deterministic encryption, we propose a new “comparison-based” semantic-security style definition which we call D-SO-CPA and show how to realize it efficiently. Note that if the adversary can open $d$ messages, for any meaningful privacy we need that the distribution of messages is what we call $(\mu, d)$-entropic for sufficiently large $\mu$, meaning that every message has entropy at least $\mu$ conditioned on fixed values of any $d$ others. We will refer to these parameters below. We stress that our definition and constructions allow $d$ to be $\infty$, that is, on proper message distributions, the adversary can open as many messages as it wants. For $d = 0$, our notion degenerates to the standard definition PRIV [10] of deterministic encryption. On the other hand, the standard security notion PRIV for D-PKE doesn’t imply D-SO-CPA security.

We can construct a contrived D-PKE scheme that is PRIV-secure, but vulnerable to an efficient SOA attack. Our construction relies on the recent result of Hofheinz, Rao, and Wichs [69] that separates the standard IND-CPA notion and the SOA security (IND-SO-CPA) of R-PKE. In our attack, the message sampler is $(3\ell, t)$-entropic. It first picks a string $s \leftarrow \{0, 1\}^\ell$ and then secret-shares $s$ so that any $t$ shares reveal no information about the secret. It then outputs a vector of $3\ell$-bit
messages, in which the $\ell$-bit prefix of each $i$-th message is the $i$-th share, and the $2\ell$-bit suffix is a fresh random string. However, it’s still open whether PRIV and D-SO-CPA are equivalent on independent and uniform messages. It is interesting to note that D-SO-CPA seems inequivalent to standard security notions for D-PKE even in the case of independent and uniform messages. (Very roughly, the reason has to do with the fact that security of D-PKE considers “split” adversaries that do not share state.) This means that we will need dedicated constructions and analyses regardless of the assumptions we make on the messages.

We first give constructions in the standard model for a bounded number of messages. Our constructions use lossy trapdoor functions and lossy encryption, and admit efficient instantiations under standard number-theoretic assumptions. We then extend our results for an unbounded number of “$t$-correlated” messages, meaning each set of up to $t$ messages may be arbitrarily correlated. We consider the notion of $t$-correlated messages to be interesting in its own right, and it captures a setting with password hashing where a password is correlated with a small number of others (and it is even stronger than that, in that a password may be correlated with any small number of others). Our construction uses $2t$-wise independent hash functions and regular lossy trapdoor function [84], which has practical instantiations, e.g., RSA is regular lossy [76]. A close variant of our scheme is shown to be D-SO-CPA secure in the NPROM [65].

7.1 Selective-Opening Security Definition for Deterministic Public-Key Encryption

Bellare, Dowsley, and Keelveedhi [17] were the first to consider selective-opening security of deterministic PKE (D-PKE). They propose a “simulation-based” semantic
security notion, but then show that this definition is unachievable in both the standard model and the non-programmable random-oracle model (NPROM), even if the messages are independent and uniformly random. To address this, we introduce an alternative, “comparison-based” semantic-security notion that generalizes the original PRIV definition for D-PKE of Bellare, Boldyreva, and O’Neill [10]. In particular, our notion follows the IND-SO-CPA notion of Bellare, Hofheinz, and Yilek (BHY) [13] in the sense that we compare what partial information the adversary learns from the unopened messages, versus messages resampled from the same conditional distribution. Following BHY, we require that the message space be efficiently resamplable, which we define first.

MESSAGE SAMPLERS. A message sampler \( M \) is a PT algorithm that takes as input the unary representation \( 1^k \) of the security parameter and a string \( \text{param} \in \{0,1\}^* \), and outputs a vector \( m \) of messages. We require that \( M \) be associated with functions \( v(\cdot) \) and \( n(\cdot) \) such that for any \( \text{param} \in \{0,1\}^* \), for any \( k \in \mathbb{N} \), and any \( m \in [M(1^k,\text{param})] \), we have \( |m| = v(k) \) and \( |m[i]| = n(k) \), for every \( i \leq |m| \).

Moreover, the components of \( m \) must be distinct. Let \( \text{Coins}[k] \) be the set of coins for \( M(1^k,\cdot) \). Define \( \text{Coins}[k, m, I, \text{param}] = \{ \omega \in \text{Coins}[k] \mid m[I] = m'[I], \text{ where } m' \leftarrow M(1^k,\text{param}; \omega) \} \).

A message sampler \( M \) is \((\mu,d)\)-entropic if for any \( k \in \mathbb{N} \), any \( I \subseteq [v(k)] \) such that \( |I| \leq d \), any \( \text{param} \in \{0,1\}^* \), and any \( m \in \{0,1\}^* \), it holds that \( \Pr\left[ m'[i] = m : m \leftarrow M(1^k, \text{param}) ; \omega \leftarrow \text{Coins}[k, m, I, \text{param}] ; m' \leftarrow M(1^k, \text{param}; \omega) \right] \leq 2^{-\mu(k)} \) for all \( i \in [v(k)] \setminus I \). Note that in this definition, \( d \) can be \( \infty \), which corresponds to a message sampler in which the conditional distribution of each message, given all other messages, has at least \( \mu \) bits of min-entropy.

A message sampler \( M \) is \((\mu,d)\)-correlated if for any \( k \in \mathbb{N} \), any \( \text{param} \in \{0,1\}^* \), every \( m \in [M(1^k, \text{param})] \) and any \( i \in [v] \), \( m[i] \) have min-entropy at least \( \mu \) and is
independent of at least \( v - d \) messages. Note that in this definition, \( d \) can be 0, which corresponds to a message sampler in which each message is independent of all other messages and has at least \( \mu \) bits of min-entropy.

**Resampling.** Following [13], let \( \text{Resamp}_M(1^k, I, x, \text{param}) \) be the algorithm that samples \( r \leftarrow \text{Coins}[k, m, I, \text{param}] \) and returns \( M(1^k, \text{param}; r) \). (Note that \( \text{Resamp} \) may run in exponential time.) A resampling algorithm of \( M \) is an algorithm \( \text{Rmsp} \) such that \( \text{Rmsp}(1^k, I, x, \text{param}) \) is identically distributed\(^1\) as \( \text{Resamp}_M(1^k, I, x, \text{param}) \). A message sampler \( M \) is efficiently resamplable if it admits a PT resampling algorithm.

**D-SO-CPA security.** Let \( \text{PKE} = (\text{Kg}, \text{Enc}, \text{Dec}) \) be a D-PKE scheme. To a message sampler \( M \) and an adversary \( A = (A.\text{pg}, A.\text{cor}, A.g, A.f) \), we associate the experiment in Figure 7.1 for every \( k \in \mathbb{N} \). We say that \( \text{DE} \) is D-SO-CPA secure for a class \( \mathcal{M} \) of efficiently resamplable message samplers and a class \( \mathcal{A} \) of adversaries if for every \( M \in \mathcal{M} \) and any \( A \in \mathcal{A} \),

\[
\text{Adv}_{\text{DE}, A, \mathcal{M}}^{d\text{-so-cpa}}(\cdot) = \Pr[D\text{-CPA1-REAL}_{\text{DE}}^{A, \mathcal{M}}(\cdot) \Rightarrow 1] - \Pr[D\text{-CPA1-IDEAL}_{\text{DE}}^{A, \mathcal{M}}(\cdot) \Rightarrow 1]
\]

is negligible.

We refer to the messages indexed by \( I \) as the “opened” messages. Note that if the adversary always specifies \( I = \emptyset \) (meaning it opens no messages) then the D-SO-CPA notion collapses to the PRIV notion of Bellare et al. [10].\(^2\)

---

\(^1\)Here for simplicity, we only consider \( M \) and \( \text{Rmsp} \) such that the distributions of \( \text{Rmsp}(1^k, I, x, \text{param}) \) and \( \text{Resamp}_M(1^k, I, x, \text{param}) \) are identical. Following [13], one might also consider \( M \) and \( \text{Rmsp} \) such that the two distributions above are statistically close.

\(^2\)A minor technical difference is that that here, to be consistent with [17], we require the “partial information” be an efficiently computable function of the messages. We note that this formulation can be shown equivalent to a definition in the style of [10] up to a difference of one in the size of the message vectors output by \( M \), following [11, Appendix A].
When considering selective-opening attacks against a D-PKE scheme in which an adversary can open $d$ messages, it is clear that message sampler must be $(\mu,d)$-entropic where $2^{-\mu(\cdot)}$ is a negligible function for any meaningful privacy to be achievable. To justify this, consider $M$ for which there exists $I^* \subseteq [v(k)]$ such that $|I^*| \leq d$ and for every $\text{param}^* \in \{0,1\}^*$, $m^* \in \{0,1\}^*$, and $i^* \in [v(k)] \setminus I^*$, it holds that $\Pr[m'[i^*] = m^* : m \leftarrow M(1^k, \text{param}^*) ; \omega \leftarrow \text{Coins}[k,m, I^*, \text{param}^*] ; m' \leftarrow M(1^k, \text{param}^* ; \omega)]$ is at least $1/p(k)$ for a polynomial $p(\cdot)$. Define $A = (A,pg,A,cor,A,g,A,f)$ as follows. First, algorithm $A,pg(1^k)$ outputs $\text{param}^*$ and $A,cor(pk,c,\text{param}^*)$ returns $((pk,c),I^*)$. Next, $A,g((pk,c),m_1[I^*],\text{param}^*)$ runs $m_0 \leftarrow \text{Resamp}_M(1^k, m_1[I^*], I^*, \text{param}^*)$ until $\text{Enc}(pk,m_0[i^*]) = c[i^*]$ and outputs $m_0[i^*]$. Finally algorithm $A,f(m^*, \text{param}^*)$ outputs $m^*[i^*]$. Then $A$ runs in expected polynomial time$^3$ and $\text{Adv}^\text{d-socpa}_{DE,A,M}(\cdot) \geq 1 - 1/2$.

D-SO-CCA SECURITY. To add a CCA flavor to D-SO-CPA, a notion which we call D-SO-CCA, one would allow adversaries $A,cor$ and $A,g$ oracle access to $\text{Dec}(sk, \cdot)$. They are forbidden from querying a ciphertext in the given $c$ to this oracle. Let

$^3$It is easy to modify the adversary to run in strict polynomial time and adapt the argument.
D-CCA-REAL and D-CCA-IDEAL be the corresponding experiments, and define

\[
\text{Adv}_{\text{DE}, A, M}^{d-so-cca}(\cdot) = \Pr\left[D-\text{CCA-REAL}_{DE}^A, M(\cdot) \Rightarrow 1\right] - \Pr\left[D-\text{CCA-IDEAL}_{DE}^A, M(\cdot) \Rightarrow 1\right].
\]

We show that it is suffices to consider balanced D-SO-CPA adversaries where output of \( A.f \) is boolean. We call \( A \) \( \delta \)-balanced boolean D-SO-CPA adversary if for all \( b \in \{0, 1\} \),

\[
\left| \Pr \left[ t = b : t \leftarrow A.f(m, \text{param}) \right] - \frac{1}{2} \right| \leq \delta,
\]

for all \( \text{param} \) and \( m \) output by \( A.pg \) and \( M \), respectively.

**Theorem 39** Let \( PKE = (Kg, Enc, Dec) \) be a D-PKE scheme. Let \( A \) be a D-SO-CPA adversary against \( PKE \) with respect to message sampler \( M \). Then for any \( 0 \leq \delta < 1/2 \), there is a \( \delta \)-balanced boolean D-SO-CPA adversary \( B \) such that for all \( k \in \mathbb{N} \)

\[
\text{Adv}_{\text{DE}, A, M}^{d-so-cca}(k) \leq \left( \frac{2\sqrt{2}}{\delta} + \sqrt{2} \right)^2 \cdot \text{Adv}_{\text{DE}, B, M}^{d-so-cca}(k).
\]

The running time of \( A \) is about that of \( B \) plus \( \mathcal{O}(1/\delta) \).

We refer to [96, Theorem 3.1] for the proof of Theorem 39.

### 7.2 Scheme in the Standard Model (I)

In this section, we show that a specific instantiation of the “Encrypt-with-Hardcore” deterministic encryption scheme of Fuller et al. [58] achieves D-SO-CPA in the standard model. We then show a novel extension to this scheme that achieves D-SO-CCA in the standard model as well. We note our results here only apply to the bounded number of messages (i.e., the number of messages depends on the size of the public key). We start by defining the building blocks that we used in our constructions.
7.2.1 Building Blocks

Lossy encryption. A randomized PKE with message space $\text{Msg}(\cdot)$ is lossy if it can be written as a tuple of algorithms $\text{LE} = (\text{LE.IKg, LE.LKg, LE.Enc, LE.Dec})$ and a “lossy” ciphertext-space $\text{LCtxt}(\cdot)$ with the following requirements. Algorithm $\text{LE.IKg}$ on input a unary encoding of the security parameter $1^k$ outputs an “injective” public key $pk$ and matching secret key $sk$. Algorithm $\text{LE.LKg}$ on input a unary encoding of the security parameter $1^k$ outputs a “lossy” public key $pk'$. Algorithm $\text{LE.Enc}$ on input an (either injective or lossy) public key $pk$ and a message $m \in \text{Msg}(k)$ outputs a ciphertext $c$. Algorithm $\text{LE.Dec}$ on input a secret key $sk$ and a ciphertext $c$ outputs a message $m'$. We require the following properties:

**Correctness:** For all $k \in \mathbb{N}$ and all $m \in \text{Msg}(k)$

$$\Pr[\text{LE.Dec}(sk, c) \Rightarrow m : (pk, sk) \leftarrow \text{LE.IKg}(1^k) ; c \leftarrow \text{LE.Enc}(pk, m)] = 1.$$

**Indistinguishability of real and lossy keys:** For every PT adversary $D$,

$$\text{Adv}^k_{\text{LE}, D}(k) = \Pr[D(pk) \Rightarrow 1 : (pk, sk) \leftarrow \text{LE.IKg}(1^k)] - \Pr[D(pk') \Rightarrow 1 : pk' \leftarrow \text{LE.LKg}(1^k)]$$

is negligible.

**Perfect lossiness:** For any $k \in \mathbb{N}$ and any $m \in \text{Msg}(k)$

$$\Delta((pk, \text{LE.Enc}(pk', m)), (pk', U)) = 0,$$

where $pk' \leftarrow \text{LE.LKg}(1^k)$ and $U$ is uniform and independent on $\text{LCtxt}(k)$.

The last condition strengthens that of [13], which only requires that the encryptions under $pk'$ of any two messages are statistically close. Most known realizations (e.g., the ones based on DDH or Paillier’s DCR given in [13]) meet our strengthening.

Hash functions. A hash function with domain $\text{HDom}(\cdot)$ and range $\text{HRng}(\cdot)$ is a pair of algorithms $H = (\text{HKg}, h)$ that work as follows. Algorithm $\text{HKg}$ on input
$1^k$ outputs a key $K$. Algorithm $h$ on inputs a key $K$ and $x \in \text{HDom}(1^k)$ outputs $y \in \text{HRng}(1^k)$. We say that $H$ is $t(\cdot)$-wise independent if for all $k \in \mathbb{N}$ and all distinct $x_1, \ldots, x_{t(k)} \in \text{HDom}(1^k)$

$$\Delta((h(K, x_1), \ldots, h(K, x_{t(k)})), (U_1, \ldots, U_{t(k)})) = 0$$

where $K \leftarrow \text{HKg}(1^k)$ and $U_1, \ldots, U_{t(k)}$ are uniform and independent on $\text{HRng}(k)$.

**Target collision-resistant.** A hash function with domain $\text{TCRDom}(\cdot)$ and range $\text{TCRRng}(\cdot)$ is a pair of algorithms $\text{TCR} = (\text{TCRKg}, \text{tcr})$ that work as follows. Algorithm $\text{TCRKg}$ on input $1^k$ outputs a key $K$. Algorithm $\text{tcr}$ on inputs a key $K$ and $x \in \text{TCRDom}(1^k)$ outputs $y \in \text{TCRRng}(1^k)$. We say that $\text{TCR}$ is target collision-resistant, if $\text{Adv}_{\text{tcr}, \text{TCR}}(k) = \Pr[(m, \text{state}) \leftarrow A_1(1^k) ; K \leftarrow \text{TCRKg}(1^k) ; m' \leftarrow A_2(K, \text{state}) : m \neq m' \land \text{tcr}(K, m) = \text{tcr}(K, m')]$ is negligible for all $k \in \mathbb{N}$ and any adversary $A = (A_1, A_2)$.

### 7.2.2 The D-SO-CPA Scheme

Let $\text{LT} = (\text{IKg}, \text{LKg}, \text{Eval}, \text{Inv})$ be a lossy trapdoor function with domain $\text{LDom}(\cdot) = \{0, 1\}^{\text{LT}.\text{il}(\cdot)}$, range $\text{LRng}(\cdot)$, and lossiness $\tau$. Let $\text{LE} = (\text{LE}.\text{IKg}, \text{LE}.\text{LKg}, \text{LE}.\text{Enc}, \text{LE}.\text{Dec})$ be a perfectly-lossy encryption scheme with message-space $\text{LRng}(\cdot)$ and coin-space $\text{Coins}(\cdot) = \{0, 1\}^{\text{LE}.\text{rl}(\cdot)}$. Let $H = (\text{HKg}, h)$ be a $2t(\cdot)$-wise independent hash function with domain $\text{LDom}(\cdot)$ and range $\text{Coins}(\cdot)$. Define the associated deterministic encryption scheme $\text{DE}[\text{LE}, \text{LT}, H] = (\text{DE}.\text{Kg}, \text{DE}.\text{Enc}, \text{DE}.\text{Dec})$ with message-space $\text{LDom}(\cdot)$ in Figure 7.2.

The following theorems, which are proven in Appendix B.1 and Appendix B.2 respectively, characterize its security. We discuss them after the theorem statements.

**Theorem 40** Let $\text{DE}[\text{LE}, \text{LT}, H]$ be the D-PKE scheme defined above. Let $\mathcal{M}$ be a $(\mu, d)$-entropic, efficiently resamplable message sampler and let $v$ be the number of
messages that $\mathcal{M}$ produces. Assume $H = (HKg, h)$ is a family of $2(v - d)$-wise independent hash function. Then for any adversary $A$ opening at most $d$ ciphertexts,

$$\text{Adv}_{\text{DE}[\text{LE}, \text{LT}, H], A, M}(\cdot) \leq 2 \cdot \text{Adv}_{\text{LE}, B}(\cdot) + 2 \cdot \text{Adv}_{\text{LT}, D}(\cdot) + \frac{(2v - d)}{2r},$$

where $r = \frac{1}{3}(\mu - (v - d)(\text{LT}.\text{il} + \text{LE}.\text{rl} - \tau)) - u - 1$ and $u = \min\{1 + d\log v, v\}$. The running time of each of $B$ and $D$ is about the same as that of $A$ plus the running time of an efficient resampling algorithm of $\mathcal{M}$ plus the time to use $\text{DE}$ to encrypt $A$’s messages.

**Theorem 41** Let $\text{DE}[\text{LE}, \text{LT}, H]$ be as above. Let $\mathcal{M}$ be a $(\mu, \infty)$-entropic, efficiently resamplable message sampler, and let $v$ be the number of messages that $\mathcal{M}$ produces. Assume $H = (HKg, h)$ is a family of pair-wise independent hash function. Then for any adversary $A$, there are adversaries $B$ and $D$ such that

$$\text{Adv}_{\text{DE}[\text{LE}, \text{LT}, H], A, \mathcal{M}}(\cdot) \leq 2 \cdot \text{Adv}_{\text{LE}, B}(\cdot) + 2 \cdot \text{Adv}_{\text{LT}, D}(\cdot) + \frac{v}{2r},$$

where $r = \frac{1}{2}(\mu + \tau - \text{LT}.\text{il} - \text{LE}.\text{rl}) - v$. The running time of each of $B$ and $D$ is about the same as that of $A$ plus the running time of an efficient resampling algorithm of $\mathcal{M}$ plus the time to use $\text{DE}$ to encrypt $A$’s messages.
DISCUSSION. The first theorem above covers the more general case of $(\mu, d)$-entropic message samplers, whereas the second is specialized for the case of $(\mu, \infty)$-entropic message samplers.

In the first theorem, the term $(v - d)(LT.il + LE.rl - \tau)$ in the bound on $r$ is what one would expect to have based on the Crooked Leftover Hash Lemma[58, Lemma 3.10]. The dominant factor $u$ in the additional term is intuitively, due to the fact that we “guess” the subset of ciphertexts $I$ that the adversary chooses to open. Interestingly, although this means that in some sense we do not “solve” the problem of selective-opening, we pay this cost only in the information-theoretic part of the proof and still rely only on standard computational hardness assumptions. Also note that when $d$ is large, the public-key size in our scheme is small relative to the number of challenge ciphertexts. It is somewhat counter-intuitive that allowing the adversary to open more ciphertexts permits a smaller public-key size. The reason is that the message sampler is then correspondingly more restrictive.

In the second theorem, the bound we get on $r$ is slightly better than what is obtained by in first theorem in the case of $(\mu, \infty)$-entropic message samplers. This has to do with not paying a cost of a “worst-case” to “average-case” translation in the case of the Generalized Leftover Hash Lemma, cf. [50, Lemma 2.4]. Also note that in this case we only need pairwise-independent hashing and therefore the size of the public-key in our scheme does not grow with the number of challenge ciphertexts. Unfortunately, the bound on $r$ still implies that we can only handle a bounded number of messages in this case, whereas it is known how to handle an unbounded number without SOA [27].

INSTANTIATIONS. It is important in our scheme to use an underlying LE scheme whose coin-length is short relative to the size of the messages. Such a Paillier-based
LE scheme was given by [13], and we detail a sample Paillier-based instantiations of our scheme. Specifically, we use the Paillier-based LT from [27, 54] combined with the Paillier-based LE scheme from [13]. Consider the case of \((\mu, \infty)\)-entropic messages and pairwise independent hashing, as per Theorem 41. In this case, suppose for the LT we work modulo \(N^s\) for an RSA modulus \(N\) of length \(k\) and some \(s \in \mathbb{N}\). Then for the LE scheme we will work modulo \(N^{s+1}\) so that the output of the LT fits inside the message space of the LE scheme. Suppose we have \(\mu = csk\) for a constant \(c\). In the theorem we get \(r = \frac{1}{2}(csk + (s - 1)k - sk - k) - v = (cs - 2)k/2 - v\). Thus one could encrypt up to \((cs - 4)k/2\) messages with \(k\)-bit security. In the more general case of \((\mu, d)\)-entropic messages and \((v - d)\)-wise independent hashing, as per Theorem 40 we get \(r = (cs - d)k/3 - 2v(k - 1)/3\), which implies we can encrypt about \((cs - d)/2\) messages with \(k\)-bit security.

7.2.3 The D-SO-CCA Scheme

In order to achieve D-SO-CCA security, we need a few more building blocks that we introduce. The idea, following [27, 63, 84], is to combine the lossy TDF and lossy encryption with “all-but-\(N\)” analogues of these primitives so that we can switch from inverting one to inverting the other in answering decryption queries in the security proof. Interestingly, all-but-\(N\) lossy encryption has not been previously defined as it does not seem useful for achieving CCA security in the setting of R-PKE (see [63] for a more complex way of achieving SOA-CCA security for R-PKE that adapts this approach), but it turns out to work well in the D-PKE setting.

All-but-\(N\) Lossy Trapdoor Functions. An all-but-\(N\) lossy trapdoor function [63] with domain \(\text{Dom}(\cdot)\), range \(\text{Rng}(\cdot)\), tag-space \(\text{Tag}(\cdot)\) and lossiness \(\tau(\cdot)\) is a tuple of algorithms \(\text{NLT} = (\text{NLT.Kg}, \text{NLT.Eval}, \text{NLT.Inv})\) that work as follows. Algo-
Algorithm NLT.Kg on input a unary encoding of the security parameter $1^k$ and tags $t_1, \ldots, t_n \in \text{Tag}(k)$ outputs an evaluation key $ek$ and matching trapdoor $td$. Algorithm NLT.Eval on inputs an evaluation key $ek$, tag $t \in \text{Tag}(k)$ and $x \in \text{Dom}(k)$, outputs $y \in \text{Rng}(1^k)$. Algorithm NLT.Inv on inputs a trapdoor $td$, tag $t' \in \text{Tag}(k)$ and $y' \in \text{Rng}(k)$, outputs $x' \in \text{Dom}(k)$. We require the following properties.

**Correctness**: For all $k, n \in \mathbb{N}$, $t_1, \ldots, t_n \in \text{Tag}(k)$, all pairs $(ek, td) \in \text{NLT.Kg}(1^k, t_1, \ldots, t_n)$ we have $\text{NLT.Inv}(td, t', \text{NLT.Eval}(ek, t', x)) = x$ for every $x \in \text{Dom}(k)$ and and every $t' \in \text{Tag}(k)$ such that $t' \neq t_i$ for all $1 \leq i \leq n$.

**Key-indistinguishability**: For security we require that for any distinguisher $D$, we have $\text{Adv}_{\text{NLT},D}^{\text{abn}-\text{tdf}}(k) = \text{Pr}[D(ek) \Rightarrow 1 : (ek, td) \leftarrow \text{NLT.Kg}(1^k, t)] - \text{Pr}[D(ek') \Rightarrow 1 : (ek', td') \leftarrow \text{NLT.Kg}(1^k, t')]$ is negligible, for all $n \in \mathbb{N}$ and all $t, t' \in \text{Tag}^n(\cdot)$.

**Key-Lossiness**: The size of the co-domain of $\text{Eval}(ek, t, \cdot)$ is at most $|\text{Rng}(k)|/2^{r(k)}$ for all $k, n \in \mathbb{N}$, all $t_1, \ldots, t_n \in \text{Tag}(k)$, all $(ek, td) \in [\text{NLT.Kg}(1^k, t_1, \ldots, t_n)]$ and all $t = t_i$ for some $1 \leq i \leq n$.

**All-but-N Encryption.** We define a new “all-but-\(N\)” analogue of lossy encryption. A tag-based, randomized PKE with message space $\text{Msg}(\cdot)$, ciphertext space $\text{Ctxt}(\cdot)$, and tag space $\text{Tag}(\cdot)$ is an all-but-\(N\) lossy encryption scheme if it can be written as a tuple of algorithms $\text{NLE} = (\text{NLE.Kg}, \text{NLE.Enc}, \text{NLE.Dec})$ and a “lossy” ciphertext space $\text{LCtxt}(\cdot)$ with the following requirements. Algorithm NLE.Kg on input a unary encoding of the security parameter $1^k$ and tags $t_1, \ldots, t_n$ outputs a public key $pk$ and matching secret key $sk$. Algorithm NLE.Enc on input a public key $pk$, tag $t \in \text{Tag}(k)$, and a message $m \in \text{Msg}(k)$ outputs a ciphertext $c$. Algorithm NLE.Dec on input a secret key $sk$, tag $t \in \text{Tag}(k)$, and a ciphertext $c$ outputs a message $m'$. We require the following properties:
Correctness: For all $k, n \in \mathbb{N}$, all $t_1, \ldots, t_n \in \text{Tag}(k)$, all $t \in \text{Tag}(k)$ such that $t \neq t_i$ for all $1 \leq i \leq n$, and all $m \in \text{Msg}(k)$, $\Pr[\text{NLE.Dec}(sk, t, c) \Rightarrow m : (pk, sk) \leftarrow \text{NLE.Kg}(1^k, t_1, \ldots, t_n); c \leftarrow \text{NLE.Enc}(pk, t, m)] = 1$.

Indistinguishability of real and lossy keys: For every distinguisher $D$, for all $n \in \mathbb{N}$ and all $t, t' \in \text{Tag}^n(\cdot)$, it holds that $\text{Adv}_{\text{NLE},D}^{\text{abn-enc}}(k) = \Pr[D(pk) \Rightarrow 1 : (pk, sk) \leftarrow \text{NLE.Kg}(1^k, t)] - \Pr[D(pk') \Rightarrow 1 : (pk', sk') \leftarrow \text{NLE.Kg}(1^k, t')]$ is negligible.

Perfect lossiness: For any $k, n \in \mathbb{N}$, any $t_1, \ldots, t_n \in \text{Tag}(k)$, any $t \in \text{Tag}(k)$ such that $t = t_i$ for some $1 \leq i \leq n$, and any $m \in \text{Msg}(k)$

$$\Delta((pk, t, \text{NLE.Enc}(pk, t, m)), (pk, t, U)) = 0$$

where $(pk, sk) \leftarrow \text{NLE.Kg}(1^k, t_1, \ldots, t_n)$ and $U$ is uniform and independent on $\text{LCtxt}(k)$.

We construct such a scheme from Paillier’s DCR assumption, by adapting the Paillier-based all-but-$N$ TDF construction from [63]. Let $\mathcal{K}$ be an RSA modulus-generation algorithm, meaning on input $1^k$ it outputs $(N, p, q)$ where $N = pq$ and $p, q$ are $k/2$-bit primes. For $s \in \mathbb{N}$ define all-but-$N$ encryption scheme $\text{NLE} = (\text{NLE.Kg}, \text{NLE.Enc}, \text{NLE.Dec})$ given in Figure 7.3; here the message-space is $\mathbb{Z}_{N^s}$, the lossy ciphertext-space are the $N^s$-th residues modulo $\mathbb{Z}^{N^{s+1}}$, and the tag-space is $\mathbb{Z}^*_N$. It is easy to argue the above properties based on the analysis in [63]; we omit a formal statement.\footnote{Technically, message-space and tag-space should be adjusted so that they depend only on $k$ and “fit inside” the above sets for any choice of $N$; for example, in the case of the message-space this is done by taking all strings of length at most $ks - 1$. Moreover, the above properties should then be relaxed to hold only for efficiently generated tags rather than quantifying over all tags.}

The scheme. Let $\text{LT}$ be a lossy trapdoor function with domain $\text{LDom}(\cdot) = \{0, 1\}^{\text{LT}.\text{il}(\cdot)}$, range $\text{LRng}(\cdot)$, and lossiness $\tau_1$. Let $\text{NLT}$ be an all-but-$N$ trapdoor
Figure 7.3: Our Paillier-based all-but-\(N\) lossy encryption scheme.

The function with domain \(\text{NLTDom}(\cdot) = \{0, 1\}^{\text{NLT:il}(\cdot)}\) and range \(\text{NLTRng}(\cdot)\) and lossiness \(\tau_2\). Let \(\text{LE}\) be a perfectly-lossy encryption scheme with message-space \(\text{LRng}(\cdot)\) and coin-space \(\text{Coins}(\cdot) = \{0, 1\}^{\text{LE:rl}(\cdot)}\). Let \(\text{NLE}\) be an all-but-\(N\) encryption scheme with message-space \(\text{LRng}(\cdot) \times \text{NLTRng}(\cdot)\) and coin-space \(\text{NLECoins}(\cdot) = \{0, 1\}^{\text{NLE:rl}(\cdot)}\). Let \(H = (\text{HKg}, h)\) be a \(2t(\cdot)\)-wise independent hash function with domain \(\text{LDom}(\cdot)\) and range \(\text{LECoins}(\cdot) \times \text{NLECoins}(\cdot)\). Let \(\text{TCR}_1 = (\text{TCRKg}_1, \text{tcr}_1)\) be a target-collision resistant hash function with domain \(\text{LRng}(\cdot)\) and range \(\text{NLTTag}(\cdot)\). Let \(\text{TCR}_2 = (\text{TCRKg}_2, \text{tcr}_2)\) be a target-collision resistant hash function with domain \(\text{Ctxt}(\cdot)\) and range \(\text{NLETag}(\cdot)\). Define the associated deterministic encryption scheme \(\text{DE}[\text{LE}, \text{NLE}, \text{LT}, \text{NLT}, H, \text{TCR}_1, \text{TCR}_2] = (\text{DE.Kg}, \text{DE.Enc}, \text{DE.Dec})\) with message-space \(\text{LDom}(\cdot)\) in Figure 7.4.

The following theorems, respectively, characterize its security. Theorem 42 is proven in Appendix B.3 and the proof of Theorem 43 is analogous but concludes as in Theorem 41.

**Theorem 42** Let \(\text{DE}\) be the \(D\)-PKE scheme defined above. Let \(\mathcal{M}\) be a \((\mu, d)\)-entropic, efficiently resamplable message sampler and let \(v\) be the number of messages
where \( r = \min u \)

Theorem 43

that \( M \) produces. Assume \( H = (HKg, h) \) is a family of \( 2(v - d) \)-wise independent hash function. Then for any adversary \( A \) opening at most \( d \) ciphertexts, there are adversaries \( B, B', D_1, D_2, D_3 \), and \( D_4 \) such that

\[
\text{Adv}_{\text{DE}, A, M}^{d \text{-so-cca}}(\cdot) \leq 2 \cdot \text{Adv}_{\text{NLT}, D_1}^{\text{abn-ltdf}}(\cdot) + 2 \cdot \text{Adv}_{\text{LT}, D_2}^{\text{ltdf}}(\cdot) + 2 \cdot \text{Adv}_{\text{NLE}, D_3}^{\text{abn-le}}(\cdot) + 2 \cdot \text{Adv}_{\text{LE}, D_4}^{\text{le}}(\cdot) + 2v \cdot \text{Adv}_{\text{TCR}, B}^{\text{tcr}}(\cdot) + 2v \cdot \text{Adv}_{\text{TCR}, B'}^{\text{tcr}}(\cdot) + \frac{5v}{2^r} ,
\]

where \( r = \frac{1}{3}(\mu - (v - d)(\text{LT}.il + \text{NLT}.il + \text{LE}.rl + \text{NLE}.rl - \tau_1 - \tau_2)) - u + 1 \) and \( u = \min\{1 + d \log v, v\} \). The running time of each of each of \( B, B', D_1, D_2, D_3, \) and \( D_4 \) is about the same as that of \( A \) plus the running time of an efficient resampling algorithm of \( M \) plus the time to use \( \text{DE} \) to encrypt \( A \)'s messages.

Theorem 43 Let \( \text{DE} \) be as above. Let \( M \) be a \((\mu, \infty)\)-entropic, efficiently resamplable message sampler, and let \( v \) be the number of messages that \( M \) produces. Assume \( H = (HKg, h) \) is a family of pair-wise independent hash function. Then for any adversary \( A \),
there are adversaries $B, B', D_1, D_2, D_3,$ and $D_4$ such that

$$\text{Adv}^{d-so-cca}_{DE,A,M}(\cdot) \leq 2 \cdot \text{Adv}^{abn-\text{ldf}}_{\text{NLT},D_1}(\cdot) + 2 \cdot \text{Adv}^{\text{ldf}}_{\text{LT},D_2}(\cdot) + 2 \cdot \text{Adv}^{abn-\text{le}}_{\text{NLE},D_3}(\cdot)$$

$$+ 2 \cdot \text{Adv}^{\text{le}}_{\text{LE},D_4}(\cdot) + 2v \cdot \text{Adv}^{\text{TCR}}_{\text{TCR}_1,B}(\cdot) + 2v \cdot \text{Adv}^{\text{TCR}}_{\text{TCR}_2,B'}(\cdot) + \frac{v}{2r},$$

where $r = \frac{1}{2}(\mu - \text{LT}.\text{il} - \text{NLT}.\text{il} - \text{LE}.\text{rl} - \text{NLE}.\text{rl} + \tau_1 + \tau_2) - v$. The running time of each of $B, B', D_1, D_2, D_3,$ and $D_4$ is about the same as that of $A$ plus the running time of an efficient resampling algorithm of $M$ plus the time to use $DE$ to encrypt $A$’s messages.

As before, the scheme admits an efficient Paillier-based instantiations. For the TCR hash functions, one could use either a heuristic construction like SHA3 or, to keep with Paillier’s DCR assumption, the DCR-based collision-resistant hash given in [27].

### 7.3 Scheme in the Standard Model (II)

In this section, we give a new construction of deterministic public key encryption that is D-SO-CPA secure. We note that in our results, the size of the vector of messages is independent of the size of the public key. Our scheme $DE[\mathcal{H}, \mathcal{G}, LT]$ is shown in Figure 7.5, where $LT$ is a lossy trapdoor function and $\mathcal{H}, \mathcal{G}$ are hash functions. We begin by showing in Theorem 44 that $DE$ is D-SO-CPA secure for independent messages when $\mathcal{H}, \mathcal{G}$ are pair-wise independent hash functions and $LT$ is a regular lossy trapdoor function.

**Theorem 44** Let $\mathcal{M}$ be a $(\mu, 0)$-correlated, efficiently resamplable message sampler. Let $\mathcal{H} : \mathcal{K}_\mathcal{H} \times \{0, 1\}^n \rightarrow \{0, 1\}^\ell$ and $\mathcal{G} : \mathcal{K}_\mathcal{G} \times \{0, 1\}^\ell \rightarrow \{0, 1\}^n$ be a hash function families. Suppose $\mathcal{H}$ and $\mathcal{G}$ are pair-wise independent. Let $LT$ be a regular lossy trapdoor function with domain $\{0, 1\}^{n+\ell}$, range $\{0, 1\}^p$ and lossiness $\tau$. Let $DE[\mathcal{H}, \mathcal{G}, LT]$
DE.Kg(1^k)
(ek, td) ← s.LKg(1^k)
K_H ← s.K_H(1^k)
K_G ← s.K_G(1^k)
pk ← (K_H, K_G, ek)
sk ← (K_H, K_G, td)
Return (pk, sk)

DE.Enc(pk, m)
(K_H, K_G, ek) ← pk
r ← H(K_H, m)
y ← G(K_G, r) ⊕ m
c ← Eval(ek, y||r)
Return c

DE.Dec(sk, c)
(K_H, K_G, td) ← sk
y||r ← lnv(td, c)
m ← G(K_G, r) ⊕ y
Return m

| Figure 7.5: D-PKE scheme DE[\mathcal{H}, \mathcal{G}, \text{LT}]. |

be as defined in Figure 7.5. Then for any adversary A,

\[ \text{Adv}^{d-so-cpa}_{DE,A,M}(k) \leq 2 \cdot \text{Adv}^{ltdf}_{LT,B}(k) + 2592v\sqrt{2^{1-\mu-2\tau+2p}}. \]

**Proof.** We begin by showing the following lemma.

**Lemma 45** Let \( \mathcal{H} : K_H \times \{0,1\}^n \rightarrow \{0,1\}^\ell \) and \( \mathcal{G} : K_G \times \{0,1\}^\ell \rightarrow \{0,1\}^n \) be a hash function families. Suppose \( \mathcal{H} \) and \( \mathcal{G} \) are pair-wise independent. Let \( \text{LT} \) be a regular lossy trapdoor function with domain \( \{0,1\}^{n+\ell} \), range \( \{0,1\}^p \) and lossiness \( \tau \). Let \( X \) be a random variable over \( \{0,1\}^n \) such that \( H_\infty(X) \geq \eta \). Then, for all \( \ell k \in [LKg(1^k)] \), all \( c \in \text{img}(lk) \) and any \( \epsilon > 0 \),

\[
\left| \Pr[\text{DE.Enc}(pk, X) = c] - 2^{\tau-p} \right| \geq \epsilon 2^{\tau-p},
\]

for at most \( 2^{-u} \) fraction of public key \( pk \), where \( u = \eta + 2\tau - 2p - 2\log(1/\epsilon) \).

**Proof of Lemma 45.** We will need the following tail inequality for pair-wise independent distributions.

**Claim 46** Let \( A_1, \ldots, A_n \) be pair-wise independent random variables in the interval [0,1]. Let \( A = \sum_i A_i \) and \( \mathbb{E}(A) = \mu \) and \( \delta > 0 \). Then,

\[
\Pr[|A - \mu| > \delta \mu] \leq \frac{1}{\delta^2 \mu}.
\]
Proof of Claim 46. From Chebychev’s inequality, for any $\delta > 0$ we have

$$\Pr[|A - \mu| > \delta \mu] \leq \frac{\text{Var}[A]}{\delta^2 \mu^2}.$$  

Note that $A_1, \cdots, A_n$ are pair-wise independent random variables. Thus, we have $\text{Var}[A] = \sum_i \text{Var}[A_i]$. Moreover, we know that $\text{Var}[A_i] \leq \mathbb{E}(A_i)$ for all $i \in [n]$, since the random variable $A_i$ is in the interval $[0, 1]$. Therefore, we have $\text{Var}[A] \leq \mu$. This completes the proof of Claim 46.

We now define $p_x = \Pr[X = x]$, for any $x \in \{0, 1\}^n$. We consider the probability over the choice of public key $pk$. Fix the lossy key $lk \in [\text{LKg}(1^k)]$, we consider the probability over the choice of $K_H, K_G$. For every $x \in \{0, 1\}^n$ and $c \in \text{Img}(lk)$, we also define the following random variable

$$Z_{x,c} = \begin{cases} p_x & \text{if } \text{DE.Enc}(pk, x) = c \\ 0 & \text{otherwise} \end{cases}$$

Let $A_{x,c} = Z_{x,c} 2^\eta$. Note that that for every $x$, $H(K_H, x)$ is uniformly distributed, over the uniformly random choice of $K_H$. Moreover, for every $x$ and $K_H$, $G(K_G, H(K_H, x))$ is uniformly distributed, over the uniformly random choice of $K_G$. Since LT is a regular LTDF, we have $\mathbb{E}(Z_{x,c}) = p_x \cdot 2^{\tau - p}$, for every $x, c$. Let $Z_c = \sum_x Z_{x,c}$ and $A_c = \sum_x A_{x,c}$. Then, we have $\mathbb{E}(Z_c) = 2^{\tau - p}$ and $\mathbb{E}(A_c) = 2^{\eta + \tau - p}$. Moreover, for every $x, c$, we know $A_{x,c} \in [0, 1]$ and for every $c$, the variables $A_{x,c}$ are pair-wise independent. Applying Claim 46, we obtain that for every $c$ and $\delta > 0$

$$\Pr[|A_c - 2^{\eta + \tau - p}| \geq \delta \cdot 2^{\eta + \tau - p}] \leq \frac{2^{2p - \eta - \tau}}{\delta^2}.$$  

Substituting $Z_c$ for $A_c$ and choosing $\delta = \epsilon$, we obtain that for every $\epsilon > 0$,

$$\Pr[|Z_c - 2^{\tau - p}| \geq \epsilon \cdot 2^{\tau - p}] \leq \frac{2^{2p - \eta - \tau}}{\epsilon^2}.$$  

163
Using union bound, we obtain that $|Z_{c} - 2^{r-p}| \geq \epsilon \cdot 2^{r-p}$ with probability $2^{2p-n-2r}/\epsilon^{2} = 2^{-u}$ over the choice of $K_{H}, K_{G}$, for all $lk \in [LKg(1^{k})]$, all $c \in \text{Img}(lk)$. This completes the proof of Lemma 45.

Consider games $G_{0}, G_{1}$ in Figure 7.6. Then $\text{Adv}_{d-so-cpa}^{\text{DE},A,M}(k) = 2 \cdot \Pr [G_{0}(k) \Rightarrow 1] - 1$. We now explain the game chain. Game $G_{1}$ is identical to game $G_{0}$, except that instead of generating an injective key for the lossy trapdoor function, we generate a lossy one. Consider the following adversary $B$ attacking the key indistinguishability of $\text{LT}$. It simulates game $G_{0}$, but uses its given key instead of generating a new one. It outputs 1 if the simulated game returns 1, and outputs 0 otherwise. Then

$$\Pr [G_{0}(k) \Rightarrow 1] - \Pr [G_{1}(k) \Rightarrow 1] \leq \text{Adv}_{\text{LT},B}^{\text{ldf}}(k) .$$

Utilizing similar approach from [96, Theorem 4.1] and Lemma 45, we obtain that

$$\Pr [G_{1}(k) \Rightarrow 1] \leq 1296v^{\sqrt{2^{1-\mu-2r+2p}}} + \frac{1}{2} .$$

Summing up, $\text{Adv}_{d-so-cpa}^{\text{DE},A,M}(k) \leq 2 \cdot \text{Adv}_{\text{LT},B}^{\text{ldf}}(k) + 2592v^{\sqrt{2^{1-\mu-2r+2p}}}$. This completes the proof of Theorem 44.
We now extend our result to include correlated messages. We show that it is enough to use $2t$-wise independent hash functions to extend the security to $t$-correlated messages. Let $\text{DE}[\mathcal{H}, \mathcal{G}, \text{LT}]$ be PKE scheme shown in Figure 7.5, where LT is a lossy trapdoor function and $\mathcal{H}, \mathcal{G}$ are hash functions. We show in Theorem 47 that $\text{DE}$ is D-SO-CPA secure for $t$-correlated messages when $\mathcal{H}, \mathcal{G}$ are $2t$-wise independent hash functions and LT is a regular lossy trapdoor function.

**Theorem 47** Let $\mathcal{M}$ be a $(\mu, d)$-correlated, efficiently resamplable message sampler. Let $\mathcal{H} : \mathcal{K}_H \times \{0, 1\}^n \rightarrow \{0, 1\}^\ell$ and $\mathcal{G} : \mathcal{K}_G \times \{0, 1\}^\ell \rightarrow \{0, 1\}^n$ be a hash function families. Suppose $\mathcal{H}$ and $\mathcal{G}$ are $2d$-wise independent. Let $\text{LT}$ be a regular lossy trapdoor function with domain $\{0, 1\}^{n+\ell}$, range $\{0, 1\}^p$ and lossiness $\tau$. Let $\text{DE}[\mathcal{H}, \mathcal{G}, \text{LT}]$ be as defined in Figure 7.5. Then for any adversary $A$,

$$\text{Adv}^{d\text{-so-CPA}}_{\text{DE}, A, \mathcal{M}}(k) \leq 2 \cdot \text{Adv}^{\text{ltdf}}_{\text{LT}, B}(k) + 2592 \sqrt{2^{1-\mu+2d(-\tau+p)}}.$$

The proof of Theorem 47 is very similar to the proof of Theorem 44. We refer to [96, Theorem 4.5] for the proof of Theorem 47.
Appendix A

Generalized SIE and CIE of RSA

In this section, we show black-box extractability properties of RSA, generalizing the work of Barthe et al. [3]. Namely, we show that RSA with small exponent is \((i, j)\)-second input extractable and \((i, j)\)-common input extractable for certain parameters \(i, j\).

Our proofs rely on Coppersmith’s technique [45] to find small integer roots of univariate and bivariate polynomials modulo \(N\) with unknown factorization. Let us state the results we use.

**Proposition 48 (Univariate Coppersmith)** There is an algorithm that on inputs a monic integer polynomial \(p(X)\) of degree \(\delta\) with integer coefficients, and a positive integer \(N\), outputting all integer solutions \(x_0\) to \(p(x_0) = 0 \mod N\) with \(|x_0| < N^{1/\delta}\) in time polynomial in \(\log(N)\) and \(\delta\).

**Proposition 49 (Bivariate Coppersmith (Heuristic))** There is an algorithm that on inputs a polynomial \(p(X,Y)\) of total degree \(\delta\) with a monic monomial \(X^aY^{\delta-a}\) for some \(a\), and a positive integer \(N\), outputting all integer solutions \(x_0, y_0\) to \(p(x_0, y_0) = 0 \mod N\) with \(|x_0y_0| < N^{1/\delta}\) in time polynomial in \(\log(N)\) and \(\delta\).

Note while the bivariate Coppersmith algorithm is not know to provably run in polynomial-time, [22, 31, 51, 72] shows it works well in practice.

We now give the main results of this section. We use \(t_{\text{Cop}}(N,\delta)\) to denote the maximum running-time of the univariate and bivariate Coppersmith algorithms on
inputs as above. We also use $t_{Euc(N,\delta)}$ to denote the maximum running-time of the extended Euclidean algorithm on two univariate polynomials of at most degree $\delta$ over $\mathbb{Z}_N^*$. Recall the RSA trapdoor permutation family, parameterized by $N, e$ where $n = \lceil \log N \rceil$, is defined as $f_{N,e}(x) = x^e \mod N$ for $x \in \mathbb{Z}_N^*$.

**Theorem 50** RSA trapdoor permutation family is $(i,j)$-second input extractable for $j - i > (1 - 1/e)n$. The extractor runs in time $t_{Cop(N,e)}$.

**Theorem 51** RSA trapdoor permutation family is $(i,j)$-common input extractable for $j - i > (1 - 1/e^2)n$. The extractor runs in time $t_{Cop(N,e^2)} + t_{Euc(N,e)}$.

**Proofs of Theorem 50.** Firstly, let’s recall the definition of $(i,j)$-second input extractable. Let $\mathcal{F} = (Kg, Eval, Inv)$ be a trapdoor permutation family with domain $TDom$. For $i,j \in \mathbb{N}$, we say $\mathcal{F}$ is $(i,j)$-second input extractable if there exists an efficient extractor $E$ such that for every $f \in [Kg(1^k)]$ and every $x \in TDom(k)$, extractor $E$ on inputs $f, f(x), x|_{i+1}^j$ outputs $x$.

For any element $x \in \mathbb{Z}_N^*$ and $i,j \in [0, n], i < j$, $x$ can be uniquely represented as $x = s \cdot 2^j + r \cdot 2^i + t$, where $s \in \{0, 1\}^{n-j}$, $r \in \{0, 1\}^{j-i}$, and $t \in \{0, 1\}^i$. Notice that if $j = n$ or $i = 0$, we will remove $s$ or $t$ from the formula respectively. Now, we can rewrite RSA as a function of three arguments:

$$f_{N,e}(x) = f_{N,e}(s,r,t) = (s \cdot 2^j + r \cdot 2^i + t)^e \mod N.$$

The high level idea for $(i,j)$-second input extractable is to solve the monic integer polynomial through coppersmith algorithm. Specifically, we will construct the extractor $E$ in several cases:

- $i = 0$: Then the item $t$ will be removed from RSA function, and we have $f_{N,e}(x) = f_{N,e}(s,r) = (s \cdot 2^j + r)^e \mod N$. To construct an extractor $E$ on inputs

---

1Although $\mathbb{Z}_N^*$ is not a field, if the algorithm fails it can recover a non-trivial factor of $N$. 167
\( r = x^{i+1}_j \) and \( c = f_{N_e}(x) \), we can consider the polynomial \( p(X) = 0 \mod N \) for \( p(X) = (X \cdot 2^j + r)^e - c \). The Coppersmith univariate algorithm requires monic polynomial to find the root \( s \). However, \( p(X) \) is not monic polynomial.

Notice that \( j \) and \( e \) are public, so we can easily find the inverse of \( 2^{je} \in \mathbb{Z}_N^* \), and multiply \( p(X) \) to get a new monic polynomial. On the other hand, the Coppersmith algorithm can find only roots \( s < N^{1/e} \), which means \( 2^{n-j} < N^{1/e} \), or equivalently, \( j > (1-1/e)n \). The running time of extractor \( \mathcal{E} \) can be bounded by the running time of Coppersmith algorithm \( t_{\text{Cop}(N,e)} \).

- \( j = n \): This work has been shown in section 5.1 of [3]. The requirement for \( i \) is \( i < n/e \) and the extractor \( \mathcal{E} \) runs within time \( t_{\text{Cop}(N,e)} \).

- \( i > 0 \) and \( j < n \): This case will be slightly different from the first case.

  The extractor \( \mathcal{E} \) on inputs \( r = x^{i+1}_j \) and \( c = f_{N_e}(x) \) outputs \( s, t \) such that \( f_{N_e}(s, r, t) = c \). By using the same strategy, we construct polynomial \( p(X, Y) = (X \cdot 2^j + r \cdot 2^i + Y)^e - c \mod N \) with two variables \( X \) and \( Y \). The bivariate Coppersmith algorithm could find all integer solutions \( x_0, y_0 \) such that \( |x_0 y_0| < N^{1/e} \), which equals to \( 2^{n-j} \cdot 2^i < N^{1/e} \), or in other words, such that \( j - i > (1-1/e)n \).

  The extractor \( \mathcal{E} \) executes within time \( t_{\text{Cop}(N,e)} \).

Combining these 3 cases, we thus construct an efficient \((i, j)\)-second input extractable algorithm \( \mathcal{E} \) running within time \( t_{\text{Cop}(N,e)} \) when \( j - i > (1 - 1/e)n \).

Proofs of the Theorem 51. Again, let’s recall the definition of \((i, j)\)-common input extractable. Let \( \mathcal{F} = (\text{Kg, Eval, Inv}) \) be a trapdoor permutation family with domain \( \text{TDom} \). For \( i, j \in \mathbb{N} \), we say \( \mathcal{F} \) is \((i, j)\)-common input extractable if there exists an efficient extractor \( \mathcal{E} \) such that for every \( f \in [\text{Kg}(1^k)] \) and every \( x_1, x_2 \in \text{TDom}(k) \), extractor \( \mathcal{E} \) on inputs \( f, f(x_1), f(x_2) \) outputs \( x_1, x_2 \) if \( x_1|_i^j = x_2|_i^j \).
Given two different $c_1 = f(x_1), c_2 = f(x_2)$, our goal is to find $s_1, r, t_1$ and $s_2, r, t_2$ such that $c_1 = (s_1 \cdot 2^i + r \cdot 2^i + t_1)^e \mod N$ and $c_2 = (s_2 \cdot 2^i + r \cdot 2^i + t_2)^e \mod N$. Let us consider several cases:

- $i = 0$: In this case, $t_1$ and $t_2$ will be removed in the formula. Consider two polynomials $p_1(X, Y) = X^e - c_1 \mod N$ and $p_2(X, Y) = (X + Y \cdot 2^i)^e - c_2 \mod N$. When $x_0 = s_1 \cdot 2^i + r$ and $y_0 = s_2 - s_1$, both polynomials evaluate to 0. Taking $p_1(X, Y)$ and $p_2(X, Y)$ as one variable polynomial over $X$, the determinant of the $2e \times 2e$ Sylvester Matrix is a polynomial in $Y$. On the other hand, the resultant $Res(p_1, p_2, X)$, which equals to the determinant of the Sylvester Matrix, has root at point $Y = y_0$ since at point $Y = y_0$, $p_1(X, y_0)$ and $p_2(X, y_0)$ will share the same root $x_0$. Therefore, once we get $Res(p_1, p_2, X)$ by computing Sylvester Matrix, we can use univariate Coppersmith algorithm solve polynomial $Res(p_1, p_2, X)$. Notice the specific form of the Sylvester Matrix, a straightforward but tedious calculation shows that the degree of $Res(p_1, p_2, X)$ is $e^2$ and the coefficient of $Y^{e^2}$ is $2^{je^2}$. We can easily adjust the coefficient of $Y^{e^2}$ to 1 by multiplying the inverse of $2^{je^2} \in \mathbb{Z}_N^*$. The univariate Coppersmith algorithm requires $|y_0| < N^{1/e^2}$, or equivalently, $j > (1 - 1/e^2)n$. Once we work out $y_0, p_1(X, y_0)$ and $p_2(X, y_0)$ share the same and unique root $x_0$. Hence, $x - x_0$ (or power of $(x - x_0)$) is a common factor of these two polynomials and can be found by extended Euclidean algorithm. The running time of extractor $E$ could be bounded by the running time of Coppersmith algorithm $t_{C_{op}(N, e^2)}$ and the running time of extended Euclidean algorithm $t_{Euc(N, e)}$.

- $j = n$: This work has also been shown in section 5.1 of [3]. The requirement for $i$ is $i < n/e^2$ and the extractor $E$ runs within time $t_{C_{op}(N, e^2)} + t_{Euc(N, e)}$. 

169
• $i > 0$ AND $j < n$: The high level idea is almost the same as the first case, while the detail differs. Consider two polynomials $p_1(X, Y_1, Y_2) = X^e - c_1 \mod N$ and $p_2(X, Y_1, Y_2) = (X + Y_1 \cdot 2^j + Y_2)^e - c_2 \mod N$. Both polynomials should be equal to 0 at point $(x_0 = s_1 \cdot 2^j + r \cdot 2^i + t_1, y_1 = s_2 - s_1, y_2 = t_2 - t_1)$. Hence, the resultant polynomial $Res(p_1, p_2, X)$ over $X$ has roots $y_1$ and $y_2$, since $p_1(X, y_1, y_2)$ and $p_2(X, y_1, y_2)$ share the same root $x_0$. On the other hand, The determinant of the $2e \times 2e$ Sylvester Matrix associated to the polynomial $p_1$ and $p_2$ over $X$, which equal to the resultant polynomial $Res(p_1, p_2, X)$, is a polynomial with total degree $e^2$ and has one monic monomial $Y_2^{e^2}$. Therefore, we can use bivariate Coppersmith algorithm get the roots $y_1$ and $y_2$ for polynomial $Res(p_1, p_2, X)$.

Notice that bivariate Coppersmith algorithm requires $|y_1y_2| < N^{1/e^2}$, which implies $j - i > (1 - 1/e^2)n$. The following part, including solving $x_0$ and running time will be same as the first case.

In summary, we have an efficient $(i, j)$-common input extractable algorithm for RSA if $j - i > (1 - 1/e^2)n$, as required.
B.1 Proof of Theorem 40

In the proof we make use of the following well-known properties of statistical distance.

**Lemma 52** *(Properties of statistical distance).* Let $X, Y, Z$ be random variables taking values in a universe $U$. Then,

1. $0 \leq \Delta(X, Y) \leq 1$, with equality iff $X$ and $Y$ are identically distributed,
2. $\Delta(X, Y) \leq \Delta(Y, X)$, and
3. $\Delta(X, Z) \leq \Delta(X, Y) + \Delta(Y, Z)$.

We also need the following properties of average min-entropy, given by Dodis, Ostrovsky, Reyzin, and Smith [50].

**Lemma 53** *[50]* Let $X, Y, Z$ be random variables and $\delta > 0$ be a real number.

(a) If $Y$ has at most $2^\lambda$ possible values then we have $\tilde{H}_\infty(X \mid Z, Y) \geq \tilde{H}_\infty(X \mid Z) - \lambda$.

(b) Let $S$ be the set of values $b$ such that $H_\infty(X \mid Y = b) \geq \tilde{H}_\infty(X \mid Y) - \log(1/\delta)$. Then it holds that $\Pr[Y \in S] \geq 1 - \delta$.

**The main proof.** In this proof, we adopt the following notion. If $f$ is a function on domain $S$ and $Y$ is a vector $(Y[1], \ldots, Y[r])$ whose components are elements of $S$ then $f(Y)$ is the vector $(f(Y[1]), \ldots, f(Y[r]))$. We only need to consider the case
\[G_0(k)\]

- \(b \leftarrow \{0, 1\}\)
- \(\param \leftarrow A.pg(1^k)\)
- \(\text{pk} \leftarrow \mathcal{M}(1^k, \param)\)
- \(\text{sk} \leftarrow \text{LE.IKg}(1^k)\)
- \(\text{pk} \leftarrow \text{LE.LKg}(1^k)\)
- \(\text{h} \leftarrow \text{h}((\text{pk}, \text{ek}, \text{K}))\)
- \(\text{y} \leftarrow \text{Eval}(\text{ek}, \text{m}_1)\)
- \(c \leftarrow \text{LE.Enc}(\text{pk}, \text{y}; \text{h})\)
- \(\text{m}_0 \leftarrow \text{Rsmp}(1^k, I, \text{m}_1[I], \param)\)
- \(\omega \leftarrow A.g((\text{state}, \text{m}_1[I]), \param)\)
- \(\text{t} \leftarrow A.f(\text{m}_b, \param)\)
- If \(\text{t} = \omega\) then return \(b\)
- Else return \(1 - b\)

\[G_1(k)\]

- \(b \leftarrow \{0, 1\}\)
- \(\param \leftarrow A.pg(1^k)\)
- \(\text{pk} \leftarrow \mathcal{M}(1^k, \param)\)
- \(\text{pk} \leftarrow \text{LE.LKg}(1^k)\)
- \(\text{h} \leftarrow \text{h}((\text{pk}, \text{ek}, \text{K}))\)
- \(\text{y} \leftarrow \text{Eval}(\text{ek}, \text{m}_1)\)
- \(c \leftarrow \text{LE.Enc}(\text{pk}', \text{y}; \text{h})\)
- \(\text{m}_0 \leftarrow \text{Rsmp}(1^k, I, \text{m}_1[I], \param)\)
- \(\omega \leftarrow A.g((\text{state}, \text{m}_1[I]), \param)\)
- \(\text{t} \leftarrow A.f(\text{m}_b, \param)\)
- If \(\text{t} = \omega\) then return \(b\)
- Else return \(1 - b\)

**Figure B.1:** Games \(G_0\) and \(G_1\) of the proof of Theorem 40 and Theorem 41.

\[d < v, \text{ since the case } d = v \text{ has a better bound via Theorem 41. Let } \text{Rsmp} \text{ be an efficient resampling algorithm of } \mathcal{M}. \text{ Below, for a set } S, \text{ we write } \overline{S} \text{ for } [v(k)] \setminus S.\]

Consider games \(G_0 - G_5\) in Figures B.1 and B.2. Then

\[\text{Adv}^{d-so-cpa}_{\text{DE[LE,LT,H],A,M}}(\cdot) = 2\Pr[G_0(\cdot) = 1] - 1.\]

We now explain the game chain. Game \(G_1\) is identical to game \(G_0\), but instead of using an injective key generated by \(\text{LE.IKg}(1^k)\), we use a lossy key generated by \(\text{LE.LKg}(1^k)\). Consider the following adversary \(B\) attacking the key indistinguishability of \(\text{LE}\). It simulates game \(G_0\), but uses its given key instead of generating an injective key. It outputs 1 if the simulated game returns 1, and outputs 0 otherwise. Then

\[\Pr[G_0(\cdot) = 1] - \Pr[G_1(\cdot) = 1] \leq \text{Adv}^{k}_{\text{LE,B}}(\cdot).\]

Next, game \(G_2\) is identical to game \(G_1\), except that instead of generating an injective key for the lossy trapdoor function, we generate a lossy one. Consider the
following adversary $D$ attacking the key indistinguishability of $LT$. It simulates game $G_1$, but uses its given key instead of generating a new one. It outputs 1 if the simulated

<table>
<thead>
<tr>
<th>Game $G_2(k)$</th>
<th>Game $G_3(k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b \leftarrow {0, 1}$ ; $\text{param} \leftarrow \text{A.pg}(1^k)$</td>
<td>$b \leftarrow {0, 1}$ ; $I^* \leftarrow \mathcal{I}$ ; $\text{param} \leftarrow \text{A.pg}(1^k)$</td>
</tr>
<tr>
<td>$\text{m}_1 \leftarrow \mathcal{M}(1^k, \text{param})$</td>
<td>$\text{m}_1 \leftarrow \mathcal{M}(1^k, \text{param})$</td>
</tr>
<tr>
<td>$pk' \leftarrow \text{LE.LKg}(1^k)$ ; $ek' \leftarrow \text{LKg}(1^k)$</td>
<td>$pk' \leftarrow \text{LE.LKg}(1^k)$ ; $ek' \leftarrow \text{LKg}(1^k)$</td>
</tr>
<tr>
<td>$K \leftarrow \text{HKg}(1^k)$ ; $h \leftarrow h(K, \text{m}_1)$</td>
<td>$K \leftarrow \text{HKg}(1^k)$ ; $h \leftarrow h(K, \text{m}_1)$</td>
</tr>
<tr>
<td>$\overline{pk} \leftarrow (pk', ek', K)$</td>
<td>$\overline{pk} \leftarrow (pk', ek', K)$</td>
</tr>
<tr>
<td>$y \leftarrow \text{Eval}(ek', \text{m}_1)$</td>
<td>$y \leftarrow \text{Eval}(ek', \text{m}_1)$</td>
</tr>
<tr>
<td>$c \leftarrow \text{LE.Enc}(pk', y; h)$</td>
<td>$c \leftarrow \text{LE.Enc}(pk', y; h)$</td>
</tr>
<tr>
<td>$(\text{state}, I) \leftarrow \text{A.cor}(\overline{pk}, c, \text{param})$</td>
<td>$(\text{state}, I) \leftarrow \text{A.cor}(\overline{pk}, c, \text{param})$</td>
</tr>
<tr>
<td>$\text{m}_0 \leftarrow \text{Rsm}(1^k, \text{m}_1[I], I, \text{param})$</td>
<td>If $I \neq I^*$ then</td>
</tr>
<tr>
<td>$\omega \leftarrow \text{A.g}(\text{state}, \text{m}_1[I], \text{param})$</td>
<td>$s \leftarrow {0, 1}$ ; Return $s$</td>
</tr>
<tr>
<td>$t \leftarrow \text{A.f}(\text{m}_0, \text{param})$</td>
<td>$\text{m}_0 \leftarrow \text{Rsm}(1^k, \text{m}_1[I], I, \text{param})$</td>
</tr>
<tr>
<td>If $(t = \omega)$ then return $b$</td>
<td>$\omega \leftarrow \text{A.g}(\text{state}, \text{m}_1[I], \text{param})$</td>
</tr>
<tr>
<td>Else return $1 - b$</td>
<td>$t \leftarrow \text{A.f}(\text{m}_0, \text{param})$</td>
</tr>
<tr>
<td>If $(t = \omega)$ then return $b$ Else return $1 - b$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Game $G_4(k)$</th>
<th>Game $G_5(k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b \leftarrow {0, 1}$ ; $I^* \leftarrow \mathcal{I}$</td>
<td>$b \leftarrow {0, 1}$ ; $I^* \leftarrow \mathcal{I}$</td>
</tr>
<tr>
<td>$\text{param} \leftarrow \text{A.pg}(1^k)$</td>
<td>$\text{param} \leftarrow \text{A.pg}(1^k)$</td>
</tr>
<tr>
<td>$\text{m}_1 \leftarrow \mathcal{M}(1^k, \text{param})$</td>
<td>$\text{m}_1 \leftarrow \mathcal{M}(1^k, \text{param})$</td>
</tr>
<tr>
<td>$pk' \leftarrow \text{LE.LKg}(1^k)$ ; $ek' \leftarrow \text{LKg}(1^k)$</td>
<td>$pk' \leftarrow \text{LE.LKg}(1^k)$ ; $ek' \leftarrow \text{LKg}(1^k)$</td>
</tr>
<tr>
<td>$K \leftarrow \text{HKg}(1^k)$</td>
<td>$K \leftarrow \text{HKg}(1^k)$</td>
</tr>
<tr>
<td>$\overline{pk} \leftarrow (pk', ek', K)$</td>
<td>$\overline{pk} \leftarrow (pk', ek', K)$</td>
</tr>
<tr>
<td>For $i \in \overline{I^*}$ do $h[i] \leftarrow \text{HRng}(k)$</td>
<td>For $i \in \overline{I^*}$ do:</td>
</tr>
<tr>
<td>For $i \in I^*$ do $h[i] \leftarrow h(K, \text{m}_1[i])$</td>
<td>$c[i] \leftarrow \text{Ctxt}(k)$</td>
</tr>
<tr>
<td>$y \leftarrow \text{Eval}(ek', \text{m}_1)$</td>
<td>For $i \in I^*$ do:</td>
</tr>
<tr>
<td>$c \leftarrow \text{LE.Enc}(pk', y; h)$</td>
<td>$h[i] \leftarrow h(K, \text{m}_1[i])$</td>
</tr>
<tr>
<td>$(\text{state}, I) \leftarrow \text{A.cor}(\overline{pk}, c, \text{param})$</td>
<td>$y[i] \leftarrow \text{Eval}(ek', \text{m}_1[i])$</td>
</tr>
<tr>
<td>If $I \neq I^*$ then $s \leftarrow {0, 1}$ ; return $s$</td>
<td>$c[i] \leftarrow \text{LE.Enc}(pk', y[i]; h[i])$</td>
</tr>
<tr>
<td>$\text{m}_0 \leftarrow \text{Rsm}(1^k, \text{m}_1[I], I, \text{param})$</td>
<td>$(\text{state}, I) \leftarrow \text{A.cor}(\overline{pk}, c, \text{param})$</td>
</tr>
<tr>
<td>$\omega \leftarrow \text{A.g}(\text{state}, \text{m}_1[I], \text{param})$</td>
<td>If $I \neq I^*$ then $s \leftarrow {0, 1}$ ; Return $s$</td>
</tr>
<tr>
<td>$t \leftarrow \text{A.f}(\text{m}_0, \text{param})$</td>
<td>$\text{m}_0 \leftarrow \text{Rsm}(1^k, \text{m}_1[I], I, \text{param})$</td>
</tr>
<tr>
<td>If $(t = \omega)$ then return $b$</td>
<td>$\omega \leftarrow \text{A.g}(\text{state}, \text{m}_1[I], \text{param})$</td>
</tr>
<tr>
<td>Else return $1 - b$</td>
<td>$t \leftarrow \text{A.f}(\text{m}_0, \text{param})$</td>
</tr>
<tr>
<td>If $(t = \omega)$ then return $b$ else return $1 - b$</td>
<td></td>
</tr>
</tbody>
</table>

Figure B.2: Games $G_2$–$G_5$ of the proof of Theorem 40 and Theorem 41.
game returns 1, and outputs 0 otherwise. Then

$$\Pr[G_1(\cdot) \Rightarrow 1] - \Pr[G_2(\cdot) \Rightarrow 1] \leq \text{Adv}_{\text{LT},D}(\cdot) .$$

Next, in game $G_3$, instead of using the set $I$ generated by the adversary, we try to guess it by picking a random subset $I^*$ of $\{1, \ldots, v(k)\}$ of size at most $d$. If our guess is incorrect, meaning $I \neq I^*$ then we output a random bit $s \leftarrow \{0, 1\}$. Let $V$ be the number of subsets of $[v(k)]$ that contains at most $d$ elements. A trivial bound for $V$ is $2^v$. We now show that $V \leq 2^v d$. This holds for $d \in \{0, 1\}$. For $v \geq d \geq 2$, $V = \sum_{i=0}^{d} \binom{v}{i} \leq \sum_{i=0}^{d} \frac{v^i}{i!} \leq 1 + v + \sum_{i=2}^{\infty} \frac{v^d}{2^i} = 1 + v + 0.5v^d \leq 2v^d$, as claimed. Hence $V \leq 2^u$, with $u = \min\{1 + d \log(v), v\}$, and thus $\Pr[I = I^*] = 1/V \geq 2^{-u}$. Then

$$(\Pr[G_2(\cdot) \Rightarrow 1] - 1/2) \leq 2^u \cdot (\Pr[G_3(\cdot) \Rightarrow 1] - 1/2) .$$

Next, game $G_4$ is identical to game $G_3$, except for unopened messages, we are using completely random coins in the encryption phase instead of using the hash of the messages as coins. Let $\text{pars} = (K, \text{param}, ek^l, I^*)$. For each $i \in [v(k)]$, let $U[i]$ be a fresh random string uniformly distributed over $\text{Coins}(k)$. For each $i \in [v(k)]$, let $Y[i] = U[i]$ if $i \in [v(k)] \setminus I^*$, and let $Y[i] = h(K, m_1[i])$ otherwise. For each fixed choice of $I^*$,

$$\Pr[G_3 \Rightarrow 1] - \Pr[G_4 \Rightarrow 1] \leq \Delta((\text{pars}, h(K, m_1), y), (\text{pars}, Y, y)) .$$

Let $\ell = d - |I^*|$. Partition $[v(k)] \setminus I^*$ to $L_1, \ldots, L_{\ell+1}$ such that $|L_1| = v(k) - d$, and each other $L_i$ is a singleton set. Let $J_0 = I^*$ and $J_{i+1} = J_i \cup L_{i+1}$ for every $i \in \{1, \ldots, \ell\}$. For each $j \leq \{0, 1, \ldots, \ell + 1\}$, let $X_j$ be the vector of $v(k)$ components such that $X_j[i] = Y[i]$ for every $i \in J_j$, and $X_j[i] = h(K, m_1[i])$ for every $i \in \overline{J}_j$. 174
Then \(X_0 = h(K, m_1)\) and \(X_{\ell+1} = Y\). From the triangle inequality,

\[
\Delta((h(K, \text{pars}, m_1), y), (Y, \text{pars}, y)) \\
\leq \sum_{j=0}^{\ell} \Delta((X_j, \text{pars}, y), (X_{j+1}, \text{pars}, y)) \\
\leq \sum_{j=1}^{\ell+1} \Delta((h(K, m_1[L_j]), m_1[L_1 \setminus L_j], \text{pars}, \text{y}), (U[L_j], m_1[L_1 \setminus L_j], \text{pars}, \text{y}));
\]

the last inequality holds because each \(h(K, m_1[i])\) is completely determined from \(m_1[i]\) and \(\text{pars}\). Note that each component of \(y\) has at most \(2^{\text{LT}_i(k) - \tau(k)}\) values. Then for each \(j \in \{1, \ldots, \ell + 1\}\) and each \(i \in L_j\), since \(\mathcal{M}\) is \((\mu, d)\)-entropic and \(|L_1 \setminus L_j| \leq |L_1| = d\), from property (a) of Lemma 53, \(H_\infty(m_1[i] \mid \text{pars}, y, m_1[L_1 \setminus L_j]) \geq \mu - (v - d)(\text{LT}_i \cdot \text{il}_j - \tau)\). Next, we need the following lemma whose proof we’ll give later.

**Lemma 54** Let \(H = (\text{HKg}, h)\) be a \(2t\)-wise independent hash function with domain \(D\) and range \(R\). Let \(Z\) be a random variable and \(X = (X_1, \cdots, X_t)\) where \(X_i\) is a random variable over \(D\) such that \(\tilde{H}_\infty(X_i | Z) \geq \mu\) for every \(1 \leq i \leq t\), and \(X_1, \ldots, X_t\) are distinct. Then

\[
\Delta((K, Z, h(K, X)), (K, Z, U)) \leq \frac{5t}{4} |R|^{t/3} \cdot 2^{-\mu/3},
\]

for every \(k \in \mathbb{N}\), \(K \leftarrow \text{HKg}(1^k)\) and \(U \leftarrow \text{R}^t\).

For convenience, let \(\overline{m}_1 = m_1[L_1 \setminus L_j]\). From the Lemma 54,

\[
\sum_{j=1}^{\ell+1} \Delta((h(K, m_1[L_j]), \overline{m}_1, \text{pars}, \text{y}), (U[L_j], \overline{m}_1, \text{pars}, \text{y})) \leq \sum_{j=1}^{\ell+1} \frac{5|L_j|}{4} 2^w \leq \frac{5v}{4} 2^w,
\]

where \(w = \frac{1}{3}(\mu - (v - d)(\text{LE}_i \cdot \text{rl} + \text{LT} \cdot \text{il} - \tau))\).

Next, game \(G_5\) is identical to game \(G_4\), except for unopened messages, we are returning a fresh random values as ciphertexts. Note that we are using perfect lossy
encryption and the coins for encrypting unopened messages are completely random. 

Thus

\[ \Pr[G_4(\cdot) \Rightarrow 1] = \Pr[G_5(\cdot) \Rightarrow 1] \]

Finally, in game \( G_5 \), from the joint views of \( A.f \) and \( A.g \), \( m_0 \) is identically distributed as \( m_1 \). Hence \( \Pr[G_5(\cdot) \Rightarrow 1] = 1/2 \). Summing up,

\[ \text{Adv}^{\text{d-so-cpa}}_{\text{DE}[\text{LE},\text{LT},X],A,M,\cdot} \leq 2 \cdot \text{Adv}^{\text{le}}_{\text{LE},B,\cdot} + 2 \cdot \text{Adv}^{\text{ltdf}}_{\text{LT},D,\cdot} + \frac{5v}{2^r} \]

where \( r = \frac{1}{3}(\mu - (v - d)(\text{LT}.il + \text{LE}.rl - \tau)) - u + 1 \).

**Proof of Lemma 54.** For completeness, we now give a proof for Lemma 54. Fix \( \delta > 0 \) and for each \( i \leq t \), let \( S_i \) be the sets of values \( z \) such that \( \tilde{H}_\infty(X_i \mid Z = z) \geq \mu - \log(\delta) \); we shall determine the value of \( \delta \) later. From property (b) of Lemma 53, \( \Pr[Z \in S_i] \geq 1 - \delta \). Let \( S = S_1 \cap \cdots \cap S_t \). From the union bound, \( \Pr[Z \in S] \geq 1 - t\delta \).

Now, let \( \Delta_z \) be the statistical distance, with respect to probability measure \( \Pr[\cdot \mid Z = z] \). Note that for any random variables \( A \) and \( B \) on a common finite domain \( D \),

\[ \Delta(A, B) = \frac{1}{2} \sum_{x \in D} \left| \Pr[A = x] - \Pr[B = x] \right| \]

\[ = \frac{1}{2} \sum_{x \in D} \sum_{z} \Pr[Z = z] \cdot \left| \Pr[A = x \mid Z = z] - \Pr[B = x \mid Z = z] \right| \]

\[ = \sum_{z} \Pr[Z = z] \cdot \Delta_z(A, B) . \]

For convenience, let \( A = h(K, X) \). Hence

\[ \Delta((K, Z, A), (K, Z, U)) = \sum_{z} \Pr[Z = z] \cdot \Delta_z((K, Z, A), (K, Z, U)) \]

\[ = \sum_{z} \Pr[Z = z] \cdot \Delta_z((K, A), (K, U)) \]

\[ \leq \Pr[Z \notin S] + \sum_{z \in S} \Pr[Z = z] \cdot \Delta_z((K, A), (K, U))(B.1) \]

To bound \( \Delta_z((K, h(K, X)), (K, U)) \), we shall use the following result of [58].
Lemma 55 (Crooked Leftover Hash Lemma) [58, Lemma 3.10] Let \( H = (HKg, h) \) be a \( 2t \)-wise independent hash function with domain \( D \) and range \( R \).

Let \( X = (X_1, \ldots, X_t) \) where \( X_i \) is a random variable over \( D \) such that \( H_\infty(X_i) \geq \nu \) for every \( 1 \leq i \leq t \), and \( X_1, \ldots, X_t \) are distinct. Then

\[
\Delta((K, h(K, X)), (K, U)) \leq \frac{t}{2} \sqrt{|R|^t/2^{2\nu}},
\]

for every \( k \in \mathbb{N} \), \( K \leftarrow HKg(1^k) \), and \( U \leftarrow R'. \)

Note that Lemma 55 holds for any probability measure, including \( \Pr[\cdot \mid Z = z] \). Moreover, the min-entropy of \( X_i \) with respect to probability measure \( \Pr[\cdot \mid Z = z] \) is exactly the conditional min-entropy of \( X_i \) given \( Z = z \), with respect to probability measure \( \Pr[\cdot] \). Hence for any \( z \in S \), by applying Lemma 55 for \( \nu = \mu - \log(1/\delta) \) with probability measure \( \Pr[\cdot \mid Z = z] \), it holds that

\[
\Delta_z((K, h(K, X)), (K, U)) \leq \frac{t}{2} \sqrt{|R|^t/(\delta \cdot 2^\mu)} . \tag{B.2}
\]

From Equation (B.1) and Equation (B.2),

\[
\Delta((K, Z, h(K, X)), (K, Z, U)) \leq t\delta + \frac{t}{2} \sqrt{|R|^t/(\delta \cdot 2^\mu)} .
\]

By substituting \( \delta = \frac{1}{4}|R|^{t/3} \cdot 2^{-\mu/3} \) we obtain

\[
\Delta((K, Z, h(K, X)), (K, Z, U)) \leq \frac{5t}{4}|R|^{t/3} \cdot 2^{-\mu/3} .
\]

B.2 Proof of Theorem 41

In this proof, we adopt the following notion. If \( f \) is a function on domain \( S \) and \( Y \) is a vector \( (Y[1], \ldots, Y[r]) \) whose components are elements of \( S \) then \( f(Y) \) is the vector \( (f(Y[1]), \ldots, f(Y[r])) \). Let \( \mathsf{Rmp} \) be an efficient resampling algorithm of \( \mathcal{M} \).

We again use the games \( G_0 - G_5 \) in Figures B.1 and B.2. Then

\[
\text{Adv}_{\text{d-so-cpa}}^{d, \mathsf{DE}[\mathsf{LE}, \mathcal{H}]}(\cdot) = 2 \Pr[G_0(\cdot) \Rightarrow 1] - 1 .
\]
We now explain the game chain. Game $G_1$ is identical to game $G_0$, except that instead of generating an injective key for the lossy encryption, we generate a lossy one. Consider the following adversary $B$ attacking the key indistinguishability of $\text{LE}$. It simulates game $G_0$, but uses its given key instead of generating a new one. It outputs 1 if the simulated game returns 1, and outputs 0 otherwise. Then

$$\Pr[G_0(\cdot) \Rightarrow 1] - \Pr[G_1(\cdot) \Rightarrow 1] \leq \text{Adv}_{\text{LE}, B}^\text{le}(\cdot).$$

Next, game $G_2$ is identical to game $G_1$, except that instead of generating an injective key for the lossy trapdoor function, we generate a lossy one. Consider the following adversary $D$ attacking the key indistinguishability of $\text{LT}$. It simulates game $G_1$, but uses its given key instead of generating a new one. It outputs 1 if the simulated game returns 1, and outputs 0 otherwise. Then

$$\Pr[G_1(\cdot) \Rightarrow 1] - \Pr[G_2(\cdot) \Rightarrow 1] \leq \text{Adv}_{\text{LT}, D}^{\text{tdf}}(\cdot).$$

Next, in game $G_3$, instead of using the set $I$ generated by the adversary, we try to guess it by picking a random subset $I^*$ of $\{1, \ldots, v(k)\}$. If our guess is incorrect, meaning $I \neq I^*$ then we output a random bit $s \leftarrow \{0,1\}$. Then

$$(\Pr[G_2(\cdot) \Rightarrow 1] - 1/2) \leq 2^n \cdot (\Pr[G_3(\cdot) \Rightarrow 1] - 1/2).$$

Next, in game $G_4$, when we encrypt $y[i]$, with $i \in \{1, \ldots, v(k)\}\setminus I^*$, we use fresh random coins instead of $h(K, m_1[i])$. To account for the gap between game $G_3$ and $G_4$, we shall use the following result of [50].

**Lemma 56 (Generalized Leftover Hash Lemma) [50]** Let $H = (\text{HKg}, h)$ be pairwise independent hash function with range $\{0,1\}^\ell$. Then for any random variables $X$ and $Y$ and any $k \in \mathbb{N}$, we have $\Delta((h_K(X), Y, K), (U, Y, K)) \leq \frac{1}{2} \sqrt{2^{\ell - H_\infty(X|Y)}}$, where $K \leftarrow \text{HKg}(1^k)$ and $U \leftarrow \{0,1\}^\ell$ are generated independent of $X$ and $Y$.  

178
For each $i \in \{1, \ldots, v(k)\}$, let $U[i]$ be a fresh random string uniformly distributed in $\text{Coins}(k)$. Let $J = \{1, \ldots, v(k)\} \setminus I^*$. Let $\text{pars} = (K, I^*, ek', \text{param})$. For each fixed choice of $J$,

$$
\Pr[G_3 \Rightarrow 1] - \Pr[G_4 \Rightarrow 1] \\
\leq \Delta((h(K, m_1[J]), m_1[I^*], y[J], \text{pars}), (U[J], m_1[I^*], y[J], \text{pars})) \tag{B.3}
$$

For each $i \in \{1, \ldots, v(k)\}$, let $W_i = \{1, \ldots, v(k)\} \setminus \{i\}$. Since $M$ is ($\mu, \infty$)-entropic, we have

$$
\tilde{H}_\infty(m_1[i] \mid m_1[W_i], \text{pars}) \geq \mu. 
$$

From property (a) of Lemma 53 (stated in the proof of Theorem 40), since $y[i]$ has at most $2^{LT.\alpha-\tau}$ values, it follows that $\tilde{H}_\infty(m_1[i] \mid m_1[W_i], y[i], \text{pars}) \geq \mu - LT.\alpha + \tau$. Let $J = \{u_1, \ldots, u_\ell\}$, with $\ell \leq v(k)$. For each $j \in \{0, \ldots, \ell\}$, let $X_j$ be the vector of $\ell$ components such that $X_j[i] = U[i]$ for every $i \leq \ell$, and $X_j[i] = h(K, m_1[u_i])$ for every $i > \ell$. Then $h(K, m_1[J]) = X_0$ and $U = X_\ell$. From the triangle inequality, the right-hand side of Equation (B.3) is at most

$$
\sum_{i=1}^{\ell} \Delta((X_{i-1}, m_1[I^*], y[J], \text{pars}), (X_i, m_1[I^*], y[J], \text{pars})) \\
\leq \sum_{j \in J} \Delta((h(K, m[j]), m[W_j], y[j], \text{pars}), (U[j], m[W_j], y[j], \text{pars})) \\
\leq v \cdot \frac{1}{2\sqrt{2^{LE.\eta-(\mu+\tau-2LT.\alpha)}}},
$$

the second last inequality is due to the fact that each $y[i]$ and $h(K, m_1[i])$ are completely determined from $m_1[i]$ and $\text{pars}$; and the last inequality is due to the Generalized Leftover Hash Lemma. Finally, in game $G_5$, when we encrypt messages $m_1[i]$, with $i \in \{1, \ldots, v(k)\} \setminus I^*$, we generate truly random ciphertexts. Since $\text{LE}$ is perfectly lossy,

$$
\Pr[G_4(\cdot) \Rightarrow 1] = \Pr[G_5(\cdot) \Rightarrow 1]. 
$$
On the other hand, in game $G_5$, whatever adversaries $A_{\text{cor}}$ and $A_{\text{g}}$ receive are $m[l]$ and random strings independent of the message vector $m_1$. Thus $\Pr[G_5(\cdot) \Rightarrow 1] = 1/2$. Summing up,

$$\text{Adv}_{\text{DE}, A, M}(\cdot) \leq 2 \cdot \text{Adv}_{\text{LE}, B}(\cdot) + 2 \text{Adv}_{\text{LT}, D}(\cdot) + \frac{v}{2^r},$$

where $r = \frac{1}{2} (\mu + \tau - \text{LT}.\text{il} - \text{LE}.r) - v$.

\section*{B.3 Proof of Theorem 42}

In this proof, we adopt the following notion. If $f$ is a function on domain $S$ and $Y$ is a vector $(Y[1], \ldots, Y[r])$ whose components are elements of $S$ then $f(Y)$ is the vector $(f(Y[1]), \ldots, f(Y[r]))$. Let $\text{Rsm}$ be an efficient resampling algorithm of $M$. We use the games $G_0 - G_9$ in Figure B.3, B.4, B.5, and B.6. Then

$$\text{Adv}_{\text{DE}, A, M}(\cdot) = 2 \Pr[G_0(\cdot) \Rightarrow 1] - 1.$$ 

We now explain the game chain. Game $G_1$ is identical to game $G_0$, except that instead of generating an injective key for the all-but-$N$ lossy trapdoor function, we generate a lossy one with respect to tag $y_1$, where $y_1$ is the vector of images of the messages in $m_1$ under $\text{LT}$. Consider the following adversary $D_1$ attacking the key indistinguishability of $\text{NLT}$. It simulates game $G_0$, but uses its given key instead of generating a new one. It outputs 1 if the simulated game returns 1, and outputs 0 otherwise. Then

$$\Pr[G_0(\cdot) \Rightarrow 1] - \Pr[G_1(\cdot) \Rightarrow 1] \leq \text{Adv}_{\text{NLT}, D_1}(\cdot).$$

Next, game $G_2$ is identical to game $G_1$, except that in the decryption oracle, if the adversary can query $c = (c_1, c_2)$ such that the corresponding plaintext $(y_1, y_2)$ of $c_1$ satisfies $\text{TCR}(K_1, y_1) \in \text{TCR}(K_1, y_1)$ then we return $\bot$. Consider
Game $G_0$

\begin{align*}
& b \leftarrow \{0, 1\}; \ param \leftarrow A.pg(1^k) \\
& m_1 \leftarrow \mathcal{M}(1^k, \ param) \\
& K \leftarrow \text{HKg}(1^k) \\
& K_1 \leftarrow \text{TCRKg}_1(1^k) \\
& K_2 \leftarrow \text{TCRKg}_2(1^k) \\
& (ek_1, td_1) \leftarrow \text{IKg}(1^k) \\
& (ek_2, td_2) \leftarrow \text{NLT.Kg}(1^k, \perp) \\
& (pk_1, sk_1) \leftarrow \text{LE.IKg}(1^k) \\
& (pk_2, sk_2) \leftarrow \text{NLE.IKg}(1^k, \perp) \\
& pk \leftarrow (pk_1, pk_2, ek_1, ek_2, K, K_1, K_2) \\
& sk \leftarrow (sk_1, sk_2, td_1, td_2) \\
& y_1 \leftarrow \text{Eval}(ek_1, m_1) \\
& h_1 \leftarrow \text{tc}_{r_1}(K_1, y_1) \\
& y_2 \leftarrow \text{NLT.Eval}(ek_2, h_1, m_1) \\
& (r_1, r_2) \leftarrow h(K, m_1) \\
& c_1 \leftarrow \text{LE.Enc}(pk, (y_1, y_2); r_1) \\
& h_2 \leftarrow \text{tc}_{r_2}(K_2, c_1) \\
& c_2 \leftarrow \text{NLE.Enc}(pk, h_2, (y_1, y_2); r_2) \\
& c \leftarrow (c_1, c_2) \\
& (state, I) \leftarrow \text{A.cor}^{\text{DEC}}(pk, c, \ param) \\
& m_0 \leftarrow \text{Rsmpl}(m_1, I, \ param) \\
& \omega \leftarrow \text{A.g}^{\text{DEC}}(state, m_1[I]) \\
& t \leftarrow \text{A.f}(m_0, \ param) \\
& \text{If } (t = \omega) \text{ then return } b \\
& \text{Else return } 1 - b
\end{align*}

Procedure $\text{Dec}(c)$ $G_0$

\begin{align*}
& \text{If } c \in c \text{ then return } \perp \\
& \text{Else} \\
& \quad (c_1, c_2) \leftarrow c \\
& \quad (y_1, y_2) \leftarrow \text{LE.Dec}(sk_1, c_1) \\
& \quad m \leftarrow \text{lnv}(td_1, y_1) \\
& \text{If } \text{DE.Enc}(pk, m) = (c_1, c_2) \\
& \quad \text{Return } m \\
& \text{Else return } \perp
\end{align*}

Game $G_1, G_2$

\begin{align*}
& b \leftarrow \{0, 1\}; \ param \leftarrow A.pg(1^k) \\
& m_1 \leftarrow \mathcal{M}(1^k, \ param) \\
& K \leftarrow \text{HKg}(1^k) \\
& K_1 \leftarrow \text{TCRKg}_1(1^k) \\
& K_2 \leftarrow \text{TCRKg}_2(1^k) \\
& (ek_1, td_1) \leftarrow \text{IKg}(1^k) \\
& y_1 \leftarrow \text{Eval}(ek_1, m_1) \\
& (ek_2, td_2) \leftarrow \text{NLT.Kg}(1^k, y_1) \\
& (pk_1, sk_1) \leftarrow \text{LE.IKg}(1^k) \\
& (pk_2, sk_2) \leftarrow \text{NLE.IKg}(1^k) \\
& pk \leftarrow (pk_1, pk_2, ek_1, ek_2, K, K_1, K_2) \\
& sk \leftarrow (sk_1, sk_2, td_1, td_2) \\
& y_1 \leftarrow \text{Eval}(ek_1, m_1) \\
& y_2 \leftarrow \text{NLT.Eval}(ek_2, h_1, m_1) \\
& (r_1, r_2) \leftarrow h(K, m_1) \\
& c_1 \leftarrow \text{LE.Enc}(pk, (y_1, y_2); r_1) \\
& h_2 \leftarrow \text{tc}_{r_2}(K_2, c_1) \\
& c_2 \leftarrow \text{NLE.Enc}(pk, h_2, (y_1, y_2); r_2) \\
& c \leftarrow (c_1, c_2) \\
& (state, I) \leftarrow \text{A.cor}^{\text{DEC}}(pk, c, \ param) \\
& m_0 \leftarrow \text{Rsmpl}(m_1, I, \ param) \\
& \omega \leftarrow \text{A.g}^{\text{DEC}}(state, m_1[I]) \\
& t \leftarrow \text{A.f}(m_0, \ param) \\
& \text{If } (t = \omega) \text{ then return } b \\
& \text{Else return } 1 - b
\end{align*}

Procedure $\text{Dec}(c)$ $G_1, G_2$

\begin{align*}
& \text{If } c \in c \text{ then return } \perp \\
& \text{Else} \\
& \quad (c_1, c_2) \leftarrow c; (y_1, y_2) \leftarrow \text{LE.Dec}(sk_1, c_1) \\
& \quad h_1^* \leftarrow \text{tc}_{r_1}(K_1, y_1) \\
& \text{If } h_1^* \in h_1 \text{ then } \text{bad } \leftarrow \text{true}; \text{ return } \perp \\
& \quad m \leftarrow \text{lnv}(td_1, y_1) \\
& \text{If } \text{DE.Enc}(pk, m) = (c_1, c_2) \\
& \quad \text{Return } m \\
& \text{Else return } \perp
\end{align*}

Figure B.3: Games $G_0$–$G_2$ of the proof of Theorem 42.
Game $G_3$

$b \leftarrow \{0, 1\}$; $\param \leftarrow A.pg(1^k)$
$m_1 \leftarrow \mathcal{M}(1^k, \param)$
$K \leftarrow \text{HKg}(1^k)$
$K_1 \leftarrow \text{TCR Kg}_1(1^k)$
$K_2 \leftarrow \text{TCR Kg}_2(1^k)$
$ek'_1 \leftarrow \text{L Kg}(1^k)$
$y_1 \leftarrow \text{Eval}(ek'_1, m_1)$
$(ek'_2, td'_2) \leftarrow \text{NLT Kg}(1^k, y_1)$
$(pk_1, sk_1) \leftarrow \text{LE Kg}(1^k)$
$(pk_2, sk_2) \leftarrow \text{NLE Kg}(1^k, \bot)$
$pk \leftarrow (pk_1, pk_2, ek'_1, ek'_2, K_1, K_2)$
$sk \leftarrow (sk_1, sk_2, td'_2)$
$h_1 \leftarrow \text{tcr}_1(K_1, y_1)$
$y_2 \leftarrow \text{NLT Eval}(ek'_2, h_1, m_1)$
$(r_1, r_2) \leftarrow h(K, m_1)$
$c_1 \leftarrow \text{LE Enc}(pk, (y_1, y_2); r_1)$
$c_2 \leftarrow \text{NLE Enc}(pk, h_2, (y_1, y_2); r_2)$
$c \leftarrow (c_1, c_2)$
$(\text{state}, I) \leftarrow A.cor^{\text{Dec}}(pk, c, \param)$
$m_0 \leftarrow \text{Rsm}(m_1, I, \param)$
$\omega \leftarrow A.g^{\text{Dec}}(\text{state}, m_1[I])$
$t \leftarrow A.f(m_0, \param)$
If $(t = \omega)$ then return $b$
Else return $1 - b$

Procedure $\text{Dec}(c)$ $G_3$

If $c \in c$ then return $\bot$
Else

$(c_1, c_2) \leftarrow c$
$(y_1, y_2) \leftarrow \text{LE Dec}(sk_1, c_1)$
$h'_1 \leftarrow \text{tcr}_1(K_1, y_1)$
If $h'_1 \in h_1$ then return $\bot$
$m \leftarrow \text{NLT Inv}(td'_2, h'_1, y_2)$
If $\text{DE Enc}(pk, m) = (c_1, c_2)$
Return $m$
Else return $\bot$

Game $G_4$, $G_5$

$b \leftarrow \{0, 1\}$; $\param \leftarrow A.pg(1^k)$
$m_1 \leftarrow \mathcal{M}(1^k, \param)$
$K \leftarrow \text{HKg}(1^k)$
$K_1 \leftarrow \text{TCR Kg}_1(1^k)$
$K_2 \leftarrow \text{TCR Kg}_2(1^k)$
$ek'_1 \leftarrow \text{L Kg}(1^k)$
$y_1 \leftarrow \text{Eval}(ek'_1, m_1)$
$(ek'_2, td'_2) \leftarrow \text{NLT Kg}(1^k, y_1)$
$(pk_1, sk_1) \leftarrow \text{LE Kg}(1^k)$
$h_1 \leftarrow \text{tcr}_1(K_1, y_1)$
$y_2 \leftarrow \text{NLT Eval}(ek'_2, h_1, m_1)$
$(r_1, r_2) \leftarrow h(K, m_1)$
$c_1 \leftarrow \text{LE Enc}(pk, (y_1, y_2); r_1)$
$c_2 \leftarrow \text{NLE Enc}(pk, h_2, (y_1, y_2); r_2)$
$c \leftarrow (c_1, c_2)$
$(\text{state}, I) \leftarrow A.cor^{\text{Dec}}(pk, c, \param)$
$m_0 \leftarrow \text{Rsm}(m_1, I, \param)$
$\omega \leftarrow A.g^{\text{Dec}}(\text{state}, m_1[I])$
$t \leftarrow A.f(m_0, \param)$
If $(t = \omega)$ then return $b$
Else return $1 - b$

Procedure $\text{Dec}(c)$ $G_4$, $G_5$

If $c \in c$ then return $\bot$
Else

$(c_1, c_2) \leftarrow c$
$h'_2 \leftarrow \text{tcr}_2(K_2, c_1)$
If $h'_2 \in h_2$ then $\text{bad} \leftarrow \text{true}$; return $\bot$
$(y_1, y_2) \leftarrow \text{LE Dec}(sk_1, c_1)$
$h'_1 \leftarrow \text{tcr}_1(K_1, y_1)$
If $h'_1 \in h_1$ then return $\bot$
$m \leftarrow \text{NLT Inv}(td'_2, h'_1, y_2)$
If $\text{DE Enc}(pk, m) = (c_1, c_2)$
Return $m$
Else return $\bot$

Figure B.4: Games $G_3$–$G_5$ of the proof of Theorem 42.
Table B.5: Games $G_6$, $G_7$ of the proof of Theorem 42.
the following adversary $B = (B_1, B_2)$ attacking TCR$_1$. Adversary $B_1$ first runs $\text{param} \leftarrow A.\text{pg}(1^k); \text{m}_1 \leftarrow \mathcal{M}(1^k, \text{param})$ and then generates a lossy key $ek'$ of LT. It then computes the images $y$ of $\text{m}$ under $\text{Eval}(ek', \cdot)$. Finally, it picks $i \leftarrow \{1, \ldots, v(k)\}$, and outputs $(state, m)$, where $m \leftarrow y[i]$ and $state \leftarrow (\text{param}, \text{m})$.
Adversary \( B_2 \) simulates game \( G_1 \), but uses the provided \((\text{param}, \mathbf{m}, \mathbf{ek}')\) instead of generating new ones. It also uses the given key \( K \) as the key \( K_1 \) for \( \text{TCR}_1 \) instead of generating a new one. If \( \mathcal{A}, \mathcal{g} \) can query some ciphertext \( c = (c_1, c_2) \) to the decryption oracle such that the corresponding plaintext \((y_1, y_2)\) of \( c_1 \) satisfies \( \text{TCR}(K, m) = \text{TCR}(K, y_1) \) then \( B_2 \) halts, outputting \( y_1 \). Then

\[
\Pr[G_1(\cdot) \Rightarrow 1] - \Pr[G_2(\cdot) \Rightarrow 1] \leq \Pr[G_1(\cdot) \text{ sets good}] \leq v \cdot \text{Adv}^\text{lusr}_{\text{TCR}_1, B}(\cdot). 
\]

Next, game \( G_3 \) is identical to game \( G_2 \), except that instead of generating an injective key for the lossy trapdoor function, we generate a lossy one. Consider the following adversary \( D_2 \) attacking the key indistinguishability of \( \text{LT} \). It simulates game \( G_2 \), but uses its given key instead of generating a new one. It outputs 1 if the simulated game returns 1, and outputs 0 otherwise. Then

\[
\Pr[G_2(\cdot) \Rightarrow 1] - \Pr[G_3(\cdot) \Rightarrow 1] \leq \text{Adv}^\text{ldf}_{\text{LT}, D_2}(\cdot). 
\]

Game \( G_4 \) is identical to game \( G_3 \), except that instead of generating an injective key for the all-but-\( N \) lossy encryption, we generate a lossy one with respect to tag \( \mathbf{y}_2 \), where \( \mathbf{y}_2 \) is the vector of ciphertexts of \( \mathbf{m}_1 \) under \( \text{NLT} \) for tags \( \mathbf{h}_1 \leftarrow \text{TCR}_1(K_1, \mathbf{y}_1) \). Consider the following adversary \( D_3 \) attacking the key indistinguishability of \( \text{NLE} \). It simulates game \( G_3 \), but uses its given key instead of generating a new one. It outputs 1 if the simulated game returns 1, and outputs 0 otherwise. Then

\[
\Pr[G_3(\cdot) \Rightarrow 1] - \Pr[G_4(\cdot) \Rightarrow 1] \leq \text{Adv}^\text{abn-le}_{\text{NLE}, D_3}(\cdot). 
\]

Next, game \( G_5 \) is identical to game \( G_4 \), except the following: If \( h^*_2 \leftarrow \text{tcr}_2(K_2, \mathbf{y}_2) \) is in the vector \( \mathbf{h}_2 \) then we return \( \perp \). Consider the following adversary \( B' = (B'_1, B'_2) \) attacking \( \text{TCR}_2 \). Adversary \( B'_1 \) follows game \( G_4 \) to generate vector \( \mathbf{y}_2 \) and picks \( i \leftarrow s \{1, \ldots, v(k)\} \). It then outputs \((\text{state}, m)\), where \( m \leftarrow \mathbf{y}_2[i] \), and \( \text{state} \) consists
of $m$ and the coins that $B'_1$ uses. Adversary $B'_2$ simulates game $G_4$, but uses the coins of $B'_1$ on the randomized algorithms that $B'_1$ runs. Moreover, it uses the given key as $K_2$ instead of generating a new one. If $A$ can query some ciphertext $c = (c_1, c_2)$ to the decryption oracle such that $\text{TCR}(K, c_1) = \text{TCR}(K, m)$ then $B'_2$ halts, outputting $c_1$. Thus,

$$\Pr[G_4(\cdot) \Rightarrow 1] - \Pr[G_5(\cdot) \Rightarrow 1] \leq \Pr[G_4(\cdot) \text{ sets bad}] \leq v \cdot \text{Adv}^{\text{TCR}}_{T_{KR_2}, B'_{B'}}(\cdot).$$

Next, game $G_6$ is identical to game $G_5$, except that instead of generating an injective key for the lossy encryption, we generate a lossy one. Consider the following adversary $D_4$ attacking the key indistinguishability of $\text{LE}$. It simulates game $G_5$, but uses its given key instead of generating a new one. It outputs 1 if the simulated game returns 1, and outputs 0 otherwise. Then

$$\Pr[G_5(\cdot) \Rightarrow 1] - \Pr[G_6(\cdot) \Rightarrow 1] \leq \text{Adv}_{\text{LE}, D_4}^{\text{LE}}(\cdot).$$

Next, in game $G_7$, instead of using the set $I$ generated by the adversary, we try to guess it by picking a random subset $I^*$ of $\{1, \ldots, v(k)\}$ with size at most $d$. If our guess is incorrect, meaning $I \neq I^*$ then we output a random bit $s \leftarrow \{0, 1\}$. Note $\Pr[I = I^*] \geq 2^{-u}$, where $u = \min\{1 + d \log v, v\}$. Then

$$(\Pr[G_6(\cdot) \Rightarrow 1] - 1/2) \leq 2^u \cdot (\Pr[G_7(\cdot) \Rightarrow 1] - 1/2).$$

Next, in game $G_8$, when we encrypt $y_1[i], y_2[i]$, with $i \in \{1, \ldots, v(k)\} \setminus I^*$, we use fresh random coins instead of $(r_1[i], r_2[i]) \leftarrow h(K, m[i])$. To account for the gap between game $G_7$ and $G_8$, we shall use the same argument we had in the proof of Theorem 40. Then

$$\Pr[G_7(\cdot) \Rightarrow 1] - \Pr[G_8(\cdot) \Rightarrow 1] \leq \frac{5v}{4 \cdot 2^w},$$

where $w = \frac{1}{3}(\mu - (v - d)(\text{LT}.il + \text{NLT}.il + \text{LE}.rl + \text{NLE}.rl - \tau_1 - \tau_2)).$
Finally, in game $G_9$, when we encrypt messages $m_1[i]$, with $i \in \{1, \ldots, v(k)\}\setminus I^*$, we generate truly random ciphertexts. Since $\text{NLE}$ and $\text{LE}$ are perfectly lossy,

$$\Pr[G_8(\cdot) \Rightarrow 1] = \Pr[G_9(\cdot) \Rightarrow 1].$$

On the other hand, in game $G_9$, whatever adversaries $A_{\text{cor}}$ and $A_{\text{g}}$ receive are $m[I]$ and random strings independent of the message vector $m_1$. Thus $\Pr[G_9(\cdot) \Rightarrow 1] = 1/2$. Summing up,

$$\text{Adv}_{\text{DE}, A, M}(\cdot) \leq 2 \cdot \text{Adv}^{\text{abn-ltdf}}_{\text{NLT}, D_1}(\cdot) + 2 \cdot \text{Adv}^{\text{ltdf}}_{\text{LT}, D_2}(\cdot) + 2 \cdot \text{Adv}^{\text{abn-le}}_{\text{NLE}, D_3}(\cdot)$$

$$+ 2 \cdot \text{Adv}^{\text{le}}_{\text{LE}, D_4}(\cdot) + 2v \cdot \text{Adv}^{\text{tcr}}_{\text{TCR}_1, B}(\cdot) + 2v \cdot \text{Adv}^{\text{tcr}}_{\text{TCR}_2, B}(\cdot) + \frac{5v}{2^v},$$

where $r = \frac{1}{3}(\mu - (v - d)(\text{LT}.il + \text{NLT}.il + \text{LE}.rl + \text{NLE}.rl - \tau_1 - \tau_2)) - u + 1$. 

187
Bibliography


188


[48] Dana Dachman-Soled, Georg Fuchsbauer, Payman Mohassel, and Adam O’Neill. Enhanced chosen-ciphertext security and applications. In Hugo Krawczyk,


198


[73] Eike Kiltz and Krzysztof Pietrzak. On the security of padding-based encryption schemes - or - why we cannot prove OAEP secure in the standard model. In


