Encryption in the Presence of Key-Dependent Messages and Related-Key Attacks

Gareth Thomas Davies

A dissertation submitted to the

University of Bristol

in accordance with the requirements

for award of the degree of

Doctor of Philosophy

in the

Faculty of Engineering

September 2015

c. 27,500 words

Abstract

This thesis investigates enhanced adversarial models for encryption, for scenarios where established notions of security are not sufficient to accurately model the capabilities of real-world adversaries. In particular we focus on achieving encryption schemes that are secure even when the adversary has more power than granted by standard notions such as indistinguishability under chosen-plaintext attack (IND-CPA). These extended models allow researchers, implementors and end-users to confidently pinpoint areas of strength and weakness in cryptographic hardware and software.

Our first contribution considers key-dependent message (KDM) security, meaning security even when an adversary has access to encryptions of the decryption key. Our focus is the hybrid encryption framework, a method for public key encryption that is widely deployed. We give sufficient conditions for achieving KDM security for this paradigm in the random oracle model using novel proof techniques, and cast known impossibility results in KDM security in the context of hybrid encryption.

Next we investigate modelling an adversary that is yet more powerful: related-key-attack-and-key-dependent message (RKA-KDM) security considers when an adversary has to access encryptions, performed under related keys, of key-dependent messages. Our main result is a composition theorem showing how to generically achieve RKA-KDM security. To indicate the efficacy of our approach, we present a number of symmetric key instantiations that use known KDM-secure public key encryption schemes as a starting point.

Acknowledgements

I would like to thank Martijn Stam and Bogdan Warinschi for their committed supervision and support throughout my studies. Their enthusiasm for challenging problems, encouragement and attention to detail have made me a stronger researcher and a better person.

A major part of the work in this thesis was completed in collaboration with Dennis Hofheinz and Florian Böhl; I extend my gratitude to Dennis for hosting me in Karlsruhe in summer 2013 and the stimulating and productive discussions the three of us had during and after my visit. I'm also thankful to Georg Fuchsbauer and Krzysztof Pietrzak for hosting me at IST in Vienna in summer 2014.

Big thanks to Chris Wuthrich of Nottingham University who first introduced me to cryptography and subsequently encouraged me to pursue postgraduate study, and to EPSRC and the Bristol University Alumni Foundation for providing financial support.

Thanks to every member, past and present, of the Bristol cryptography group that has made my PhD experience so enjoyable. To all those have contributed to the proofreading effort, I thank and salute you.

Pursuing a PhD has been a wonderful experience and certainly would not have been possible without the unwavering support of my family and, of course, Camilla. Thank you.

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

Gareth Thomas Davies
18th September 2015

Contents

Page

List of Figures **vii**

Abbreviations **viii**

Notation **ix**

1 Introduction **1**

1.1 Motivation and Context . . . 1

1.2 Publications . . . 4

1.3 Thesis Outline . . . 5

2 Preliminaries **7**

2.1 Provable Security . . . 8

2.1.1 Game Hopping . . . 9

2.1.2 Cryptographic Adversaries . . . 10

2.1.3 Random Oracle Model . . . 11

2.1.4 Security Models for Encryption . . . 12

2.2 Symmetric Encryption . . . 13

2.2.1 Indistinguishability under Chosen-Plaintext Attack . . . 14

2.3 Public Key Encryption . . . 15

2.3.1 Indistinguishability under Chosen-Ciphertext Attack . . . 16

2.3.2 Plaintext Checking Oracles . . . 18

2.4 Hybrid Encryption . . . 18

2.4.1 Security Models for KEMs and DEMs . . . 22

2.5 Pseudorandom Functions . . . 23

2.6 Assumptions in Cryptography . . . 24

2.6.1 Hard Problems . . . 24

3 Key-Dependent Message Security 28

3.1 Overview and Motivation 28

3.2 Related Literature 30

3.3 Security Models and Formalism 33

3.3.1 IND-KDM-CPA Security of SKE 34

3.3.2 IND-KDM-CCA Security of PKE 35

3.3.3 Circular Security and Alternative Notions 36

3.4 Realising KDM Security 37

3.4.1 Black et al.'s Random Oracle Scheme 37

3.4.2 Boneh et al.'s IND-KDM-CPA-Secure Scheme 38

3.4.3 Camenisch et al.'s IND-KDM-CCA-Secure Scheme 39

3.5 Achieving KDM Security of Hybrid Encryption 39

3.5.1 Tools 40

3.5.2 Achieving KDM-CCA Security of Hybrid Encryption 43

3.6 Separation Results for KDM Security in the Hybrid Framework 58

3.6.1 Writing Acar et al.'s es_2 as a KEM and DEM 59

3.6.2 Writing Cash et al.'s Π_{CPA} as a KEM and DEM 60

3.7 Conclusions 62

4 Related-Key Attack Security 63

4.1 Overview and Motivation 63

4.2 Related Literature 64

4.3 Existing Security Models and Formalism 67

4.3.1 RKA Security for PRFs and PRPs 68

4.3.2 RKA Security for Symmetric Encryption 69

4.4 RKA-KDM Security 70

4.4.1 Security Definition 71

4.4.2 A Generic Construction 72

4.4.3 A Retraction 75

4.4.4 Instantiations 79

4.5 Conclusions 97

Bibliography 99

List of Figures

2.1 IND-CPA game for SKE 14

2.2 IND-CPA security experiment for SKE 15

2.3 IND-CCA game for PKE 16

2.4 IND-CCA security experiment for PKE 17

2.5 Diagram of Hybrid Encryption 19

2.6 Algorithms for a Hybrid Cryptosystem Hyb. 20

2.7 μOW-CCA security experiment for KEMs 22

2.8 IND-CCA security experiment for DEMs 23

2.9 PRF experiment 24

2.10 Factoring, DCR and QR experiments 25

2.11 DLP and DDH experiments 26

3.1 IND-KDM-CPA security experiment for SKE 34

3.2 IND-KDM-CCA security experiment for PKE 35

3.3 Black et al.'s **ver** construction 37

3.4 Boneh et al.'s Π_{BHHO} construction 38

3.5 IND-PKDM-CCA security experiment for DEMs 41

3.6 Hybrid games Hyb-j to prove Thm. 3.1 42

3.7 Reduction $\mathcal{A}_1$ used to prove Eqn. 3.2 43

3.8 IND-CCA-KDM security experiment for Hybrid Encryption 44

3.9 Diagram of game hops used to prove Eqn. 3.2 45

3.10 Description of game hops used to prove Eqn. 3.2. 47

3.11 Security games used for proof Thm. 3.2 49

3.12 Reduction $\mathcal{A}_2$ used to prove Eqn. 3.11. 53

3.13 Reduction $\mathcal{A}_1$ used to prove Eqn. 3.12. 55

3.14 Reduction $\mathcal{A}_3$ used to prove Eqn. 3.13. 56

3.15 Hybrid version of Acar et al.'s es_2 construction 60

3.16 Hybrid version of Cash et al.'s Π construction 61

4.1 RKA-PRP experiment 68

4.2 RKA security experiment for SKE 70

4.3 RKA-KDM security experiment for SKE 71

4.4 RKA-transformer experiment 73

4.5 Security games for Generic Framework Theorem 74

4.6 Reduction $\mathcal{A}_1$ used to prove Eqn. 4.2. 74

4.7 Reduction $\mathcal{A}_2$ used to prove Eqn. 4.3. 75

4.8 Reduction $\mathcal{A}_1$ used to prove Eqn. 4.4. 75

4.9 KDM-transformer experiment 76

4.10 Security games for Retracted Generic Framework Corollary 77

4.11 Reduction $\mathcal{A}_1$ used to prove Eqn. 4.7. 78

4.12 Reduction $\mathcal{A}_2$ used to prove Eqn. 4.8. 78

4.13 Reduction $\mathcal{A}_1$ used in attempted proof of Eqn. 4.9. 79

4.14 Table of RKA-KDM-secure Instantiations 79

4.15 SKE scheme Σ'_{BHHO} 81

4.16 Reduction $\mathcal{A}_1$ used to prove Eqn. 4.12. 84

4.17 Reduction $\mathcal{A}_2$ used to prove Eqn. 4.13. 84

4.18 SKE scheme Σ'_{BG} 86

4.19 SKE scheme Σ_{MTY} 91

4.20 Interactive Vector Game IV_1 92

4.21 Interactive Vector Game IV_2 92

4.22 Reduction $\mathcal{A}_1$ used to prove Eqn. 4.16. 93

4.23 Reduction $\mathcal{A}_2$ used to prove Eqn. 4.17. 94

4.24 SKE scheme Σ'_{ACPS} 95

Abbreviations

AES	Advanced Encryption Standard
CCA	Chosen-Ciphertext Attack
CPA	Chosen-Plaintext Attack
DCR	Decisional Composite Residuosity (assumption)
DDH	Decisional Diffie-Hellman (assumption)
DEM	Data Encapsulation Mechanism
DES	Data Encryption Standard
DLP	Discrete Logarithm Problem
IND	Indistinguishability
KDF	Key Derivation Function
KDM	Key-Dependent Message (security)
KEM	Key Encapsulation Mechanism
LWE	Learning With Errors (assumption)
NIZK	Non-Interactive Zero Knowledge (proof)
OW	One-Wayness
PKDM	Prior Key-Dependent Message (security)
PRF	Pseudorandom Function
QR	Quadratic Residuosity (assumption)
RKA	Related-Key Attack (security)
TDP	Trapdoor Permutation
XOR	Exclusive-Or operation

Notation

$\mathrm{A} := \mathrm{B}$	Notion A is defined by B
$\mathrm{x}\|\|\mathrm{y}$	Concatenation of bitstrings x and y
$\{0,1\}^n$	Bitstrings of length *n* bits
$\{0,1\}^*$	Bitstrings of arbitrary length
$\mathrm{x} \oplus \mathrm{y}$	The bit-wise exclusive-or (XOR) of x and y
$\|\mathrm{x}\|$	The length of bitstring x
$\mathsf{ord}(\mathrm{g})$	The order of a group element g
$\mathrm{x} \overset{\$}{\leftarrow} \mathrm{X}$	x is chosen uniformly at random from set X
$\mathrm{x} \leftarrow \mathcal{A}(\mathrm{y})$	x is output by algorithm $\mathcal{A}$ which has input y
$\mathrm{x} \leftarrow \mathrm{y}$	x is assigned the value y
$\mathrm{x}[1 \ldots \mathrm{y}]$	The first y bits of x
$\perp$	Error symbol
$\mathbb{Z}_\mathrm{N}$	Integers modulo N
$\mathbb{Z}_\mathrm{N}^*$	The group of integers modulo N under multiplication
$\mathsf{Primes}[\mathrm{x}]$	Set of primes of length x
$\mathsf{Blum}[\mathrm{x}]$	Set of Blum integers of length x
$\mathsf{SafePrimes}[\mathrm{x}]$	Set of safe primes of length x
φ_E	Euler's Phi (totient) Function
$\langle \mathbf{x}; \mathbf{y} \rangle$	Inner product of vectors **x** and **y**
$\mathcal{K}$	Keyspace
$\mathcal{M}$	Message Space

CHAPTER 1

Introduction

Contents

1.1 Motivation and Context **1**

1.2 Publications **4**

1.3 Thesis Outline **5**

1.1 Motivation and Context

The recent and rapid computerisation of society has meant that cryptographic components are playing an increasing role in how humans interact with each other and the machines we use for our social lives and our work. These exciting developments bring with them a need for security against a growing number of adversaries of varying power. Over time our understanding of what it means for a cryptographic scheme[1] to be *secure* has changed dramatically. This means that we need to carefully consider what power an adversary against the scheme has, in a way that accurately reflects reality. This thesis will focus on encryption schemes that are secure in two strong adversarial models: the scenario when an adversary has access to encryptions of the private keys of the scheme, and when the adversary can tamper a scheme to use a modified key.

Historical ciphers were often broken because there was no concept of what it meant for them to be 'secure' in the first place: ad hoc techniques that appeared difficult for a few minds to break were deployed and subsequently trivially broken. This idea of confidence in security stemming from absence of attacks was rife until soon after the Second World War when Shannon [180] introduced the one-time pad and information-theoretically secure (aka perfectly secure) cryptography. This represents a fundamental gold standard for symmetric encryption. However for the one-time pad to achieve information-theoretic security each key needs to be as long as each mes-

[1] In this thesis we will use *schemes* to mean both primitives (low-level constructs such as block ciphers, signature algorithms and hash functions) and protocols (systems such as TLS and Kerberos that use primitives).

sage: a system is not usable if it is impractical. This instigated the consideration of a trade-off between security and efficiency, and precision in this context has been sought ever since.

Rigorous analysis of cryptography lay mainly in the purview of government agencies until the 1970s, when Diffie and Hellman [103] and subsequently Rivest, Shamir and Adleman [174] and Rabin [172] introduced novel methods for key exchange, public key encryption and digital signatures that used properties of algebraic groups. This bridge between computer science and mathematics encouraged many mathematicians to start working on open problems in communication that were, until that point, of little relation to theoretical mathematics. The work in this thesis falls under the umbrella of *provable security*, a mathematical framework giving security definitions, and proofs relative to these definitions, for cryptographic primitives. The rigour that is now associated with cryptography as a field has roots in both computer science and mathematics, and in particular the development of cryptographic primitives based on mathematical hardness assumptions has rapidly enhanced our collective understanding of these problems. This idea of practice-oriented provable security has allowed researchers with a preference for theoretical topics to move cryptography from its fundamental roots in pseudorandom functions and information theory to the study of widely-used schemes such as DES and RSA-OAEP, and larger protocols.

Security definitions allow us to assess the strength of a cryptographic scheme, and thoroughly analyse its resilience against adversaries. Definitions strictly define an adversary's behaviour, yielding elegant and precise results stating that schemes are secure with respect to the definition for a class of adversaries. In many scenarios this class accurately reflects the realistic adversaries that exist in the context of particular schemes, giving a clear indicator to implementors and users of what security guarantees they can expect. Developing the 'correct' model is not easy and many factors affect what should be taken into consideration, yet the desire for precision is unquestionable. For many years it appeared that IND-CPA, where a scheme is secure even in the presence of an adversary that can access encryptions under the 'live' key, was regarded as the necessary security level for encryption. The concept of chosen-ciphertext attacks, where the adversary can additionally send elements of the ciphertext space and have them decrypted under the active decryption key, was viewed by many as being too strong: for some implementors this was an unrealistic attack scenario and deployed schemes would simply not be vulnerable to it. In 1998 Bleichenbacher [66] presented the first practical chosen-ciphertext attack, on the widely-deployed SSL key establishment protocol that is based on RSA PKCS #1 v1.5.

The definitions of security in the presence of chosen-plaintext and chosen-ciphertext attacks mentioned above are independent of the way a scheme is implemented; a scheme that is deemed to be extremely secure by theoreticians may be trivially insecure when even a passive adversary has access to side-channel information (such as observations of the power consumption of a device whilst it executes a cryptographic algorithm [145]). The proliferation of mass surveillance has encouraged consideration of a new type of adversary vastly more powerful than those normally considered in the real world and in academic literature. This means that it is vital to regard security proofs as a component in the wider analysis of a system, and with this full analysis implementors can be aware of which adversaries pose the highest threats to the system.

Standard definitions of security for encryption (i.e. IND-CPA and IND-CCA) do not consider how an adversary can physically tamper with the system or the types of queries that an adversary can make, and this thesis will investigate these two facets. In some real-world scenarios it may be possible for an adversary to modify an encryption, for example flipping bits of an AES key. The theoretical study of this type of attack is known as related-key attack (RKA) security: in this model we ask the adversary to submit a related-key deriving function φ from some function class Φ and a message m, and then distinguish encryptions under $\varphi(\mathrm{k})$ of either m or a random element of the message space (for key k)—if no adversary can do this then the encryption scheme is RKA secure with respect to the class Φ. In the case of symmetric encryption, if an adversary can specify constant functions (i.e. removing the key from the encryption algorithm altogether) then it will be able to trivially distinguish in the RKA security game. We wish to develop schemes that are RKA secure with respect to rich and meaningful function classes, to best represent the tampering attacks that can exist in the wild.

Key-Dependent Message (KDM) security approaches from another angle, asking whether a scheme is secure even when an adversary has access to messages that depend on the decryption key. In this framework an adversary submits a function ψ to its encryption oracle and receives either an encryption of $\psi(\mathrm{k})$ or an encryption of a random message of the same length, where k is the decryption key of the system. This is intuitively a concern in hard disk encryption—we might expect that a system will store (a representation of) the key encrypted under the key itself, and the definition is additionally useful in formal security proofs and anonymous credential systems.

This thesis will give two main results: firstly a composition theorem showing how to construct hybrid encryption that is KDM secure, and secondly the means to achieve security when an adversary has *both* RKA and KDM capabilities *at the same time*. The first result states that if the Key Encapsulation Mechanism (KEM)—used to transport a symmetric key—is OW-CCA secure, the Data Encapsulation Mechanism (DEM)—that encrypts a message under the symmetric key chosen by the KEM—is IND-CCA secure and the intermediary key derivation function is modelled as a random oracle, then the resulting hybrid encryption scheme is IND-KDM-CCA secure. We then show if a symmetric encryption scheme is KDM-CPA secure and if there exists an efficient *RKA transformer* that produces encryptions under related keys without knowledge of the underlying key, then the resulting SKE scheme is RKA-KDM secure. To demonstrate the utility of this approach we give a number of instantiations of schemes that are secure in this model under a number of hardness assumptions.

RKA and KDM security are just two of many adversarial models for encryption that model the scenario where standard IND-CPA (or IND-CCA) security is not strong enough: others include selective opening security [107, 41, 33] and leakage resilience [161, 108, 11]. The idea of leakage resilient cryptography is to formally model the amount of information an adversary can acquire using side-channel attacks, however these models are fraught with difficulties and have inherent limitations in achieving generality. There are parallels between the KDM security and leakage resilient cryptography, with both models considering the leakage of secret data as encryptions are performed.

In summary, the modern interconnected world relies on the ubiquity and robustness of cryptography in its many forms. Provable security gives a clear idea of which types of adversary are impotent against a scheme, allowing designers and implementors to focus on preventing other avenues of attack. It is vitally important that the theoretical treatment of the cryptography that is used in practice takes into account as many attack scenarios as possible, and the work in this thesis furthers the understanding of security in the presence of adversaries with strong capabilities.

1.2 Publications

The material in this thesis is primarily based on the following two papers that have appeared at international conferences:

[98] Gareth T. Davies and Martijn Stam: KDM security in the Hybrid Framework. In CT-RSA 2014, LNCS volume 8366, pages 461–480, Springer, Berlin, Germany.

[68] Florian Böhl, Gareth T. Davies, and Dennis Hofheinz: Encryption Schemes Secure under Related-Key and Key-Dependent Message Attacks. In PKC 2014, LNCS volume 8383, pages 483–500, Springer, Berlin, Germany.

1.3 Thesis Outline

Chapter 2 gives some preliminary notions and fixes notation. Chapter 3 shows how to achieve KDM security when using hybrid encryption, with a proof in the random oracle model, and much of the work in this chapter is included in the extended abstract [98] and the full version [97]. Chapter 4 investigates the link between RKA security and KDM security, giving a novel composition theorem and a number of instantiations indicating the utility of a joint definition; this chapter reflects some of the content in [68] and the full version [67], however this thesis additionally contains a number of changes from the published work.

CHAPTER 2

Preliminaries

Contents

2.1 Provable Security 8

2.1.1 Game Hopping 9

2.1.2 Cryptographic Adversaries 10

2.1.3 Random Oracle Model 11

2.1.4 Security Models for Encryption 12

2.2 Symmetric Encryption 13

2.2.1 Indistinguishability under Chosen-Plaintext Attack 14

2.3 Public Key Encryption 15

2.3.1 Indistinguishability under Chosen-Ciphertext Attack 16

2.3.2 Plaintext Checking Oracles 18

2.4 Hybrid Encryption 18

2.4.1 Security Models for KEMs and DEMs 22

2.5 Pseudorandom Functions 23

2.6 Assumptions in Cryptography 24

2.6.1 Hard Problems 24

This chapter will outline notational conventions and present some standard notions of security that will be used later in the thesis. We will first discuss provable security and mechanisms used for encryption, then give a number of mathematical tools and assumptions used later on.

This thesis will look in detail at security models for symmetric encryption, where the communicating parties share a key that is used for both encryption and decryption, and public key encryption, where each user has a key pair with which they can communicate. Throughout this thesis we will write $(\mathsf{Pg}, \mathsf{Kg}, \mathsf{E}, \mathsf{D})$ for the parameter generation, key generation, encryption and decryption algorithms of symmetric key cryptosystems, and $(\mathsf{PGen}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ for the algorithms associated with public key cryptography.

2.1 Provable Security

In this chapter we give some notions of security for some fundamental cryptographic primitives and protocols, and show how to use hardness assumptions. A proof of security gives confidence in the ability of a cryptographic primitive to resist a well-defined class of potential attacks. This means that we need to formally define the primitive in terms of the algorithms and their inputs/outputs, and the adversarial environment in which the class of adversaries operates.

There are two main approaches for security proofs, namely *unconditional/information-theoretic security* which means that a scheme is secure against even a computationally unbounded adversary, and *computational security* where we define security against some class of adversarial attackers with certain resources. The former approach was introduced by Shannon [180] with respect to the one-time pad, and in modern cryptography it is often used in the study of secret sharing [179, 52] and commitment schemes [169, 84]. The latter approach can be roughly divided into two categories: using term algebras to abstract the operations in cryptographic systems then only considering adversaries that respect this term structure (i.e. only perform operations that are meaningful in the term algebra), and proofs using so-called security games to model an adversary's environment. The former approach, known as the symbolic approach, takes cryptographic primitives as black boxes and is often used to analyse relatively complex protocols. Throughout the thesis we will use game-based security definitions and employ reductionist proofs. This means a security property for a particular cryptographic scheme is defined as a game between a challenger and an adversary, and we show security by reducing the task of winning the game, and consequently 'breaking' security, to the task of solving an underlying 'hard' problem.

The general approach is to define what it means for a scheme to be secure, in terms of the possible interactions that can occur between the adversary and the challenger as well as the adversary's winning condition, and to clearly define a class of adversaries. A proof by reduction will then show that, as long as some 'well-studied' problem is indeed hard, the scheme is in fact secure against that class of adversaries. In more detail, we first assume that problem P is hard, and then show that if there exists some efficient algorithm $\mathcal{A}$ that breaks our scheme, then there exists an efficient algorithm $\mathcal{A}_1$ that breaks P. We ask that the algorithm $\mathcal{A}_1$, often called 'the reduction', uses $\mathcal{A}$ in a black-box way and does not assume anything about $\mathcal{A}$'s internal behaviour: this is because we need the previous statement to hold for *every* possible winning algorithm $\mathcal{A}$ in the class. Of course this means that there may exist attacks that are outside of the assumptions we made about the capabilities of the adversary, and whether or not this approach is appropriate is a topic of contention [26, 144].

2.1.1 Game Hopping

The technique of proof by reduction for a scheme is an extremely useful tool, but it may not be possible to create a reduction directly from the security experiment to some hard problem or assumption. It is often necessary to employ the technique of *game hopping*, which means slowly manipulating the adversary's environment until the attack success can be computed or bounded. Along the way we bound the adversary's probability in distinguishing these minor changes to its environment, giving us an overall bound for the success in the security experiment that we started with. We will write $\mathbf{Pr}\left[\mathsf{G}^{\mathcal{A}} = 1\right]$ to represent the probability that the game output is 1 when we run game G with an adversary $\mathcal{A}$.

The concept of a game hopping proof stems from *hybrid arguments* first used by Goldwasser and Micali [120] and Yao [188] and was stated more deliberately by Bellare and Goldwasser [34]. The development of code-based game hopping proofs started with Killian and Rogaway's work analysing DESX [140, 141]—this important step of regarding games as programs instead of abstract environments has been embraced by a large number of authors since. The works of Bellare and Rogaway [51] and Shoup [183] created and explained the formalism necessary for authors to extend game hopping proofs to other primitives.

There are three main types of game hop:

- Transitions that are just a re-formulation of the previous game: as far as the adversary is concerned its interface with the challenger (and its success advantage) is identical. This means that conceptual changes or rephrasing occur, but mathematical changes do not.

- Transitions based on failure events: if the code of games G_i and G_j differs only in statements that follow the setting of a Boolean flag bad to `true`, then we say these two games are *identical until* bad. This utilises the fundamental lemma of game playing [51] (also known as the *difference lemma*):

  Lemma 2.1. *Let* G_i, G_j *be* identical until bad *games, and let* $\mathcal{A}$ *be an adversary. Then*

  $$\left|\mathbf{Pr}\left[\mathsf{G}_i^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G}_j^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Pr}\left[\mathsf{G}_j^{\mathcal{A}} \textit{ sets } \mathsf{bad}\right] .$$

- Transitions based on indistinguishability: this type of hop involves modifying one of the adversary's inputs or interfaces (for example an oracle). Instead of drawing a value from the correct distribution D_1, we instead choose it from some modified distribution D_2 that is computationally indistinguishable from D_1. The idea here is that if the adversary's behaviour changes significantly then we have a way of distinguishing the distributions, so the adversary's attack success may only increase by its advantage in distinguishing D_1 from D_2.

Dent [101] provides a distinction between two types of failure events: those with small probability as above and failure events that are large (but not overwhelming). The syntax used here is inspired by Dent's explanations.

The majority of security definitions in this thesis are indistinguishability-based notions, where an adversary is asked to distinguish either a real execution from a 'random' execution, or provide two inputs and decide which one has been used. Authors face a choice when conducting proofs on indistinguishability-based security definitions: either give the base game as the security experiment where a challenger is yet to select the challenge bit b and manipulate this environment, or start at one of the 'sides' of the security experiment and hop to the other side. We will use both approaches in this thesis, and indicate clearly which approach is being used for each proof.

A convention we use in this thesis is to start counting games from zero ($\mathsf{G}_0, \mathsf{G}_1, \dots$), and keys (in the case they are plural) from one ($\mathsf{k}_1, \mathsf{k}_2, \dots$).

2.1.2 Cryptographic Adversaries

There is much discussion in the literature on how to represent adversaries in cryptography, and a chasm has formed between authors preferring 'concrete security' [49, 43, 35] (meaning adversarial success probability is defined by the number of queries made and is bounded by some small value) and those in the 'PPT security' camp (adversaries are probabilistic polynomial time algorithms where the scheme is secure if success is a negligible function in the security parameter). For this thesis we will follow the concrete security approach.[1] This means that our definitions of primitives are in terms of an advantage function, and we say that a scheme is 'secure' if, for all adversaries with resources that are 'practical', the advantage is 'small', with each term in inverted commas deliberately informal. Any scheme (other than the one-time pad) is breakable if we don't limit the adversary, so it makes sense to define the adversary's advantage as a function of its own resources.

We consider adversaries to be algorithms, and the syntax $\mathbf{Adv}^{\mathsf{notion}}_{\mathsf{scheme},\,\mathcal{A}}$ defines adversary $\mathcal{A}$'s

[1] For the simple reason that the primary PhD supervisor is vehemently against the use of PPT (and footnotes) in the field of cryptography.

advantage in attacking the notion security of scheme. We will often utilise a Boolean variable called *flag*, which is initialised to `false` and once set to `true` it stays `true`. When referring to algorithms, the error symbol $\lightning$ will refer to an adversary attempting to do something which is explicitly forbidden in the security game, and the symbol $\perp$ will indicate where a computation has gone wrong or an incorrect input has been supplied (e.g. invalid ciphertext to a decryption algorithm).

The definitions that we use will try to place as few limits as is possible on the adversary (within reason), so we only regard security in terms of the adversary's resources rather than its strategy.

2.1.3 Random Oracle Model

One of the major simplifying assumptions used in game-playing proofs is the introduction of a so-called *random oracle* [48], an item that models a perfectly random (i.e. idealised) hash function. This means that there exists an oracle, available to all parties, that takes arbitrary-length input values and outputs an element of the output domain chosen uniformly at random. It takes unit time to evaluate and has memory so that if it is ever given the same input twice, it will give the same output to both queries. This means that we simply regard a random oracle as a large input-output table, and in security proofs we use a technique called *lazy sampling* to fill in this table each time a party queries the random oracle: initially the table is empty, and when a party makes a query the oracle first checks if the value has previously been asked. This approach—removing each party's ability to compute hash values themselves—means that reductions can not only monitor and simulate what hash function calls the adversary makes but in some cases also modify these calls (leading to techniques such as programming [164, 113]).

This function has an infinite description yet is a useful tool in security proofs where we wish to gauge security in the presence of a hash function. Schemes that are secure in the random oracle model give no guarantees about what happens when we replace this random oracle with a hash function such as SHA-3 or RIPEMD160, and as such proofs in the random oracle model are regarded with caution and some scepticism [119, 160, 153]. In fact there exist encryption and signature schemes that are secure in the random oracle model yet are trivially insecure when implemented with any hash function [85, 164, 27].

2.1.4 Representing Security Models for Encryption

We now elaborate on the types of security models that we will use and develop later on. We follow the literature and use the convention goal-tools to denote security notions for encryption. The item $\mathsf{goal} \in \{\mathsf{KR}, \mathsf{OW}, \mathsf{IND}\}$ follows a clear hierarchy: KR refers to *key recovery*[2] and is a weaker notion than OW which is *one-wayness* (inversion), which in turn is weaker than IND which means *indistinguishability* or the ability to distinguish ciphertexts. For the purpose of this work we will consider $\mathsf{tools} \in \{\mathsf{CPA}, \mathsf{CCA}\}$, which refers to chosen-plaintext attacks and the stronger notion of chosen-ciphertext attacks respectively. We will write definitions in mathsf mode, e.g. IND-CPA, to denote the precise security definition fully detailed in this thesis, and roman lettering, e.g. IND-CPA, to refer to notions and proofs detailed by other authors (that may subtly vary from the ones presented herein).

The definitions of IND-CPA for symmetric key encryption (SKE) and IND-CCA for public key encryption (PKE) detailed later in this section relate to the privacy (confidentiality) of the scheme and ignore the goals of integrity (a recipient is assured that the message it receives is the one sent without accidental changes or intentional tampering), and authenticity (the receiver is convinced of the origin of the message). Achieving these tangential goals requires tools such as message authentication codes (MACs) for SKE and digital signatures for PKE. In the SKE setting, the notion of authenticated encryption with associated data (AEAD) [46, 175] combines these properties in one clean definition and for some time this has been the gold standard. This thesis will only consider privacy unless explicitly stated; notions of integrity and authenticity are beyond the intended scope.

In the following sections, pictorial representations indicate that the adversary sends one pair of messages and receives a single challenge ciphertext from the challenger which is generally referred to as the single-query, left-or-right setting. If we allow the adversary to submit a number of message pairs and receive a number of challenge ciphertexts, it is possible to use a hybrid argument to show that in fact this is equivalent[3] (up to a (polynomial) tightness factor) to the single-query scenario. While it is often intuitively (and formally) easier to use single-query definitions, there are a number of reasons to employ the multi-query versions. Firstly, these hybrid arguments do not work for RKA and KDM security definitions, meaning that the single- and multi-query scenarios are two distinct cases. The second reason is tightness [50, 131, 89]: these hybrid arguments incur a loss of a factor of q (where q is the number of queries the adversary makes) in

[2] Note that this is short for *security against key recovery*, the goal of the scheme.

[3] Intuitively a successful adversary uses one of the challenge ciphertexts to distinguish, hence in the PPT setting an adversary is only allowed a polynomial number of challenge ciphertexts.

the security bound for the multi-query case, and in some scenarios this loss may be unacceptable and using the multi-query setting as a starting point is preferable. For these reasons the formal definitions refer to this multi-query scenario, with these queries and the challenger's responses represented by a 'left-or-right oracle'. The decision to use left-or-right notions in this section is for ease of exposition—later on in Chapter 4 it will be more intuitive to use *real-or-random* notions where the adversary is allowed to submit a message and receives either a legitimate encryption of that message or an encryption of a random element of the message space. For the $\mathsf{IND\text{-}CPA}$ and $\mathsf{IND\text{-}CCA}$ notions in this chapter, left-or-right and real-or-random are equivalent notions [32], and are equivalent to two further notions: *find-then-guess* and *semantic security*[4] (introduced by Goldwasser and Micali [120]), which are beyond the scope of this work.

Many algorithms throughout this thesis will take as input a security parameter λ. We will use $\mathsf{x} \leftarrow \mathsf{Alg}(\lambda)$ and assume that all algorithms can interpret integer values as bitstrings and vice versa. Many authors write this as $\mathsf{x} \leftarrow \mathsf{Alg}(1^{\lambda})$ to reflect the fact that PPT algorithms run in time that is polynomial in the size of the input.

2.2 Symmetric Encryption

A symmetric encryption scheme Σ with message space $\mathcal{M}$ is defined by a tuple of algorithms $(\mathsf{Pg}, \mathsf{Kg}, \mathsf{E}, \mathsf{D})$ which operate as follows:

- Pg: Parameter generation takes as input a security parameter λ and outputs some public parameters pp, we write this as $\mathsf{pp} \leftarrow \mathsf{Pg}(\lambda)$.

- Kg: Key generation takes as input parameters pp and outputs a symmetric key k, denoted by $\mathsf{k} \leftarrow \mathsf{Kg}(\mathsf{pp})$.

- E: Encryption takes as input parameters pp, a message $\mathsf{m} \in \mathcal{M}$ and key k and outputs a ciphertext c, written $\mathsf{c} \leftarrow \mathsf{E}_{\mathsf{k}}(\mathsf{m}, \mathsf{pp})$.

- D: Decryption takes parameters pp, a ciphertext c and key k and outputs either a message $\mathsf{m} \in \mathcal{M}$ or error symbol $\perp$ and we write this $\mathsf{m}/\perp \leftarrow \mathsf{D}_{\mathsf{k}}(\mathsf{c}, \mathsf{pp})$.

Correctness requires that $\mathsf{D}_{\mathsf{k}}(\mathsf{c}, \mathsf{pp}) = \mathsf{m}$ for all $\mathsf{m} \in \mathcal{M}$, all $\mathsf{c} \leftarrow \mathsf{E}_{\mathsf{k}}(\mathsf{m}, \mathsf{pp})$ and all $\mathsf{k} \leftarrow \mathsf{Kg}(\mathsf{Pg}(\lambda))$. Parameter and key generation are randomised algorithms, the encryption algorithm may be randomised or stateful, decryption is deterministic. For many instantiations in the literature and this

[4] Roughly: a scheme is secure if an adversary, when given a ciphertext (and length of the plaintext), must use an 'infeasible' amount of resources to gain any additional information about the underlying plaintext message.

work, the parameter generation algorithm Pg and the parameters pp that it outputs are omitted for clarity; we will often write $\mathsf{c} \leftarrow \mathsf{E_k(m)}$ for encryption and $\mathsf{m} \leftarrow \mathsf{D_k(c)}$ for decryption. Note that many authors use the notation $\mathsf{E(k, m)}$ to denote that message m is encrypted under key k, this thesis will use the subscript notation throughout. Many authors focusing on the single-user setting choose to omit parameter generation; however, to reflect the multi-query definitions employed later in this work we make a distinction between parameter and key generation.

The most well known and widely used primitives for constructing symmetric encryption schemes are stream ciphers (such as RC4) and block ciphers (such as AES and DES). Symmetric encryption requires the keys to be securely distributed to each party involved in the communication, and this is often a challenging task—in a network of n users each pair of communicating parties need to agree a key, requiring $\frac{n(n-1)}{2}$ keys. In many scenarios this will be an insurmountable barrier, and use of public key cryptography is desirable.

2.2.1 Indistinguishability under Chosen-Plaintext Attack

We now describe the $\mathsf{IND\text{-}CPA}$ game for a symmetric encryption scheme $\Sigma = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{E}, \mathsf{D})$.

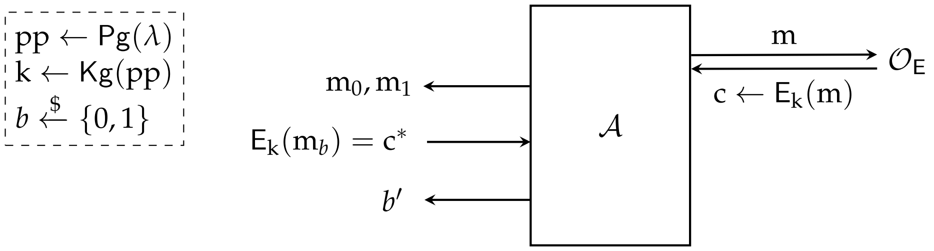

$\mathsf{pp} \leftarrow \mathsf{Pg}(\lambda)$
$\mathsf{k} \leftarrow \mathsf{Kg}(\mathsf{pp})$
$b \xleftarrow{\$} \{0,1\}$
$\mathsf{m_0, m_1}$
$\mathsf{E_k(m}_b) = \mathsf{c}^*$
b'
$\mathcal{A}$
m
$\mathcal{O}_\mathsf{E}$
$\mathsf{c} \leftarrow \mathsf{E_k(m)}$

Figure 2.1: $\mathsf{IND\text{-}CPA}$ *game for Symmetric Encryption*

The challenger runs Pg to get parameters pp and Kg to get a key k, and chooses the challenge bit b. Fig. 2.1 pictorially represents the game in the case where the adversary can only make one 'left-or-right' query. $\mathcal{A}$ runs and eventually outputs b' (as its guess for which 'world' it is in) with the assistance of an encryption oracle $\mathcal{O}_\mathsf{E}$. We say that $\mathcal{A}$ wins iff $b' = b$, and the scheme Σ is $\mathsf{IND\text{-}CPA}$ secure if $\mathcal{A}$'s advantage is no better than guessing. A more formal definition is given in Def. 2.1.

Note that the encryption algorithm aborts if the input message is not in the message space—this behaviour will be assumed throughout the thesis and explicit mention will often be omitted. The left-or-right oracle LR_b is written with a subscript b to emphasise that it has the challenge bit hardwired. The oracles $\mathsf{E}(\cdot)$ and $\mathsf{LR}_b(\cdot)$ also take the public parameters pp as input, again this is assumed and explicit mention will be omitted for clarity. The LR_b oracle checks if the two messages $\mathsf{m_0}$ and $\mathsf{m_1}$ are of the same length; this stops trivial attacks on schemes where the length

```
$\mathbf{Exp}^{\mathsf{IND\text{-}CPA}\text{-}b}_{\Sigma,\,\mathcal{A}}(\lambda)$ :
    $\mathrm{pp} \leftarrow \mathsf{Pg}(\lambda)$
    $\mathrm{k} \leftarrow \mathsf{Kg}(\mathrm{pp})$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b,\,\mathsf{E}}(\mathrm{pp})$
    return $b'$

$\mathsf{E}(\mathrm{m})$ :
    if $\mathrm{m} \notin \mathcal{M}$ then
        return $\perp$
    $\mathrm{c} \leftarrow \mathsf{E}_{\mathrm{k}}(\mathrm{m})$
    return $\mathrm{c}$
```

```
$\mathsf{LR}_b(\mathrm{m}_0, \mathrm{m}_1)$ :
    if $\mathrm{m}_0$ or $\mathrm{m}_1 \notin \mathcal{M}$ then
        return $\lightning$
    if $|\mathrm{m}_0| \neq |\mathrm{m}_1|$ then
        return $\lightning$
    $\mathrm{c} \leftarrow \mathsf{E}_{\mathrm{k}}(\mathrm{m}_b)$
    return $\mathrm{c}$
```

Figure 2.2: *The experiment defining* IND-CPA *security for symmetric encryption.*

of the ciphertext depends on the message length.

Definition 2.1 (IND-CPA Security for Symmetric Encryption)**.** *Let* $\Sigma = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{E}, \mathsf{D})$ *be a symmetric encryption scheme. Then the* IND-CPA *advantage for an adversary* $\mathcal{A}$ *against* Σ *is defined by*

$$\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Sigma,\,\mathcal{A}}(\lambda) \overset{def}{=} \left| \sum_{b\in\{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}CPA}\text{-}b}_{\Sigma,\,\mathcal{A}}(\lambda) = 1\right] \right|$$

where experiment $\mathbf{Exp}^{\mathsf{IND\text{-}CPA}\text{-}b}_{\Sigma,\,\mathcal{A}}$ *is given in Fig. 2.2.*

The $\sum_{b\in\{0,1\}}(-1)^b$ notation will be used throughout this thesis and is a consequence of page-width constraints.

2.3 Public Key Encryption

In public key cryptography each entity has a public key and a secret key. If Alice wants to send a message to Bob she looks up Bob's public key pk_B and uses that to encrypt her message, Bob then uses his secret key sk_B to decrypt. PKE was invented independently in the 1970s by Cocks and Ellis [111, 90, 112] (only declassified in 1997) and Diffie and Hellman [103]. A public key encryption scheme Π with message space $\mathcal{M}$ is defined by a tuple of algorithms $(\mathsf{PGen}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ which operate as follows:

- PGen: Parameter generation takes as input a security parameter λ and outputs some public parameters pp, we write this as $\mathrm{pp} \leftarrow \mathsf{PGen}(\lambda)$.
- KGen: Key generation takes as input parameters pp and outputs a pubic (encryption) key pk and corresponding secret (decryption) key sk, denoted $(\mathrm{sk}, \mathrm{pk}) \leftarrow \mathsf{KGen}(\mathrm{pp})$.

- Enc: Encryption takes as input parameters pp, a message $\mathrm{m} \in \mathcal{M}$ and public key pk and outputs a ciphertext c, written $\mathrm{c} \leftarrow \mathsf{Enc}_{\mathrm{pk}}(\mathrm{m}, \mathrm{pp})$.

- Dec: Decryption takes parameters pp, a ciphertext c and decryption key k and outputs either a message $\mathrm{m} \in \mathcal{M}$ or error symbol $\perp$ and we write this $\mathrm{m}/\perp \leftarrow \mathsf{Dec}_{\mathrm{sk}}(\mathrm{c}, \mathrm{pp})$.

Correctness requires that $\mathsf{Dec}_{\mathrm{sk}}(\mathrm{c}, \mathrm{pp}) = \mathrm{m}$ for all $\mathrm{m} \in \mathcal{M}$, all $\mathrm{c} \leftarrow \mathsf{Enc}_{\mathrm{pk}}(\mathrm{m}, \mathrm{pp})$ and all $(\mathrm{sk}, \mathrm{pk}) \leftarrow \mathsf{KGen}(\mathsf{PGen}(\lambda))$. Similarly to symmetric encryption, parameter and key generation are randomised algorithms, the encryption algorithm may be randomised or stateful, decryption is deterministic. Again we will often omit the parameters pp and write $\mathrm{c} \leftarrow \mathsf{Enc}_{\mathrm{pk}}(\mathrm{m})$ for encryption and $\mathrm{m} \leftarrow \mathsf{Enc}_{\mathrm{pk}}(\mathrm{c})$ for decryption. The message space $\mathcal{M}$ will often depend on the parameter generation algorithm PGen, and sometimes KGen too in cases like vanilla RSA. Many PKE schemes work in algebraic groups so the parameters could be for example $\mathrm{pp} = (\mathbb{G}, \mathrm{g})$, a description of the group and a generator.

2.3.1 Indistinguishability under Chosen-Ciphertext Attack

We now give the $\mathsf{IND\text{-}CCA}$ game for a public key encryption scheme $\Pi = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$.

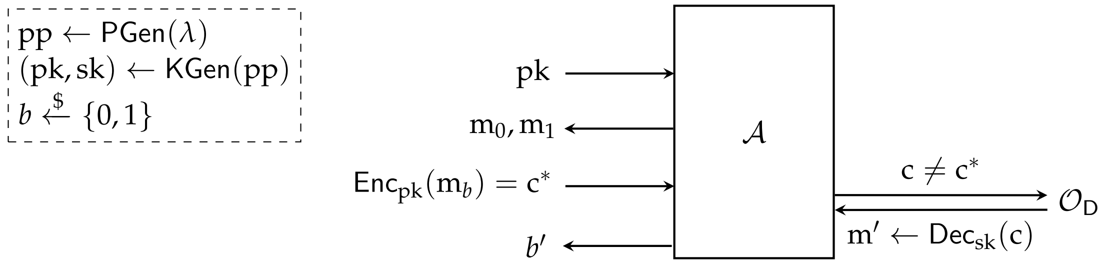


Figure 2.3: $\mathsf{IND\text{-}CCA}$ *game for public key encryption*

The diagram in Fig. 2.3 represents a single-user, single-query version of the $\mathsf{IND\text{-}CCA}$ game for public key encryption. The challenger runs the key generation algorithm to get a key pair $(\mathrm{pk}, \mathrm{sk})$, and chooses the challenge bit b. $\mathcal{A}$ runs and eventually outputs b' with the assistance of a decryption oracle $\mathcal{O}_{\mathsf{D}}$ (note that since $\mathcal{A}$ has the public key, there is no need for an encryption oracle). We say that $\mathcal{A}$ wins iff $b' = b$, and the scheme Π is $\mathsf{IND\text{-}CCA}$ secure if $\mathcal{A}$'s advantage is no better than guessing. A more formal version of multi-user, multi-query $\mathsf{IND\text{-}CCA}$ encryption is given in Def. 2.2.

The 'forbidden list' FL ensures that the adversary cannot trivially win by asking to decrypt something it has received from the left-or-right oracle. The forbidden list FL is initially empty, and we denote this by $\emptyset$ for ease of exposition. Note that the additions to FL are pairs of values,

```
$\mathbf{Exp}^{\mathsf{IND\text{-}CCA\text{-}}b}_{\Pi,\mathcal{A}}(\lambda)$ :
    $\mathrm{pp} \leftarrow \mathsf{PGen}(\lambda)$
    $t \leftarrow 0$
    $\mathrm{FL} \leftarrow \emptyset$
    $b' \leftarrow \mathcal{A}^{\mathsf{New},\,\mathsf{LR}_b,\,\mathsf{Dec}}(\mathrm{pp})$
    return $b'$

$\mathsf{New}()$:
    $t \leftarrow t + 1$
    $(\mathrm{pk}_t, \mathrm{sk}_t) \leftarrow \mathsf{KGen}(\mathrm{pp})$
    return $\mathrm{pk}_t$
```

```
$\mathsf{LR}_b(\mathrm{m}_0, \mathrm{m}_1, i)$ :
    if $\mathrm{m}_0$ or $\mathrm{m}_1 \notin \mathcal{M}$ then
        return $\lightning$
    if $|\mathrm{m}_0| \neq |\mathrm{m}_1|$ then
        return $\lightning$
    $\mathrm{c} \leftarrow \mathsf{Enc}_{\mathrm{sk}_i}(\mathrm{m}_b)$
    $\mathrm{FL} \leftarrow \mathrm{FL} \cup \{(\mathrm{c}, i)\}$
    return $\mathrm{c}$

$\mathsf{Dec}(\mathrm{c}, i)$ :
    if $(\mathrm{c}, i) \in \mathrm{FL}$ then
        return $\lightning$
    $\mathrm{m} \leftarrow \mathsf{Dec}_{\mathrm{sk}_i}(\mathrm{c})$
    return $\mathrm{m}$
```

Figure 2.4: *The experiment defining* IND-CCA *security for public key encryption.*

so for example if $\mathcal{A}$'s first LR query results in (c, i) being added to FL, it could then immediately ask Dec for (c, i') for any $i' \neq i$.

Definition 2.2 (IND-CCA Security for Public Key Encryption). *Let* $\Pi = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public key encryption scheme. The adversary* $\mathcal{A}$ *can call the key generation algorithm* KGen *to create new users for the scheme, and will receive their public key. Then the* IND-CCA *advantage for an adversary* $\mathcal{A}$ *against* Π *is defined by*

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\Pi,\mathcal{A}}(\lambda) \overset{def}{=} \left| \sum_{b\in\{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}CCA\text{-}}b}_{\Pi,\mathcal{A}}(\lambda) = 1\right] \right|$$

where experiment $\mathbf{Exp}^{\mathsf{IND\text{-}CCA\text{-}}b}_{\Pi,\mathcal{A}}$ *is given in Fig. 2.4.*

The definition of indistinguishability under chosen-ciphertext security given here is the IND-CCA2 notion developed by Rackoff and Simon [173], which is stronger than the IND-CCA1 notion given by Naor and Yung [163]. The Naor-Yung definition is also known as a 'lunchtime attack' because it models the scenario when a user's computer is compromised while they are out to lunch: the adversary is only allowed to query the decryption oracle Dec before it receives the challenge ciphertext.

Many known IND-CCA-secure PKE schemes take an IND-CPA-secure scheme and use a non-interactive proof system; the TDH2 'Signed ElGamal' scheme of Gennaro and Shoup [184], the Cramer-Shoup encryption scheme [93] and the DDN scheme by Dolev, Dwork and Naor [105] follow this approach. This strategy means that intuitively an adversary in the IND-CCA game needs to prove that it has knowledge of a plaintext in some valid ciphertext to win.

2.3.2 Plaintext Checking Oracles

Okamoto and Pointcheval [165] introduced a notion for the PKE setting that is weaker than chosen-ciphertext security, where the adversary has access to an oracle that on input a ciphertext c and plaintext message m outputs 1 if c is a valid encryption of m and 0 if it is not. The authors called it a Plaintext Checking Attack (PCA) and since no restrictions are placed on what can be submitted (even the challenge ciphertext is allowed), indistinguishability notions don't make much sense so $\mathsf{OW\text{-}PCA}$ is the target for some applications where the oracle is a more appropriate representation of reality than a CCA oracle. In the next section we describe a similar notion for hybrid encryption.

2.4 Hybrid Encryption

While public key encryption solves the key transport problem, instantiations are generally considerably slower than symmetric schemes and thus in practice a combination of the two is desirable. We can succinctly describe a convergence of these goals using the notion of *hybrid encryption*. We follow the work of Dent [100] and regard a hybrid encryption scheme as a public encryption scheme that uses a (keyed) symmetric encryption scheme in a black-box way. Many hybrid encryption schemes (but not all, see EPOC-2 [166]) can be separated to regard an asymmetric part and a symmetric part; this is known as the KEM-DEM framework. This means that when Alice wants to send a message m to Bob she picks a random symmetric key k, encrypts k under Bob's public key $\mathsf{pk_B}$ and encrypts m using k to yield a two-component ciphertext $(\omega, \mathsf{C}) = (\mathsf{Enc}_{\mathsf{pk_B}}(\mathsf{k}), \mathsf{E_k}(\mathsf{m}))$. When Bob receives this ciphertext, he can 'decapsulate' k using his secret key $\mathsf{sk_B}$, then decrypt the message.

A KEM-DEM encryption scheme Hyb consists of a key encapsulation mechanism $\mathsf{KEM} = (\mathsf{KEM.PGen}, \mathsf{KEM.KGen}, \mathsf{KEM.Encap}, \mathsf{KEM.Decap})$, a data encapsulation mechanism $\mathsf{DEM} = (\mathsf{DEM.Pg}, \mathsf{DEM.E}, \mathsf{DEM.D})$, and often a key derivation function $\mathsf{KDF} : \mathcal{K}_{\mathsf{KEM}} \to \mathcal{K}_{\mathsf{DEM}}$ as a compatibility layer in between. We will use DEMs constructed from symmetric encryption schemes, but note that DEMs only require encryption and decryption components since their key generation is done by the KEM. The syntax of a hybrid encryption scheme with message space $\mathcal{M}$ is as follows, this is detailed pictorially in Fig. 2.5 and the algorithms are described in Fig. 2.6.

- Parameter and key generation for the KEM are done in the same manner as for a public key encryption scheme. Key encapsulation $\mathsf{KEM.Encap}$ takes as input a public key, and returns both a (symmetric) key $\mathsf{k} \in \mathcal{K}_{\mathsf{KEM}}$ and an encapsulation ω thereof. Key decapsulation

KEM.Decap takes as input a private key and a purported key encapsulation and returns a key in $\mathcal{K}_{\mathsf{KEM}}$ or some designated error symbol $\perp$.

- Data encapsulation DEM.E takes as input a message $\mathrm{m} \in \mathcal{M}$ and a symmetric key in $\mathcal{K}_{\mathsf{DEM}}$ and outputs an encryption C. A data decapsulation DEM.D takes a message encapsulation C and a symmetric key in $\mathcal{K}_{\mathsf{DEM}}$ and outputs the message m or error symbol $\perp$.
- A key derivation function KDF is simply a deterministic algorithm implementing a mapping from $\mathcal{K}_{\mathsf{KEM}}$ to $\mathcal{K}_{\mathsf{DEM}}$. In addition to some key k the algorithm takes as input KEM.pk and pp[DEM] (in order to determine $\mathcal{K}_{\mathsf{KEM}}$ and $\mathcal{K}_{\mathsf{DEM}}$).

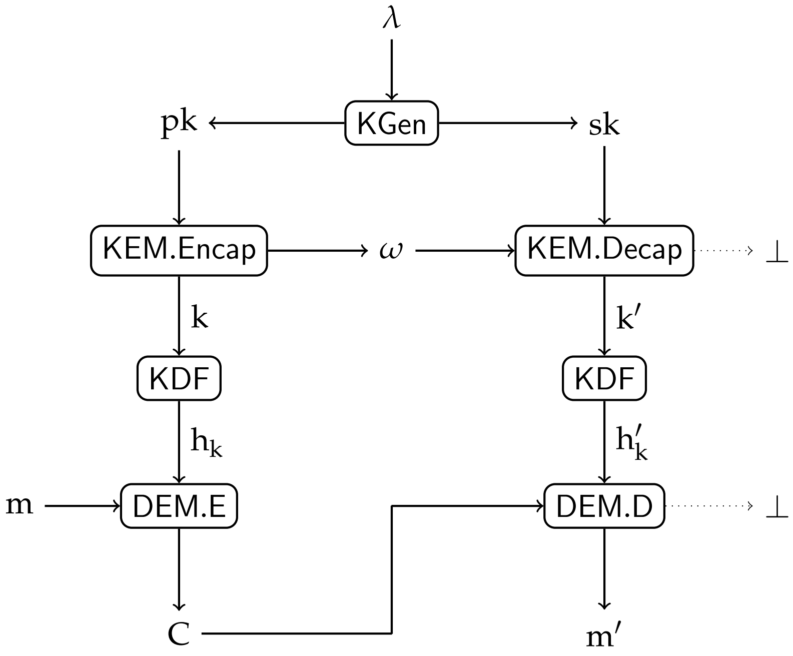


Figure 2.5: *Diagram of Hybrid Encryption*

The diagram in Fig. 2.5 omits the parameter generation algorithm and the distribution of public parameters to the other algorithms. We use the term *protokey* to describe the input to the KDF (denoted k in Figures 2.5 and 2.6). The error symbol $\perp$ could be produced by the KEM.Decap algorithm if it is given an invalid ω, or by DEM.D if it is given an invalid C.

Figure 2.6 emphasises the fact that hybrid encryption schemes are public key encryption schemes, where the ciphertext is comprised of two components C and ω. Note that each time we want to send a message we choose the symmetric key k so this is actually part of encryption rather than key generation of the overall PKE scheme.

While hybrid encryption has been in widespread use ever since the advent of public key cryptosystems, the first formalisation of the KEM-DEM paradigm was given by Cramer and

```
$\mathsf{Hyb.PGen}(\lambda)$
    $\mathrm{pp}[\mathsf{KEM}] \leftarrow \mathsf{KEM.PGen}(\lambda)$
    $\mathrm{pp}[\mathsf{DEM}] \leftarrow \mathsf{DEM.Pg}(\lambda)$
    return $(\mathrm{pp}[\mathsf{KEM}], \mathrm{pp}[\mathsf{DEM}])$

$\mathsf{Hyb.Enc}(\mathrm{pp}, \mathrm{pk}, \mathrm{m})$
    $(\mathrm{k}, \omega) \leftarrow \mathsf{KEM.Encap}_{\mathrm{pk}}()$
    $\mathrm{h}_{\mathrm{k}} \leftarrow \mathsf{KDF}_{\mathrm{pp},\mathrm{pk}}(\mathrm{k})$
    $\mathrm{C} \leftarrow \mathsf{DEM.E}_{\mathrm{h}_{\mathrm{k}}}(\mathrm{m})$
    return $(\omega, \mathrm{C})$
```

```
$\mathsf{Hyb.KGen}(\mathrm{pp})$
    $(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathsf{KEM.KGen}(\mathrm{pp}[\mathsf{KEM}])$
    return $(\mathrm{pk}, \mathrm{sk})$

$\mathsf{Hyb.Dec}(\mathrm{pp}, \mathrm{sk}, \omega, \mathrm{C})$
    $\mathrm{k} \leftarrow \mathsf{KEM.Decap}_{\mathrm{sk}}(\omega)$
    $\mathrm{h}_{\mathrm{k}} \leftarrow \mathsf{KDF}_{\mathrm{pp},\mathrm{pk}}(\mathrm{k})$
    $\mathrm{m} \leftarrow \mathsf{DEM.D}_{\mathrm{h}_{\mathrm{k}}}(\mathrm{C})$
    return $\mathrm{m}$
```

Figure 2.6: *Algorithms for a Hybrid Cryptosystem* Hyb.

Shoup [94, 95] in 2001. This followed the work of Shoup [182] at Eurocrypt '00 that sketched a definition for IND-CCA security of KEMs, and stated that combining a secure KEM with a pseudorandom bit generator, an almost-XOR universal function [148] for message authentication and a symmetric scheme that works using a one-time pad yields an IND-CCA-secure hybrid scheme. Cramer and Shoup gave formal security definitions of IND-CPA and IND-CCA security for both the KEM and the DEM part and proved that in the standard model, where the key derivation function only needs to be (close to) balanced, the public key cryptosystem inherits security from its constituent parts, e.g. IND-CCA security for both the KEM and the DEM part is a sufficient condition to obtain an IND-CCA-secure hybrid PKE scheme. They additionally show that the Cramer-Shoup cryptosystem [93] presented at CRYPTO '98 and its generalisation to hash proof systems neatly fits the hybrid framework, and the journal article [96] by Cramer and Shoup in 2003 covers both the cryptosystem from 1998 and their formalisation of hybrid cryptography that followed soon after.

In 2004 Kurosawa and Desmedt [150] were the first to show that in fact it is possible to weaken the security requirement on the KEM part, while maintaining IND-CCA security for the combined scheme. They gave a hybrid scheme that replaces some of Cramer-Shoup's algebraic components with information-theoretically secure SKE primitives (resulting in an efficiency gain over the constructions suggested previously by Shoup [182] and Cramer and Shoup [93]), yet they could not prove the KEM to be IND-CCA secure[5] and Herranz et al. [127] showed in 2006 that it in fact was *not* IND-CCA secure. At Crypto '07 Hofheinz and Kiltz [130] showed that combining a notion called constrained CCA (IND-CCCA) security—a notion that is stronger than IND-CPA but strictly weaker than IND-CCA—with authenticated symmetric encryption yields an IND-CCA

[5] The Kurosawa-Desmedt hybrid scheme could not be rigourously cast in the hash proof system framework that Shoup described since the underlying scheme is not quite universal$_2$; it uses TCR hash functions, a computational component, whereas universal$_2$ is a statistical property.

secure hybrid PKE scheme.

In work that inspired the results in Chapter 3, Dent [99] looked at various constructions of KEMs from one-way secure public key cryptosystems (operating on a restricted message space). He modelled the key derivation function as a random oracle and considered it as part of the KEM. A typical example of such a construction is the use of a trapdoor function to encapsulate a protokey r that is subsequently hashed to yield a derived key $\mathsf{H}(\mathsf{r})$. He shows several elegant, generic KEM constructions that are IND-CCA secure based on fairly minimal assumptions on the encryption scheme used to encrypt the protokey. For instance, in the example above security is attained if the trapdoor function is one-way secure even in the presence of an oracle that checks whether a ciphertext is a valid ciphertext or not (i.e., the actual range of the trapdoor function is easily recognizable by the adversary), which Dent calls $\mathsf{OW\text{-}CPA+}$ security. If the KEM is constructed from a *randomised* public key cryptosystem, security based on one-wayness is proven, provided that there is an efficient plaintext-ciphertext checking oracle that, when given a message and ciphertext pair, correctly determines whether the ciphertext is an encryption of the message or not. A fact that we will need later is the dichotomy in KEMs depending on the availability of this key-encapsulation–encapsulated-key checking oracle $\mathsf{KEM.Chk}_{\mathsf{pk}}(\omega, \mathsf{k})$ that, on input a key encapsulation ω and purported encapsulated (proto) key k decides whether $\mathsf{KEM.Decap}_{\mathsf{sk}}(\omega) = \mathsf{k}$ or not. This leads to the following two types of KEMs; each type will give a different reduction in the security analysis in Section 3.5:

- In $\mathsf{TYPE\text{-}1}$ KEMs there is an efficient checking oracle $\mathsf{KEM.Chk}_{\mathsf{pk}}(\omega, \mathsf{k})$. This class encompasses all schemes that determine the encapsulation ω deterministically based on the key k, including the usual schemes based on trapdoor permutations/functions. Diffie-Hellman type KEMs in a pairing-based setting (where DDH is easy) can also be part of this class.

- In $\mathsf{TYPE\text{-}2}$ KEMs there is no efficient checking oracle. This class contains all $\mathsf{IND\text{-}CPA}$ secure KEMs. (The lack of a checking oracle means that the reduction will need to guess whether a query $\mathsf{H}(\mathsf{k})$ corresponds to a challenge ciphertext or not, leading to a less tight reduction.)

We refer to the above-mentioned articles for general security notions for KEMs and DEMs; for Chapter 3 we will be particularly concerned with $\mu\mathsf{OW\text{-}CCA}$ for KEMs, where the μ indicates there can be multiple key pairs in the game and the adversary can make multiple encapsulation queries for each public key (see Def. 2.3) and $\mathsf{IND\text{-}CCA}$ for DEMs (see Def. 2.4).

2.4.1 Security Models for KEMs and DEMs

When we regard a KEM on its own, the one-way definition of security is more intuitively useful than an indistinguishability-based notion—this is because in practice the protokey output by the KEM is usually fed into some hash function to derive the key used by the DEM. Indeed some applications such as identification schemes [12] and selective-opening-secure schemes [128] only require OW-secure KEMs. Later on we will need the μOW-CCA definition of security for key encapsulation mechanisms, so we detail this security game now. $\mathcal{A}$ can obtain as many key encapsulations as desired (specifying an index i for the public key it wishes to receive an encryption under) and needs to predict correctly the key of one of the encapsulations. $\mathcal{A}$ can decapsulate values of ω, under specified secret key sk_i, as long as such a pair was not created by Enc. The normal OW-CCA game limits the number of Encap queries to one.

Definition 2.3. *Let* $\mathsf{KEM} = (\mathsf{KEM.PGen}, \mathsf{KEM.KGen}, \mathsf{KEM.Encap}, \mathsf{KEM.Decap})$ *be a key encapsulation mechanism. Then the* μOW-CCA *advantage of an adversary* $\mathcal{A}$ *against* KEM *is defined by*

$$\mathbf{Adv}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{A}}(\lambda) \overset{def}{=} \mathbf{Pr}\left[\mathbf{Exp}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{A}}(\lambda) = 1\right]$$

where the experiment $\mathbf{Exp}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{A}}(\lambda)$ *is given in Fig. 2.7.*

```
$\mathbf{Exp}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{A}}(\lambda)$:
    $\mathrm{pp} \leftarrow \mathsf{KEM.PGen}(\lambda)$
    $t \leftarrow 0$
    $n \leftarrow 0$
    $\mathbf{sk} \leftarrow ()$
    $\mathrm{FL} \leftarrow \emptyset$
    $(j, \mathrm{k}') \leftarrow \mathcal{A}^{\mathsf{New,\ H,\ Encap,\ Decap}}(\mathrm{pp})$
    if $\mathrm{k}' = \mathrm{k}_j$
        return 1
    return 0

$\mathsf{Encap}(i)$:
    $n \leftarrow n + 1$
    $(\omega, \mathrm{k}) \leftarrow \mathsf{KEM.Encap}_{\mathrm{pk}_i}()$
    $\mathrm{FL} \leftarrow \mathrm{FL} \cup \{(\omega, i)\}$
    $\mathrm{k}_n \leftarrow \mathrm{k}$
    return $\omega$
```

```
$\mathsf{New}()$:
    $t \leftarrow t + 1$
    $(\mathrm{pk}_t, \mathrm{sk}_t) \leftarrow \mathsf{KGen}(\mathrm{pp})$
    Append $\mathrm{sk}_t$ to $\mathbf{sk}$
    return $\mathrm{pk}_t$

$\mathsf{Decap}(\omega, i)$:
    if $(\omega, i) \in \mathrm{FL}$ then
        return $\lightning$
    $\mathrm{k} \leftarrow \mathsf{KEM.Decap}_{\mathrm{sk}_i}(\omega)$
    return k
```

Figure 2.7: *The* μOW-CCA *security experiment for KEMs.*

Also required later on are security notions for data encapsulation mechanisms, and we give the standard notion of indistinguishability under chosen-ciphertext attack, or IND-CCA, here. For the purposes of this definition we assume there is an algorithm $\mathsf{DEM.Kg}$ that generates keys for DEM, even though in many constructions this algorithm will be part of the KEM.

Definition 2.4. *Let* $\mathsf{DEM} = (\mathsf{DEM.Pg}, \mathsf{DEM.Kg}, \mathsf{DEM.E}, \mathsf{DEM.D})$ *be a data encapsulation mechanism. Then the* $\mathsf{IND\text{-}CCA}$ *advantage of an adversary* $\mathcal{A}$ *against* DEM *is defined by*

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}}(\lambda) \overset{def}{=} \left| \sum_{b \in \{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}CCA}\text{-}b}_{\mathsf{DEM},\,\mathcal{A}}(\lambda) = 1\right] \right|$$

where the experiment $\mathbf{Exp}^{\mathsf{IND\text{-}CCA}\text{-}b}_{\mathsf{DEM},\,\mathcal{A}}(\lambda)$ *is given in Fig. 2.8.*

```
$\mathbf{Exp}^{\mathsf{IND\text{-}CCA}\text{-}b}_{\mathsf{DEM},\,\mathcal{A}}(\lambda)$:
    $\mathsf{pp} \leftarrow \mathsf{DEM.Pg}(\lambda)$
    $\mathrm{FL} \leftarrow \emptyset$
    $\mathrm{k} \leftarrow \mathsf{DEM.Kg}(\mathrm{pp})$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b,\,\mathsf{D}}(\mathrm{pp})$
    return $b'$

$\mathsf{LR}_b(\mathrm{m}_0, \mathrm{m}_1)$:
    if $|\mathrm{m}_0| \neq |\mathrm{m}_1|$ then
        return $\lightning$
    $\mathrm{C} \leftarrow \mathsf{DEM.E}_{\mathrm{k}}(\mathrm{m}_b)$
    $\mathrm{FL} \leftarrow \mathrm{FL} \cup \{\mathrm{C}\}$
    return $\mathrm{C}$

$\mathsf{D}(\mathrm{C})$:
    if $\mathrm{C} \in \mathrm{FL}$ then
        return $\lightning$
    $\mathrm{m} \leftarrow \mathsf{DEM.D}_{\mathrm{k}}(\mathrm{C})$
    return $\mathrm{m}$
```

Figure 2.8: *The* $\mathsf{IND\text{-}CCA}$ *security experiment for* DEM.

This is very similar to the definition of IND-CCA security for symmetric encryption; the single-query (i.e. just one query to LR) variant of this definition is often used to capture the 'one-shot' nature of DEM encryption, but we seek a more general result so use this multi-query version.

2.5 Pseudorandom Functions

While this thesis is mainly focused on security models for encryption, we will require one other tool to complete our proofs, namely the definition of a pseudorandom function (PRF). PRFs, and their cousins pseudorandom permutations (PRPs), are often used in the design of cryptographic protocols (a block cipher is a family of permutations).

Definition 2.5 (Pseudorandom functions). *Let* $\mathrm{F} : \mathcal{I} \times \mathcal{D} \to \mathcal{R}$ *be a family of functions from domain* $\mathcal{D} = \{0,1\}^\lambda$ *to range* $\mathcal{R}$ *indexed by seeds* $\mathcal{I}$. *For* $\mathrm{x} \in \mathcal{I}$ *we let* $\mathrm{F}_\mathrm{x}(\mathrm{y}) : \mathcal{D} \to \mathcal{R}$ *be defined by* $\mathrm{F}_\mathrm{x}(\mathrm{y}) = \mathrm{F}(\mathrm{x}, \mathrm{y}) \;\; \forall \mathrm{y} \in \mathcal{D}$. *Let* $\mathrm{Fun}[\mathcal{D}, \mathcal{R}]$ *be the set of all functions from* $\mathcal{D}$ *to* $\mathcal{R}$. *Then the PRF advantage of an adversary* $\mathcal{A}$ *attacking* F *is given by*

$$\mathbf{Adv}^{\mathsf{PRF}}_{\mathrm{F},\,\mathcal{A}}(\lambda) \overset{def}{=} \left| \sum_{b \in \{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{PRF}\text{-}b}_{\mathrm{F},\,\mathcal{A}}(\lambda) = 1\right] \right|$$

where experiment $\mathbf{Exp}^{\mathsf{PRF}\text{-}b}_{\mathrm{F},\,\mathcal{A}}(\lambda)$ *is given in Fig. 2.9*

```
$\mathbf{Exp}^{\mathsf{PRF}\text{-}b}_{\mathrm{F},\mathcal{A}}(\lambda)$ :
    $\mathrm{x} \xleftarrow{\$} \mathcal{I}$
    $\mathrm{g} \xleftarrow{\$} \mathrm{Fun}[\{0,1\}^{\lambda}, \mathcal{R}]$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\lambda)$
    return $b'$

$\mathsf{LR}_b(\mathrm{y})$ :
    $\mathrm{z}_1 \leftarrow \mathrm{F}_{\mathrm{x}}(\mathrm{y})$
    $\mathrm{z}_0 \leftarrow \mathrm{g}(\mathrm{y})$
    return $\mathrm{z}_b$
```

Figure 2.9: *PRF experiment for function* F.

Intuitively F is a PRF if an adversary cannot distinguish oracle access to a function chosen from F from oracle access to a random function from the same domain and range as F. A similar definition emerges for weak-PRPs if we restrict $\mathcal{D} = \mathcal{R}$, and to obtain a definition for strong-PRPs we also give the adversary access to the inverse of F (or the inverse of the randomly chosen permutation g). This thesis will not investigate block cipher design or analysis, but in a proof in Section 3.5.2 we will require a PRF as a black box.

2.6 Assumptions in Cryptography

Public key cryptography requires a number of mathematical techniques; we will detail the most important aspects here. We will describe a number of problems and assumptions; the general approach is to define a problem (where an adversary is tasked with making a computation or computing something) and then make the assumption that solving this problem is hard, at least for some parameter choices.

We will require Euler's Phi Function, denoted $\varphi_{\mathsf{E}}(\mathrm{N})$, which is defined to be the number of positive integers less than or equal to N that are relatively prime to N, and Blum integers, which are values $\mathrm{N} = \mathrm{pq}$ where p and q are primes such that $\mathrm{p}, \mathrm{q} \equiv 3 \bmod 4$. Let $\mathsf{Primes}[\mathrm{x}]$ denote the set of primes of length x bits.

2.6.1 Hard Problems

In cryptographic security proofs the general tactic is to form a *reduction* from the scheme in question to a specific property of the primitive or an assumption that is assumed to be hard. When discussing block ciphers within modes of operation, the block cipher itself is often assumed to behave like a pseudorandom permutation (i.e. PRF where domain and range are equal), and in the realm of traditional public key encryption it is typical to use reductions to hard number-theoretic problems.

The idea is that one can construct public key encryption from so-called *trapdoor one-way func-*

tions, meaning functions that are efficient to compute yet difficult to invert without the knowledge of some extra information, known as the trapdoor. Perhaps the best known method of public key encryption is the (vanilla) RSA cryptosystem, the security of which relies on the RSA problem which is closely linked to the difficulty of factoring large integers.

2.6.1.1 Problems Related to Factoring

```
$\mathbf{Exp}^{\text{Factor}}_{\mathcal{A}}(\lambda)$ :
  $\mathsf{p}, \mathsf{q} \xleftarrow{\$} \mathsf{Primes}[\lambda/2]$
  $\mathsf{N} \leftarrow \mathsf{p} \cdot \mathsf{q}$
  $\mathsf{x}' \leftarrow \mathcal{A}(\mathsf{N})$
  if $\mathsf{x}' \in \{\mathsf{p}, \mathsf{q}\}$ then
    return 1
  else
    return 0
```

```
$\mathbf{Exp}^{\text{DCR-}b}_{\mathbb{Z}^*_{\mathsf{N}^2}, \mathcal{A}}(\lambda)$ :
  $\mathsf{p}, \mathsf{q} \xleftarrow{\$} \mathsf{Primes}[\lambda/2]$
  $\mathsf{N} \leftarrow \mathsf{p} \cdot \mathsf{q}$
  if $b = 1$ then
    $\mathsf{x} \xleftarrow{\$} \mathsf{CR}_{\mathsf{N}^2}$
  if $b = 0$ then
    $\mathsf{x} \xleftarrow{\$} \mathbb{Z}^*_{\mathsf{N}^2}$
  $b' \leftarrow \mathcal{A}(\mathsf{N}, \mathsf{x})$
  return $b'$
```

```
$\mathbf{Exp}^{\text{QR-}b}_{\mathbb{Z}^*_{\mathsf{N}}, \mathcal{A}}(\lambda)$ :
  $\mathsf{N} \xleftarrow{\$} \mathsf{Blum}[\lambda]$
  if $b = 1$ then
    $\mathsf{y} \xleftarrow{\$} \mathsf{QR}_{\mathsf{N}}$
  if $b = 0$ then
    $\mathsf{y} \xleftarrow{\$} \mathbb{Z}^*_{\mathsf{N}}[+1]$
  $b' \leftarrow \mathcal{A}(\mathsf{N}, \mathsf{y})$
  return $b'$
```

Figure 2.10: *Factoring, DCR and QR experiments*

Factoring. Let $\mathsf{N} = \mathsf{p} \cdot \mathsf{q}$ be an integer of length λ. We say that the *Factoring assumption* holds for N if it is hard to recover p (or q). Formally, for security parameter λ, the advantage of an adversary $\mathcal{A}$ against factoring N is defined by

$$\mathbf{Adv}^{\text{Factor}}_{\mathsf{N}, \mathcal{A}}(\lambda) \overset{def}{=} \mathbf{Pr}\left[\mathbf{Exp}^{\text{Factor}}_{\mathsf{N}, \mathcal{A}}(\lambda) = 1\right]$$

where $\mathbf{Exp}^{\text{Factor}}_{\mathsf{N}, \mathcal{A}}(\lambda)$ is given in Fig. 2.10.

DCR assumption. [167] Let N be a RSA modulus of length λ, and let $\mathsf{CR}_{\mathsf{N}^2} = \{\mathsf{u}^{\mathsf{N}} \bmod \mathsf{N}^2 | \mathsf{u} \in \mathbb{Z}^*_{\mathsf{N}^2}\}$ be the set of Nth Residues modulo N^2. We say that the Decision Composite Residue (DCR) assumption holds if it is hard for an adversary to decide whether elements of $\mathbb{Z}^*_{\mathsf{N}^2}$ are composite residues or not. More formally, the advantage of an adversary $\mathcal{A}$ against DCR in $\mathbb{Z}^*_{\mathsf{N}^2}$ is defined by

$$\mathbf{Adv}^{\text{DCR}}_{\mathbb{Z}^*_{\mathsf{N}^2}, \mathcal{A}}(\lambda) \overset{def}{=} \left| \sum_{b \in \{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\text{DCR-}b}_{\mathbb{Z}^*_{\mathsf{N}^2}, \mathcal{A}}(\lambda) = 1\right] \right|$$

where $\mathbf{Exp}^{\text{DCR-}b}_{\mathbb{Z}^*_{\mathsf{N}^2}, \mathcal{A}}(\lambda)$ is given in Fig. 2.10.

QR assumption. Let N be a Blum integer of bitlength λ. With $\mathbb{Z}^*_{\mathsf{N}}[+1]$ we denote the set of elements in $\mathbb{Z}^*_{\mathsf{N}}$ with Jacobi symbol $+1$ and with $\mathsf{QR}_{\mathsf{N}} := \{\mathsf{x}^2 \bmod \mathsf{N} : \mathsf{x} \in \mathbb{Z}^*_{\mathsf{N}}\}$ the set of Quadratic

Residues modulo N. We say that the Quadratic Residue (QR) assumption holds if it is hard for an adversary to decide whether elements of $\mathbb{Z}_N^*$ are quadratic residues or not. More formally, the advantage of an adversary $\mathcal{A}$ against QR in $\mathbb{Z}_N^*$ is defined by

$$\mathbf{Adv}^{\mathrm{QR}}_{\mathbb{Z}_N^*,\,\mathcal{A}}(\lambda) \overset{def}{=} \left| \sum_{b\in\{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\mathrm{QR}\text{-}b}_{\mathbb{Z}_N^*,\,\mathcal{A}}(\lambda) = 1\right]\right|$$

where $\mathbf{Exp}^{\mathrm{QR}\text{-}b}_{\mathbb{Z}_N^*,\,\mathcal{A}}(\lambda)$ is given in Fig. 2.10.

2.6.1.2 Problems Related to Discrete Log

The discrete logarithm problem (DLP), which asks an adversary, given g^x, to find x, has found numerous uses in cryptography, often using cyclic groups of (known) prime order. The decisional Diffie-Hellman (DDH) problem, which is used in the proof of security of the ElGamal [110] and Cramer-Shoup [93] encryption schemes, gives an adversary (g, g^x, g^y) and either g^{xy} or g^z for random z, and asks the adversary to decide which. Clearly an adversary that can solve the DLP can calculate x and y and thus solve DDH. A related problem is the computational Diffie-Hellman (CDH) problem, which asks an adversary, given (g, g^x, g^y), to compute g^{xy}. CDH is a weaker assumption than DDH since an adversary solving CDH can trivially win the DDH, but CDH is stronger than the DLP (this is not strict since in some groups both problems may be easy) [69, 136].

We can generalise the DDH assumption: the k-linear problem [130, 178] asks an adversary, given $(g_0, g_1, \ldots, g_k, g_1^{r_1}, \ldots, g_k^{r_k}, T)$ for $r_1, \ldots, r_k \in \mathbb{Z}_p^*$ and $g_0, \ldots, g_k \in \mathbb{G}$, to decide whether $T = g_0^{r_1+\cdots+r_k}$ or $T = g_0^z$ for some $z \in \mathbb{Z}_p^*$. The k-linear assumption says that this is a hard problem; the case $k = 1$ corresponds to DDH and $k = 2$ is the Decision Linear (DLIN) assumption [70].

```
$\mathbf{Exp}^{\mathrm{DLP}}_{\mathbb{G},\,\mathcal{A}}(\lambda)$ :
  $g \xleftarrow{\$} \mathbb{G}$
  $x \xleftarrow{\$} |\mathbb{G}|$
  $y \leftarrow g^x$
  $x' \leftarrow \mathcal{A}(g, y)$
  if $x = x'$ then
    return 1
  else
    return 0
```

```
$\mathbf{Exp}^{\mathrm{DDH}\text{-}b}_{\mathbb{G},\,\mathcal{A}}(\lambda)$ :
  $g \xleftarrow{\$} \mathbb{G}$
  $x, y, z \xleftarrow{\$} |\mathbb{G}|$
  if $b = 1$ then
    $c \leftarrow g^{xy}$
  if $b = 0$ then
    $c \leftarrow g^z$
  $b' \leftarrow \mathcal{A}(g, g^x, g^y, c)$
  return $b'$
```

Figure 2.11: *Discrete Logarithm Problem and Decisional Diffie-Hellman experiments*

Discrete Logarithm Problem. The *discrete logarithm problem (DLP)* over a group $\mathbb{G}$ (that may depend on the security parameter λ) holds if given $g \in \mathbb{G}$ and $y = g^x$ where $x \xleftarrow{\$} [|\mathbb{G}|]$, it is hard to find x. Formally, the advantage of an adversary $\mathcal{A}$ against DLP in $\mathbb{G}$ is defined by

$$\mathbf{Adv}^{\mathrm{DLP}}_{\mathbb{G},\,\mathcal{A}}(\lambda) \overset{def}{=} \mathbf{Pr}\left[\mathbf{Exp}^{\mathrm{DLP}}_{\mathbb{G},\,\mathcal{A}}(\lambda) = 1\right]$$

where $\mathbf{Exp}^{\mathrm{DLP}}_{\mathbb{G},\,\mathcal{A}}(\lambda)$ is given in Fig. 2.11.

DDH assumption. [103] The *decisional Diffie-Hellman (DDH) assumption* over a group $\mathbb{G}$ (that may depend on the security parameter λ) stipulates that $(\mathrm{g}, \mathrm{g}^{\mathrm{x}}, \mathrm{g}^{\mathrm{y}}, \mathrm{g}^{\mathrm{xy}}) \overset{c}{\approx} (\mathrm{g}, \mathrm{g}^{\mathrm{x}}, \mathrm{g}^{\mathrm{y}}, \mathrm{g}^{\mathrm{z}})$, where $\mathrm{g} \overset{\$}{\leftarrow} \mathbb{G}$ and $\mathrm{x}, \mathrm{y}, \mathrm{z} \overset{\$}{\leftarrow} [|\mathbb{G}|]$. More formally, the advantage of an adversary $\mathcal{A}$ against DDH in $\mathbb{G}$ is defined by

$$\mathbf{Adv}^{\mathrm{DDH}}_{\mathbb{G},\,\mathcal{A}}(\lambda) \overset{def}{=} \left|\sum_{b\in\{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\mathrm{DDH}\text{-}b}_{\mathbb{G},\,\mathcal{A}}(\lambda) = 1\right]\right|$$

where $\mathbf{Exp}^{\mathrm{DDH}\text{-}b}_{\mathbb{G},\,\mathcal{A}}(\lambda)$ is given in Fig. 2.11.

CHAPTER 3

Key-Dependent Message Security

Contents

3.1 Overview and Motivation **28**

3.2 Related Literature **30**

3.3 Security Models and Formalism **33**

3.3.1 IND-KDM-CPA Security of SKE 34

3.3.2 IND-KDM-CCA Security of PKE 35

3.3.3 Circular Security and Alternative Notions 36

3.4 Realising KDM Security **37**

3.4.1 Black et al.'s Random Oracle Scheme 37

3.4.2 Boneh et al.'s IND-KDM-CPA-Secure Scheme 38

3.4.3 Camenisch et al.'s IND-KDM-CCA-Secure Scheme 39

3.5 Achieving KDM Security of Hybrid Encryption **39**

3.5.1 Tools 40

3.5.2 Achieving KDM-CCA Security of Hybrid Encryption 43

3.6 Separation Results for KDM Security in the Hybrid Framework **58**

3.6.1 Writing Acar et al.'s es_2 as a KEM and DEM 59

3.6.2 Writing Cash et al.'s Π_{CPA} as a KEM and DEM 60

3.7 Conclusions **62**

3.1 Overview and Motivation

The work in this chapter is largely based on ***KDM Security in the Hybrid Framework*** *by Davies and Stam [98] published at CT-RSA 2014. The full version is available via ePrint [97].*

Standard definitions of security for encryption do not consider the scenario where the message being encrypted may depend on the key used for encryption. Most cryptographic constructs would consider this a dangerous abuse and standard security criteria (e.g. IND-CPA and IND-CCA) do not take this type of behaviour into account. This concern has been known since 1984 when Goldwasser and Micali [See [120] Section 5.1] noted the danger of encrypting messages that the adversary cannot find. As a straightforward example of how this concern can arise in practice, when full-disk encryption is performed on a hard drive, (a representation of) the encryption key will be encrypted under itself, along with the disk contents.

The general field of KDM security encompasses formal definitions, positive constructions and impossibility results for the behaviour of cryptographic primitives when an adversary is allowed some access to encryptions of the secret keys. The first formal treatment of this notion was given by Black, Rogaway and Shrimpton in 2002 [65]. Intuitively, a scheme is *key-dependent message* secure (IND-KDM-CPA) with respect to a class Ψ of (efficiently evaluatable) functions if an adversary cannot distinguish an oracle that, on input $\psi \in \Psi$, returns an encryption of ψ applied to the decryption key(s) from an oracle that returns encryptions of a fixed element of the message space. This definition and its chosen-ciphertext analogue are equivalent to standard notions of IND-CPA and IND-CCA security when Ψ is limited to constant functions. The KDM definitions are in fact stronger, as there exist schemes that are IND-CPA secure yet are trivially insecure in the KDM setting: for example, take an IND-CPA secure symmetric scheme and modify the encryption algorithm so that if the input is the encryption key then output the key in the clear, otherwise proceed as normal. Omitting a straightforward correctness argument, this scheme is still IND-CPA secure (since the probability that an adversary queries a message that is equal to the key is negligible), yet the adversary in the KDM game needs only one query of the identity function to distinguish.

Encrypting a system's keys may in fact be useful: in anonymous credential systems [83] key-dependent encryptions are generated so that users are discouraged from delegating their own secret keys[1]. Aside from being a theoretical consideration of practical attacks and a component for credential systems, the study of key-dependent message attacks has seen a number of further applications. As a contribution to the formal methods community, it was shown that the definition of key-dependent message security presented by Black et al. [65] could be used to prove the equivalence between *computational security* and *axiomatic security* (i.e. results that are proven in a formal calculus are computationally sound). While it is perhaps easier to motivate KDM security for symmetric encryption due to the example of disk encryption, the research focus on KDM-secure PKE constructions and the proliferation of works on anonymous credentials and formal methods demonstrate that there is appetite for research in KDM-secure PKE. Given that in practice PKE is almost always performed using hybrid encryption (see Section 2.4), a natural

[1] A user who shares one of her credentials once gives him the ability to use all of her credentials, thus taking over her identity.

question is how can one construct KDM-secure hybrid encryption? This question was open for a number of years, and is solved by the work detailed in this chapter.

We describe how to achieve KDM chosen-ciphertext security for hybrid encryption, prescribing the sufficient conditions for the asymmetric key encapsulation mechanism (KEM) and the symmetric data encapsulation mechanism (DEM). Precisely, if the KEM is $\mu\mathsf{OW}\text{-}\mathsf{CCA}$ (where the μ indicates multiple targets for inversion), the DEM is $\mathsf{IND}\text{-}\mathsf{CCA}$ and the key derivation function is modelled as a random oracle, then the resulting hybrid construction is $\mathsf{IND}\text{-}\mathsf{KDM}\text{-}\mathsf{CCA}$ secure. Components of the proof are technically challenging and non-standard. The chapter will also give extensions of known separation results in KDM and circular security, demonstrating that hybrid encryption does not always provide leverage in resisting key-dependent message attacks.

3.2 Related Literature

The foundations of research in KDM security were laid by two (concurrent and independent) papers. The work of Camenisch and Lysyanskaya [83] looked at n-circular security for public key encryption: an adversary attempts to distinguish $\mathsf{Enc}_{\mathrm{pk}_1}(\mathrm{sk}_2), \mathsf{Enc}_{\mathrm{pk}_2}(\mathrm{sk}_3), \ldots, \mathsf{Enc}_{\mathrm{pk}_{n-1}}(\mathrm{sk}_n), \mathsf{Enc}_{\mathrm{pk}_n}(\mathrm{sk}_1)$, as well as encryptions of chosen messages, from encryptions of zero (n is the total number of key pairs in the system). This is a strictly weaker notion than the one given by Black et al. [65] that allows arbitrary functions on the decryption key (with a 'length-regularity' restriction that ensures the length of the output function does not depend on the secret key to stop trivial attacks). The main focus for Camenisch and Lysyanskaya was an anonymous credential system: encrypting different secret keys with one another creates an all-or-nothing property, discouraging users from transferring their individual credentials.

Constructing KDM-secure Public Key Encryption. Creating schemes that are KDM secure in the standard model for meaningful function classes posed a challenge for a number of years after the definitions were presented [83, 65]. The schemes detailed in this section largely use hardness assumptions and techniques generally associated with public key encryption, so schemes are in the PKE setting unless stated otherwise. Hofheinz and Unruh [132] gave a standard model construction that considered a limited class of functions that in particular did not include key cycles.

Boneh, Halevi, Hamburg and Ostrovsky (BHHO) at Crypto 2008 [72] gave the first IND-KDM-CPA-secure construction (see Section 3.3.1 for the formal definition) for the rich affine function class, given in the standard model and under the decisional Diffie-Hellman assumption—more details of this scheme can be found in Section 3.4.2. The authors also showed that one-way security does not imply circular security. Boneh et al.'s scheme was improved upon by Camenisch, Chandran and Shoup [82] who gave a scheme secure against IND-KDM-CCA attacks (see Section 3.3.2 for definition). Barak et al. [25] wanted to admit a larger function class, and presented a construction that is (IND-KDM-CPA) secure in their *bounded-KDM* model, meaning KDM security for all Boolean circuits of bounded circuit size. Soon after, Malkin et al. [159] gave an IND-KDM-CPA secure scheme relying on LWE/LPN that was considerably more efficient than prior efforts: the ciphertext size depends only on the degree bound for the polynomial (other parameters for the scheme are independent of the function or the number of users). Hofheinz [129] achieved compact (constant size) ciphertexts using lossy algebraic filters to present a scheme that is CIRC-CCA secure, and Lu et al. [157] expanded this by giving a construction with compact ciphertexts that achieves full IND-KDM-CCA security for affine functions using RKA-secure authenticated encryption. A number of other schemes [74, 75, 13] gave other positive results expanding the admissable function classes, and more details of some of the schemes in this line of work can be found in the next chapter. Wee [187] gave a general framework that captures a number of these constructions [72, 74, 75], linking the pursuit of KDM-secure encryption schemes with (homomorphic) smooth projective hashing.

Symmetric Encryption. In 2007 Halevi and Krawczyk [125] gave an analysis of deterministic symmetric schemes, and detailed a PRF even when inputs depend in an arbitrary (but *a priori* known) manner on the key. In particular they show that if one constructs an SKE scheme using such a key-dependent input-secure PRF, the underlying PRF's resilience against key-dependent inputs/messages will be passed on to the SKE scheme. Halevi and Krawczyk also describe how the IEEE P1619 standard group (Security in Storage Working Group), when working on a standard for sector-level encryption, were informed that the Windows Vista disk encryption implementation (in some situations) stored an encryption of its own secret key and consequently decided to encourage use of the XE/XEX block ciphers developed by Rogaway [176] rather than the tweakable block cipher of Liskov, Rivest and Wagner [156]. Furthering this work, Bellare, Cash and Keelveedhi [30] showed how to create PRPs that are KDM secure in the standard model, assuming PRP-CCA security of the underlying (tweakable) cipher.

Soundness and Formal Methods. Abadi and Rogaway [3] considered soundness of encryption, bridging the gap between the formal modelling approach (in terms of symbolic expressions) and the computational model (regarding adversaries in terms of complexity and probabilities), and described how security reductions go through if key-cycles are disallowed. Laud and Corin [152], Adão et al. [7], Küsters and Tuengerthal [151] and Backes et al. [22, 23] furthered the understanding of key-cycles in formal proofs, linking the symbolic approach taken by the formal methods community with the computational approach. Comon-Lundh and Cortier [91] survey results in computational soundness and discuss the importance of the assumption of disallowing key cycles.

Impossibility Results. A number of works have given negative results: Haitner and Holenstein [124] showed the impossibility of obtaining KDM security based on standard assumptions and using standard techniques; in particular there exists no reduction from a KDM-secure encryption scheme to any cryptographic assumption if the reduction regards the adversary and the function as black boxes (for meaningful function classes). Acar et al. [6] looked at the link between circular-secure encryption and cryptographic agility (meaning the ability of individually secure schemes to share a key), and presented an example of an IND-CPA secure PKE scheme (under SXDH) that is not 2-circular secure, and give a distinguishing attack—Bishop et al. [62] give counterexamples of the same result using DLIN and LWE. Cash, Green and Hohenberger [87] presented n-weak circular security where instead of distinguishing a key cycle from an encryption of zero, the adversary is handed an encrypted cycle $\mathsf{Enc}_{\mathrm{pk}_1}(\mathrm{sk}_2), \mathsf{Enc}_{\mathrm{pk}_2}(\mathrm{sk}_3), \ldots, \mathsf{Enc}_{\mathrm{pk}_{n-1}}(\mathrm{sk}_n), \mathsf{Enc}_{\mathrm{pk}_n}(\mathrm{sk}_1)$ and then goes on to play the IND-CPA or IND-CCA game as normal.[2] The authors go on to show that there exists an IND-CPA-secure PKE scheme that is not 2-weak circular secure and the adversary can in fact recover the secret keys of both users. See Section 3.6 for details of extending the negative results of Acar et al. and Cash et al. to hybrid encryption. Koppula et al. [147] use indistinguishability obfuscation to show that if indistinguishability obfuscation (iO) exists, then there exists a scheme that is IND-CPA secure but not n-circular secure for any $n \geq 1$.

Other Approaches. To give an indication of the breadth of research in the area, KDM security has also been considered in such diverse settings as identity-based encryption (IBE) [10, 114], authenticated and misuse-resistant encryption [42], programmable encryption [185, 21], bit encryption [177], point obfuscation [86] and fully homomorphic encryption [117, 76]: Gentry showed how a fully homomorphic scheme for limited depth circuits can be 'bootstrapped' to work for

[2] This disallows the Acar et al. attack. Clearly the adversary must not be able to submit any of the key cycle to the decryption oracle.

circuits of arbitrary depth, if the original system is 1-circular secure and can compute its own decryption circuit. Bellare, Meiklejohn and Thomson [45] develop a definition of KDM-secure storage which incorporates both privacy and authenticity, and the authors seperately consider RKA-secure signatures (see Chapter 4). Backes et al. [20] show that the OAEP encryption scheme is IND-KDM-CPA secure in the random oracle model; we regard our work as advancing this effort to analyse KDM security of deployed encryption schemes.

3.3 Security Models and Formalism

Notions for KDM security described in this chapter are relative to a function class Ψ, referred to as *plaintext-construction functions* by Black et al. [65], which stipulates that the adversary is bound to asking only queries $\psi \in \Psi$. For instance, if Ψ corresponds to the set of all constant functions, notions equivalent to IND-CPA and IND-CCA emerge. The challenge is to devise schemes that can be proven secure for a class Ψ that is as large as possible. However, to achieve any interesting results we need to restrict this set to functions $\psi(\mathbf{k})$ that act on the (vector of) decryption key(s) in the system $\mathbf{sk}$ to those functions where the length of the output $|\psi(\mathbf{sk})|$ does not depend on $\mathbf{sk}$ (to ensure that the adversary cannot trivially win by simply looking at the length of the ciphertext). This restriction is often referred to as *length regularity*. Black et al. formally regarded ψ modelled as an algorithm in some fixed RAM model, however for the rest of this chapter we will instead regard ψ as an arithmetic or Boolean circuit, which will imply that the output length of ψ is fixed and automatically independent of its input.

The work of Black et al. introduced a formal definition for key-dependent message security for chosen-plaintext attacks (henceforth IND-KDM-CPA), and their discussion considers the multi-user symmetric case. Simply put, an adversary submits as challenge a function ψ and receives an encryption of either $\psi(\mathbf{sk})$ or of a dummy message, which is $0^{|\psi(\mathbf{sk})|}$ if the message space is bitstrings of arbitrary length, or a random message from the message space if the message space is some fixed, finite space such as $\mathbb{Z}_p$. Camenisch et al. [82] introduced IND-KDM-CCA security, a natural blend between IND-CCA and IND-KDM-CPA; this is the version we will focus on later in the chapter. For our purposes there can be multiple keys in the system and, contrary to standard IND-CPA security, for the IND-KDM security notions it is not possible to reduce (e.g. by hybrid argument) to a single key or single query, since the reduction simply cannot simulate the KDM queries without the decryption key.

Our syntax also differs from that of Black et al. as we make a distinction between parameter and key generation; this approach is somewhat common in the multi-user PKE setting that we consider. Since ψ implements a function from a Cartesian product of secret key spaces to the message space and these spaces can depend on the parameter generation (e.g. which cyclic group is used for DLP-based systems), the security experiment incorporates a check that ψ is syntactically valid (however we will henceforth drop explicit mention of it).

The majority of this chapter will consider definitions and results in the random oracle model (ROM). When we are working in the ROM, the adversary $\mathcal{A}$ can of course access the random oracle directly but in addition, the KDM function $\psi \in \Psi$ may make a (number of) call(s) to the random oracle.

We now give formal definitions for key-dependent message security, and explain the choices made. We will present the chosen-plaintext version in the context of symmetric encryption because we will need it in Chapter 4, and the active IND-KDM-CCA definition for public key encryption as we will need that later in this chapter.

3.3.1 IND-KDM-CPA Security of Symmetric Encryption

Definition 3.1 details the multi-key variant of the IND-KDM-CPA security definition, in the context of symmetric encryption. Restricting Ψ to the set of constant functions yields the standard 'real-or-fake' flavour of the IND-CPA game presented in Section 2.2.1, where the message space is assumed to be arbitrary-length bitstrings $\{0,1\}^*$. In Chapter 4 we will require an alternative notion: the message space is fixed a priori, with the implicit assumption that messages are of fixed length. In this scenario, the security experiment picks $\mathrm{m}_0 \leftarrow \mathcal{M}$ at the start and the line $\mathrm{m}_0 \leftarrow 0^{|\mathrm{m}_1|}$ is removed. These notions are equivalent when the message space is bitstrings of fixed length.

```
$\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]\text{-}b}_{\Sigma,\,\mathcal{A}}(\lambda)$ :
    $\mathrm{pp} \leftarrow \mathsf{Pg}(\lambda)$
    $t \leftarrow 0$
    $b' \leftarrow \mathcal{A}^{\mathsf{New},\,\mathsf{LR}_b}(\mathrm{pp})$
    return $b'$

$\mathsf{New}()$:
    $t \leftarrow t + 1$
    $\mathrm{k}_t \leftarrow \mathsf{Kg}(\mathrm{pp})$
    Append $\mathrm{k}_t$ to $\mathbf{k}$
    return $\perp$
```

```
$\mathsf{LR}_b(\psi, i)$ :
    if $(\psi, i) \notin \Psi(\mathrm{pp}, t)$ then
        return $\lightning$
    $\mathrm{m}_1 \leftarrow \psi(\mathbf{k})$
    $\mathrm{m}_0 \leftarrow 0^{|\mathrm{m}_1|}$
    $\mathrm{c} \leftarrow \mathsf{E}_{\mathrm{k}_i}(\mathrm{m}_b)$
    return $\mathrm{c}$
```

***Figure 3.1:** The general* IND-KDM-CPA *experiment for symmetric encryption.*

Definition 3.1 (IND-KDM-CPA Security for Symmetric Encryption)**.** *Let* $\Sigma = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{E}, \mathsf{D})$ *be a* *symmetric encryption scheme (with security parameter λ). Let Ψ be a collection of circuits that map a (number of) key(s) to an element in the message space. Then the* $\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]$ *advantage of an adversary $\mathcal{A}$ against Σ relative to key-dependent message attacks for circuit class Ψ is defined by*

$$\mathbf{Adv}_{\Sigma,\,\mathcal{A}}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]}(\lambda) \overset{def}{=} \left| \sum_{b\in\{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}_{\Sigma,\,\mathcal{A}}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]\text{-}b}(\lambda) = 1\right]\right|$$

where the experiment $\mathbf{Exp}_{\Sigma,\,\mathcal{A}}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]\text{-}b}(\lambda)$ is given in Fig. 3.1.

This definition attempts to be as general as possible: the multi-key scenario is modelled by having a vector $\mathbf{k}$ of private keys upon which the KDM function ψ can act. This means that when adversary $\mathcal{A}$ makes a left-or-right query it must specify a key index along with ψ to receive an encryption $\mathsf{E}_{\mathrm{k}_i}(\mathrm{m}_b)$.

3.3.2 IND-KDM-CCA Security of Public Key Encryption

Introducing chosen-ciphertext queries invokes serious challenges in constructing secure schemes in the key-dependent message framework, and Definition 3.2 details IND-KDM-CCA for public key encryption. Note that some authors [25, 87, 129, 146] give a definition that is additionally parameterised by the number of public/secret key pairs in the system (rather than allowing the adversary to create an arbitrary number of key pairs), and consequently refer to notions as n-KDM-CCA or similar. Again, if we desire a definition suitable for schemes with message spaces that are not bitstrings then we choose m_0 randomly from the message space accordingly.

```
$\mathbf{Exp}_{\Pi,\,\mathcal{A}}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Psi]\text{-}b}(\lambda)$ :
  $\mathrm{pp} \leftarrow \mathsf{PGen}(\lambda)$
  $t \leftarrow 0$
  $\mathrm{FL} \leftarrow \emptyset$
  $b' \leftarrow \mathcal{A}^{\mathsf{New},\,\mathsf{LR}_b,\,\mathsf{Dec}}(\mathrm{pp})$
  return $b'$

$\mathsf{New}()$:
  $t \leftarrow t+1$
  $(\mathrm{pk}_t, \mathrm{sk}_t) \leftarrow \mathsf{KGen}(\mathrm{pp})$
  Append $\mathrm{sk}_t$ to $\mathbf{sk}$
  return $\mathrm{pk}_t$

$\mathsf{LR}_b(\psi, i)$ :
  if $(\psi, i) \notin \Psi(\mathrm{pp}, \mathbf{pk}, t)$ then
    return $\lightning$
  $\mathrm{m}_1 \leftarrow \psi(\mathbf{sk})$
  $\mathrm{m}_0 \leftarrow 0^{|\mathrm{m}_1|}$
  $\mathrm{c} \leftarrow \mathsf{Enc}_{\mathrm{pk}_i}(\mathrm{m}_b)$
  $\mathrm{FL} \leftarrow \mathrm{FL} \cup \{(\mathrm{c}, i)\}$
  return $\mathrm{c}$

$\mathsf{Dec}(\mathrm{c}, i)$ :
  if $(\mathrm{c}, i) \in \mathrm{FL}$ then
    return $\perp$
  $\mathrm{m} \leftarrow \mathsf{Dec}_{\mathrm{sk}_i}(\mathrm{c})$
  return $\mathrm{m}$
```

Figure 3.2: *The general* IND-KDM-CCA *experiment for public key encryption. Removing oracle* Dec *yields the* IND-KDM-CPA *experiment.*

Definition 3.2 (IND-KDM-CCA Security for Public Key Encryption). *Let* $\Pi = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public key encryption scheme (with security parameter λ). Let Ψ be a collection of circuits that map a (number of) secret key(s) to an element in the message space. Then the* $\mathsf{IND\text{-}KDM\text{-}atk}[\Psi]$ *advantage of an adversary $\mathcal{A}$ against Π relative to key-dependent message attacks for circuit class Ψ and* $\mathsf{atk} \in \{\mathsf{CPA}, \mathsf{CCA}\}$ *is defined by*

$$\mathbf{Adv}_{\Pi,\,\mathcal{A}}^{\mathsf{IND\text{-}KDM\text{-}atk}[\Psi]}(\lambda) \overset{def}{=} \left| \sum_{b\in\{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}_{\Pi,\,\mathcal{A}}^{\mathsf{IND\text{-}KDM\text{-}atk}[\Psi]\text{-}b}(\lambda) = 1\right] \right|$$

where the experiment $\mathbf{Exp}_{\Pi,\,\mathcal{A}}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Psi]\text{-}b}(\lambda)$ *is given in Fig. 3.2, and removing the decryption oracle yields experiment* $\mathbf{Exp}_{\Pi,\,\mathcal{A}}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]\text{-}b}(\lambda)$.

As in the IND-CCA setting we require a forbidden list FL that tracks the ciphertexts output by the LR oracle and the associated key index, because the adversary is allowed to ask for decryptions of such ciphertexts under all other secret keys.

3.3.3 Circular Security and Alternative Notions

A specific case of $\mathsf{IND\text{-}KDM\text{-}atk}[\Psi]$ security for public key encryption is n-circular security, which is particularly useful in anonymous credential systems as described by Camenisch and Lysyanskaya [83]. In this setting, the set of key-dependent message functions is restricted to

$$\Psi := \{\psi_s : \psi_s(\mathbf{sk}) = \mathrm{sk}_s\}_{s\in[n]} \cup \{\psi_{\mathrm{m}} : \psi_{\mathrm{m}}(\mathbf{sk}) = \mathrm{m}\}_{\mathrm{m}\in\mathcal{M}}$$

meaning selecting specific secret keys or messages in the message space. Some authors have referred to this as CYC or n-CYC security.

Backes et al. [23, 20] used a different framework where the adversary does not have direct access to the results of encryptions but instead can instruct the system to create keys, perform encryptions and other operations etc. with the subsequent capacity to learn part of the system's state. This is a potentially stronger framework (see also Dent [102]) reminiscent of work on cryptographic APIs (e.g. [149]). The PROG-KDM security definition provided by Unruh [185] also allows for corruptions, but it is not easy to see how it can be satisfied in a non-programmable random oracle setting (let alone the standard model). In the work in this chapter, we will contend ourselves with the easier (and weaker) notion based on the original work by Black et al.

3.4 Realising KDM Security

This section will give an overview of the most important results in achieving KDM security. It took a number of years from the early definitions of KDM and circular security to the realisation of schemes secure in the framework, and the consequent flurry of improvements and extensions.

3.4.1 Black et al.'s Random Oracle Scheme

As a proof of concept, Black et al. [65] gave a symmetric construction **ver** which is IND-KDM-CPA[Ψ] secure for all length-regular Ψ in the random oracle model. The construction $\Sigma_{\mathbf{ver}}[l] = (\mathsf{Kg}, \mathsf{E}, \mathsf{D})$ for $\mathcal{M} = \{0,1\}^y$ uses a hash function $\mathsf{H} : \{0,1\}^{2l} \to \{0,1\}^y$. The scheme, described in Fig. 3.3, is parameterised by the keylength l and has no parameter generation.

```
$\underline{\Sigma_{\mathbf{ver}}.\mathsf{Kg}(pp)}$ :
    $k \xleftarrow{\$} \{0,1\}^l$
    return $k$

$\underline{\Sigma_{\mathbf{ver}}.\mathsf{E}(m,k)}$ :
    $R \xleftarrow{\$} \{0,1\}^l$
    $c \leftarrow R||(\mathsf{H}(k||R) \oplus m)$
    return $c$
```

```
$\underline{\Sigma_{\mathbf{ver}}.\mathsf{D}(c,k)}$ :
    if $|c| < l$ then
        return $\perp$
    $R \leftarrow c[1 \ldots l]$
    $\tilde{c} \leftarrow c[(l+1) \ldots]$
    $m \leftarrow \mathsf{H}(k||R) \oplus \tilde{c}$
    return $m$
```

Figure 3.3: *Black et al.'s* KDM-CPA *secure symmetric encryption scheme* $\Sigma_{\mathbf{ver}}$.

The authors proved the scheme secure for all length-regular KDM functions, in the random oracle model. In addition, they presented a public key encryption scheme $\Pi_{\mathbf{VER}}[\mathcal{F}]$ parameterised by a trapdoor permutation generator $\mathcal{F}$, a scheme that was initially suggested by Bellare and Rogaway in their seminal 'Random Oracles are Practical' paper [48]. The encryption algorithm, parameterised by a function f chosen by $\mathcal{F}$, computes $c \leftarrow f(R)||(\mathsf{H}(R) \oplus m)$ where R is the randomness used for encryption and H is the random oracle. Clearly this scheme neatly fits into the hybrid framework, with protokey R and ephemeral key $\mathsf{H}(R)$. The authors stated optimistically:

> "One expects that $\Pi_{\mathbf{VER}}[\mathcal{F}]$ is a KDM-secure encryption scheme if $\mathcal{F}$ is a secure trapdoor permutation. At the time of this writing, we have not written up a proof of this. The above is only one natural construction; others would seem to work. In [83] Camenisch and Lysyanskaya give a different scheme which they claim is 'circular secure' (in the RO model), a notion that they define. One would expect their scheme to be KDM secure as well, though we have not written up a proof of this."
>
> *Black, Rogaway and Shrimpton [65], 2002.*

It is this discussion that we shall revisit later in the chapter.

3.4.2 Boneh et al.'s IND-KDM-CPA-Secure Scheme

At CRYPTO 2008, Boneh, Halevi, Hamburg and Ostrovsky (BHHO) [72] presented the first scheme that was provably IND-KDM-CPA secure in the standard model. The motivating scenario is closer to the circular security in anonymous credential systems as presented by Camenisch and Lysyanskaya [83], and the breakthrough component of the work is an inherent ability of all parties to create encryption cliques. The scheme is detailed in Fig. 3.4, maintaining notation where possible. The scheme is secure[3] with respect to affine functions over the group used, a function class that notably includes the case of circular security.

```
$\Pi_{\mathsf{BHHO}}.\mathsf{PGen}(\lambda)$ :
    $g \xleftarrow{\$} \mathbb{G} \setminus \{1\}$
    $pp \leftarrow (\mathbb{G}, g)$
    return pp

$\Pi_{\mathsf{BHHO}}.\mathsf{KGen}(pp)$ :
    $l \leftarrow \lceil 3 \log_2 q \rceil$
    $g_1 \dots g_l \xleftarrow{\$} \mathbb{G}$
    $(s_1, \dots, s_l) \xleftarrow{\$} \{0,1\}^l$
    $h \leftarrow (g_1^{s_1} \cdots g_l^{s_l})^{-1}$
    $pk \leftarrow (g_1, \dots, g_l, h)$
    $sk \leftarrow (g^{s_1}, \dots, g^{s_l})$
    return $(pk, sk)$
```

```
$\Pi_{\mathsf{BHHO}}.\mathsf{Enc}(m, pk)$ :
    $r \xleftarrow{\$} \mathbb{Z}_q$
    $c \leftarrow (g_1^r, \dots, g_l^r, h^r \cdot m)$
    return c

$\Pi_{\mathsf{BHHO}}.\mathsf{Dec}(c, sk)$ :
    $(x_1, \dots, x_l, y) \leftarrow c$
    $(v_1, \dots, v_l) \leftarrow sk$
    for $i = 1 \dots l$
        if $v_i = 1$ then
            $s_i \leftarrow 0$
        else
            $s_i \leftarrow 1$
    $m \leftarrow y \cdot \prod_{i \in [l]} x_i^{s_i}$
    return m
```

Figure 3.4: *Boneh et al.'s* KDM-CPA *secure encryption scheme* Π_{BHHO}

The details of this scheme are covered in more detail in Section 4.4.4.1, where we will use it as a basis for constructing a scheme that is both KDM and RKA secure. For each $1 \le i \le l$, the $(l+1)$-vector $(1 \dots 1 g 1 \dots 1)$, where g is in position i and 1 everywhere else, will decrypt to g^{s_i} so any party can generate encryptions of the secret key without knowledge of it. Since the encryption wouldn't actually output a ciphertext of this form, the authors move to an expanded variant of their scheme where each $(l+1)$-vector is a valid ciphertext and the simulator can create a random encryption of the secret key. Each ciphertext and secret key contains $l \leftarrow \lceil 3 \log_2 q \rceil$ group elements to ensure that the secret keys have sufficient entropy so that that during their proof, the public key elements can be replaced by an item that is $\frac{1}{q}$-uniform. The authors show

[3] In their paper Boneh et al. in fact define KDM security in terms of an adversary that attempts to distinguish $\psi(\mathbf{sk})$ from $0^{|\psi(\mathbf{sk})|}$ however their construction and proofs refer to a message space which is an algebraic group.

that their scheme is IND-KDM-CPA secure under the general k-linear assumption (recall that $k = 1$ corresponds to DDH and $k = 2$ corresponds to the DLIN assumption); the only change required is $l \leftarrow \lceil (k + 2) \log_2 \mathrm{q} \rceil$. The feature of the BHHO scheme that allows all parties to create encryptions of the secret key did allow the authors to prove it IND-KDM-CPA secure, however it also meant that this scheme could not be IND-KDM-CCA secure since an adversary could trivially create useful decryption queries.

3.4.3 Camenisch et al.'s IND-KDM-CCA-Secure Scheme

Just a year after Boneh et al.'s breakthrough result, Camenisch, Chandran and Shoup [82] gave a construction that turns any IND-KDM-CPA-secure scheme into an IND-KDM-CCA-secure scheme using strongly one-time secure signatures and NIZKs. The work employs the Naor-Yung *double encryption* paradigm [163]: encrypt message m with the IND-KDM-CPA-secure scheme and also using an IND-CCA-secure scheme that supports labels and place the verification key of the signature scheme in the label. Then, attach as an authentication tag a NIZK proof that says that either the encryptor has created consistent encryptions, or they know a signature for the ciphertext; the proof for consistency uses Groth-Sahai proofs [123].

3.5 Achieving KDM Security of Hybrid Encryption

The ubiquitous deployment of hybrid encryption makes it an excellent candidate for security analysis, and before the work in this chapter there had been no thorough analysis of KDM security in the context of hybrid encryption. As mentioned in Section 3.4.1, Black et al. indicated that they believed a straightforward hybrid scheme to be secure in the random oracle model, yet did not provide a proof. As it turns out, the proof of a general statement is challenging and requires non-standard components. The standard approach for proofs regarding hybrid encryption is to decouple the key encapsulated in the KEM and the one used in the DEM (triggering a 'bad' event where the adversary has managed to make the two align), and then argue that the DEM is indistinguishable as it is run with a random key. In the KDM scenario this strategy no longer works, as the reduction cannot simulate the adversary's key-dependent queries, so our approach is to move the analysis of this bad event from the key-dependent side to the key-independent segment of the analysis.

We will first introduce some tools that will be employed during the proof, and then give a deeper intuition of the proof techniques.

Subsequently to this work, Chang, Xue and Zhang [88] showed that the Kurosawa-Desmedt hybrid encryption scheme is KDM-CCA secure in the standard model with respect to the limited KDM function ensemble introduced by Qin et al. [171]. This ensemble allows the adversary to access encryptions of the difference between two secret keys, however it does not include encryptions of individual secret keys.

3.5.1 Tools

We now describe two security notions that will be necessary further on: first a notion for DEMs that is equivalent to IND-CCA, and secondly a formal definition of KDM security for hybrid encryption in the random oracle model. For the remainder of this chapter we will assume that the message space that we are dealing with is infinite bitstrings.

3.5.1.1 Restricted KDM Security of the DEM

We introduce a security notion for DEMs called IND-PKDM-CCA ('Prior-KDM'), where an adversary's KDM capability is restricted to (encryptions of) functions of all 'past' DEM keys in the system. The formal security game for IND-PKDM-CCA is depicted in Fig. 3.5. We will soon show that this restricted form of KDM attacks is not all that useful to an attacker—the notion is in fact equivalent to IND-CCA security for DEMs (cf. Def. 2.4). It is important to emphasise here that this definition is somewhat of an artefact of the forthcoming proof, and there is little to suggest that it would be useful in a wider context. Intuitively, the adversary has access to a left-or-right oracle, and when she submits a key index i and a function ϑ, receives an encryption under k_i of either the function applied to all 'past' DEM keys $\mathrm{m}_1 \leftarrow \vartheta(\mathrm{k}_1, \ldots, \mathrm{k}_{i-1})$ or a dummy message $\mathrm{m}_0 \leftarrow 0^{|\mathrm{m}_1|}$. We write $\vartheta(\mathrm{k}^{i-1})$ to represent $\vartheta(\mathrm{k}_1, \ldots, \mathrm{k}_{i-1})$ for ease of exposition.

Definition 3.3. *Let* $\mathsf{DEM} = (\mathsf{DEM.Pg}, \mathsf{DEM.Kg}, \mathsf{DEM.E}, \mathsf{DEM.D})$ *be a data encapsulation mechanism. Then the* IND-PKDM-CCA *advantage of an adversary* $\mathcal{A}$ *against* DEM *is defined by*

$$\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}}(\lambda) \overset{def}{=} \left| \sum_{b \in \{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}}b}_{\mathsf{DEM},\,\mathcal{A}}(\lambda) = 1\right] \right|$$

where the experiment $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}}b}_{\mathsf{DEM},\,\mathcal{A}}(\lambda)$ *is given in Fig. 2.8.*

We now show that IND-PKDM-CCA security is equivalent to IND-CCA security for DEMs. That IND-PKDM-CCA security implies IND-CCA security follows from standard relations between different formulations of IND-CCA security, plus the fact that a non-key-dependent message can be queried (in the KDM world) by using a constant function.

```
$\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}}b}_{\mathsf{DEM},\,\mathcal{A}}(\lambda)$ :
    $\mathrm{pp} \leftarrow \mathsf{DEM.Pg}(\lambda)$
    $t \leftarrow 0$
    $\mathrm{FL} \leftarrow \emptyset$
    $b' \leftarrow \mathcal{A}^{\mathsf{New},\,\mathsf{LR}_b,\,\mathsf{D}}(\mathrm{pp})$
    return $b'$

$\mathsf{New}()$:
    $t \leftarrow t + 1$
    $\mathrm{k}_t \leftarrow \mathsf{DEM.Kg}(\mathrm{pp})$
    return $t$
```

```
$\mathsf{LR}_b(\vartheta, i)$ :
    if $i \notin [t]$ then
        return $\lightning$
    $\mathrm{m}_1 \leftarrow \vartheta(\mathrm{k}^{i-1})$
    $\mathrm{m}_0 \leftarrow 0^{|\mathrm{m}_1|}$
    $\mathrm{C} \leftarrow \mathsf{DEM.E}_{\mathrm{k}_i}(\mathrm{m}_b)$
    $\mathrm{FL} \leftarrow \mathrm{FL} \cup \{(\mathrm{C}, i)\}$
    return $\mathrm{C}$

$\mathsf{D}(\mathrm{C}, i)$:
    if $(\mathrm{C}, i) \in \mathrm{FL}$ then
        return $\lightning$
    $\mathrm{m} \leftarrow \mathsf{DEM.D}_{\mathrm{k}_i}(\mathrm{C})$
    return $\mathrm{m}$
```

Figure 3.5: *The* IND-PKDM-CCA *security experiment for data encapsulation mechanism* DEM. *Here* $\vartheta(\mathrm{k}^{i-1})$ *indicates that the function* ϑ *can depend on all keys in range* $\{\mathrm{k}_1, ..., \mathrm{k}_{i-1}\}$.

To see that IND-CCA security for DEMs implies the IND-PKDM-CCA notion we employ a hybrid argument.

Theorem 3.1. *Let* DEM *be a data encapsulation mechanism. Then for adversary $\mathcal{A}$, there exists an algorithm $\mathcal{A}_1$ of comparable computational complexity such that*

$$\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}}(\lambda) \leq \mathrm{n} \cdot \mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}_1}(\lambda) \; .$$

where λ is the security parameter and n *is the number of keys in the DEM system.*

Proof:

We seek a contradiction, by assuming that DEM is IND-CCA secure but not IND-PKDM-CCA secure, so there exists an algorithm $\mathcal{A}$ that breaks IND-PKDM-CCA. If we have $\mathrm{n} + 1$ keys in the system, then we have $\mathrm{n} + 1$ hybrid experiments Hyb-j as described Fig. 3.6 (for $j \in \{0, ..., \mathrm{n}\}$).

In the $j = 0$ hybrid we have $i > 0$ $(\forall i)$ so this refers to $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}0}}_{\mathsf{DEM},\,\mathcal{A}}(\lambda)$, and for $j = \mathrm{n}$ we have $i \leq j$ $(\forall i)$ which always returns an encryption of m_1 corresponding to $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}1}}_{\mathsf{DEM},\,\mathcal{A}}(\lambda)$, which gives rise to equation (3.1).

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}}(\lambda) &\overset{def}{=} \left| \sum_{b \in \{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}}b}_{\mathsf{DEM},\,\mathcal{A}}(\lambda) = 1\right] \right| \\
&= \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{Hyb\text{-}n}}_{\mathsf{DEM},\,\mathcal{A}}(\lambda) = 1\right] - \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{Hyb\text{-}0}}_{\mathsf{DEM},\,\mathcal{A}}(\lambda) = 1\right] && (3.1) \\
&\leq \mathrm{n} \cdot \mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}_1}(\lambda) && (3.2)
\end{aligned}$$

```
$\mathbf{Exp}^{\mathsf{Hyb}\text{-}j}_{\mathsf{DEM},\,\mathcal{A}}(\lambda)$ :
    $\mathrm{pp} \leftarrow \mathsf{DEM.Pg}(\lambda)$
    $t \leftarrow 0$
    $\mathrm{FL} \leftarrow \emptyset$
    $b' \leftarrow \mathcal{A}^{\mathsf{New},\, \mathsf{LR}_b,\, \mathsf{D}}(\mathrm{pp})$
    return $b'$

$\mathsf{New}()$:
    $t \leftarrow t + 1$
    $\mathrm{k}_t \leftarrow \mathsf{DEM.Kg}(\mathrm{pp})$
    return $t$
```

```
$\mathsf{LR}_b(\vartheta, i)$ :
    if $i \notin [t]$ then
        return $\lightning$
    $\mathrm{m}_1 \leftarrow \vartheta(\mathrm{k}^{i-1})$
    $\mathrm{m}_0 \leftarrow 0^{|\mathrm{m}_1|}$
    if $i \leq j$ then
        $\mathrm{C} \leftarrow \mathsf{DEM.E}_{\mathrm{k}_i}(\mathrm{m}_1)$
    else
        $\mathrm{C} \leftarrow \mathsf{DEM.E}_{\mathrm{k}_i}(\mathrm{m}_0)$
    $\mathrm{FL} \leftarrow \mathrm{FL} \cup \{(\mathrm{C}, i)\}$
    return $\mathrm{C}$

$\mathsf{D}(\mathrm{C}, i)$:
    if $(\mathrm{C}, i) \in \mathrm{FL}$ then
        return $\lightning$
    $\mathrm{m} \leftarrow \mathsf{DEM.D}_{\mathrm{k}_i}(\mathrm{C})$
    return $\mathrm{m}$
```

Figure 3.6: *Hybrid games* Hyb-*j for* DEM *to prove Thm. 3.1. Again* $\vartheta(\mathrm{k}^{i-1})$ *indicates that the function* ϑ *can depend on all keys in range* $\{\mathrm{k}_1, ..., \mathrm{k}_{i-1}\}$.

Since we made the assumption that $\mathcal{A}$ breaks IND-PKDM-CCA, $\mathcal{A}$ can distinguish at least one gap between two hybrids in the sum. If we assume that $\mathcal{A}$ can distinguish between Hyb-j^* and Hyb-(j^*-1) for some j^*, then we need to build a reduction $\mathcal{A}_1$ that can simulate all queries, and thus can compute m_1 (and subsequently m_0) by itself, and feed into the IND-CCA oracle and win the game. That is to say, $\mathcal{A}$'s advantage in distinguishing between Hyb-j^* and Hyb-(j^*-1) is greater than $1/\mathrm{n} \cdot \mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}\text{-}b}_{\mathsf{DEM},\,\mathcal{A}}(\lambda)$. Utilising this fact, we create an algorithm $\mathcal{A}_1$ attacking the IND-CCA property of DEM, as detailed in Fig. 3.7, to prove equation (3.2). The key point here is that the reduction $\mathcal{A}_1$ knows all but one key, namely k_{j^*}, and it plays its own IND-CCA game against this key. Reduction $\mathcal{A}_1$ creates all the keys except for k_{j^*}, meaning that $\mathcal{A}_1$ needs to use a counter z to correctly respond to $\mathcal{A}$'s calls to New. For $i < j^*$ it is possible for the reduction to simlulate m_1 correctly using k^{i-1}. For $i > j^*$ we use the length regularity condition on ϑ to create the m_0 value to feed into the hybrids; note that in many other reductions and definitions in this thesis the 'real' message m_1 is already created so we can use $0^{|\mathrm{m}_1|}$ however this is not the case here so we abuse notation and take $|\vartheta|$ to mean the length of the output of ϑ. Since we know $\mathcal{A}$ can beat IND-PKDM-CCA for $i = j^*$, we make the input values to the IND-CCA game's LR oracle be the same as the m_1 and m_0 values used in the IND-PKDM-CCA game, ensuring that the reduction captures this correctly.

```
$\mathcal{A}_1$ playing $\mathbf{Exp}^{\mathsf{IND\text{-}CCA}\text{-}b}_{\mathsf{DEM},\mathcal{A}_1}(\lambda)$:
    $\mathrm{pp} \leftarrow \mathsf{DEM.Pg}(\lambda)$
    $z \leftarrow 0$
    for $i \in [\mathrm{n}] \setminus \{j^*\}$ do
        $\mathrm{k}_i \leftarrow \mathsf{DEM.Kg}(\mathrm{pp})$
    $b' \leftarrow \mathcal{A}^{\mathsf{New},\ \mathsf{LR}_b,\ \mathsf{D}}(\mathrm{pp})$
    return $b'$

$\mathsf{D}(\mathsf{C}, i)$:
    if $i \notin [\mathrm{n}]$ then
        return $\lightning$
    if $i = j^*$ then
        call $\mathrm{m} \leftarrow \mathsf{IND\text{-}CCA.D}(\mathsf{C})$
    else
        $\mathrm{m} \leftarrow \mathsf{DEM.D}_{\mathrm{k}_i}(\mathsf{C})$
    return $\mathrm{m}$
```

```
$\mathsf{New}()$:
    $z \leftarrow z + 1$
    return $z$

$\mathsf{LR}_b(\vartheta, i)$:
    if $i < j^*$ then
        $\mathrm{m}_1 \leftarrow \vartheta(\mathrm{k}^{i-1})$
        $\mathsf{C} \leftarrow \mathsf{DEM.E}_{\mathrm{k}_i}(\mathrm{m}_1)$
    if $i > j^*$ then
        $\mathrm{m}_0 \leftarrow |\vartheta|$
        $\mathsf{C} \leftarrow \mathsf{DEM.E}_{\mathrm{k}_i}(\mathrm{m}_0)$
    if $i = j^*$ then
        $\mathrm{m}_1 \leftarrow \vartheta(\mathrm{k}^{i-1})$
        $\mathrm{m}_0 \leftarrow 0^{|\mathrm{m}_1|}$
        call $\mathsf{C} \leftarrow \mathsf{IND\text{-}CCA.LR}(\mathrm{m}_0, \mathrm{m}_1)$
    return $\mathsf{C}$
```

Figure 3.7: *Description of reduction $\mathcal{A}_1$ used to prove (3.2). When $\mathcal{A}_1$ runs $\mathcal{A}$, it needs to create an environment $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA}\text{-}b}_{\mathsf{DEM},\mathcal{A}}$. The "**call** C" line details that $\mathcal{A}$ calls the* IND-CCA *oracle* LR, *receiving an encryption of* m_b *under the correct* IND-CCA *challenge key* k. *The line "**call** m" indicates that $\mathcal{A}$ calls the decryption oracle of the* IND-CCA *game on* C.

3.5.1.2 KDM Security for Hybrid Encryption

We now give a full definition for KDM security of a hybrid encryption scheme in the random oracle model. Fig. 3.8, which is an expanded version of Fig. 3.2, formalises this specification. This definition is in the random oracle model (the oracle $\mathsf{H}(\cdot)$ employs lazy sampling). We write '**if** $\exists \mathrm{h}_\mathrm{k}$ such that $(\mathrm{k}, \mathrm{h}_\mathrm{k}) \in \mathsf{H}_{\mathrm{list}}$' here to denote H doing a table lookup, but for ease of exposition we will write '**if** $(\mathrm{k}, \mathrm{h}_\mathrm{k}) \in \mathsf{H}_{\mathrm{list}}$' throughout the rest of this thesis; the column(s) in which H is searching will be clear from context. Note that the adversary can make a number of queries to the random oracle directly, and also it can provide functions that query the random oracle. This reflects the scenario where the adversary's key-dependent message functions Ψ are circuits with gates that call the random oracle. The adversary is allowed to query decryptions of the challenge ciphertexts under different public keys than the ones generated by LR_b, and this restriction is dealt with by the list FL. This definition assumes that the message space is bitstrings of arbitrary length, and that the functions $\psi \in \Psi$ are efficiently computable.

3.5.2 Achieving KDM-CCA Security of Hybrid Encryption

We are now in a position to state the main result of this chapter. Let $\mathsf{Hyb} = (\mathsf{Hyb.PGen}, \mathsf{Hyb.KGen}, \mathsf{Hyb.Enc}, \mathsf{Hyb.Enc})$ be a hybrid encryption scheme and let $\mathcal{A}$ be an adversary. In the hybrid setting there are two types of keys present: the private key of the KEM and

```
$\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA}\text{-}b}_{\mathsf{Hyb},\mathcal{A}}(\lambda)$:
  $\mathrm{pp} \leftarrow \mathsf{PGen}(\lambda)$
  $t \leftarrow 0$
  $\mathbf{sk} \leftarrow ()$
  $\mathsf{H}_{\mathrm{list}}, \mathrm{FL} \leftarrow \emptyset$
  $b' \leftarrow \mathcal{A}^{\mathsf{New},\mathsf{H},\mathsf{LR}^{\mathsf{H}}_{b},\mathsf{Dec}^{\mathsf{H}}}(\mathrm{pp})$
  return $b'$

$\mathsf{New}()$:
  $t \leftarrow t+1$
  $(\mathrm{pk}_t, \mathrm{sk}_t) \leftarrow \mathsf{KGen}(\mathrm{pp})$
  Append $\mathrm{sk}_t$ to $\mathbf{sk}$
  return $\mathrm{pk}_t$

$\mathsf{H}(\mathrm{k})$:
  if $\exists \mathrm{h_k}$ such that $(\mathrm{k}, \mathrm{h_k}) \in \mathsf{H}_{\mathrm{list}}$ then
    return $\mathrm{h_k}$
  else
    $\mathrm{h_k} \xleftarrow{\$} \{0,1\}^{\lambda}$
    $\mathsf{H}_{\mathrm{list}} \leftarrow \mathsf{H}_{\mathrm{list}} \cup \{(\mathrm{k}, \mathrm{h_k})\}$
    return $\mathrm{h_k}$
```

```
$\mathsf{LR}^{\mathsf{H}}_{b}(\psi, i)$:
  if $\psi \notin \Psi(\mathrm{pp}, \mathbf{pk}, i)$ then
    return $\lightning$
  $\mathrm{m}_1 \leftarrow \psi^{\mathsf{H}}(\mathbf{sk})$
  $\mathrm{m}_0 \leftarrow 0^{|\mathrm{m}_1|}$
  $(\omega, \mathrm{k}) \leftarrow \mathsf{KEM.Encap}_{\mathrm{pk}_i}()$
  $\mathrm{h_k} \leftarrow \mathsf{H}(\mathrm{k})$
  $\mathrm{C}_b \leftarrow \mathsf{DEM.E}_{\mathrm{h_k}}(\mathrm{m}_b)$
  $\mathrm{FL} \leftarrow \mathrm{FL} \cup \{(\omega, \mathrm{C}_b, i)\}$
  return $(\omega, \mathrm{C}_b)$

$\mathsf{Dec}^{\mathsf{H}}(\omega, \mathrm{C}, i)$:
  if $(\omega, \mathrm{C}, i) \in \mathrm{FL}$ then
    return $\lightning$
  $\mathrm{k} \leftarrow \mathsf{KEM.Decap}_{\mathrm{sk}_i}(\omega)$
  if $\mathrm{k} = \perp$ then
    return $\perp_{\mathsf{KEM}}$
  $\mathrm{h_k} \leftarrow \mathsf{H}(\mathrm{k})$
  $\mathrm{m} \leftarrow \mathsf{DEM.D}_{\mathrm{h_k}}(\mathrm{C})$
  if $\mathrm{m} = \perp$ then
    return $\perp_{\mathsf{DEM}}$
  return $\mathrm{m}$
```

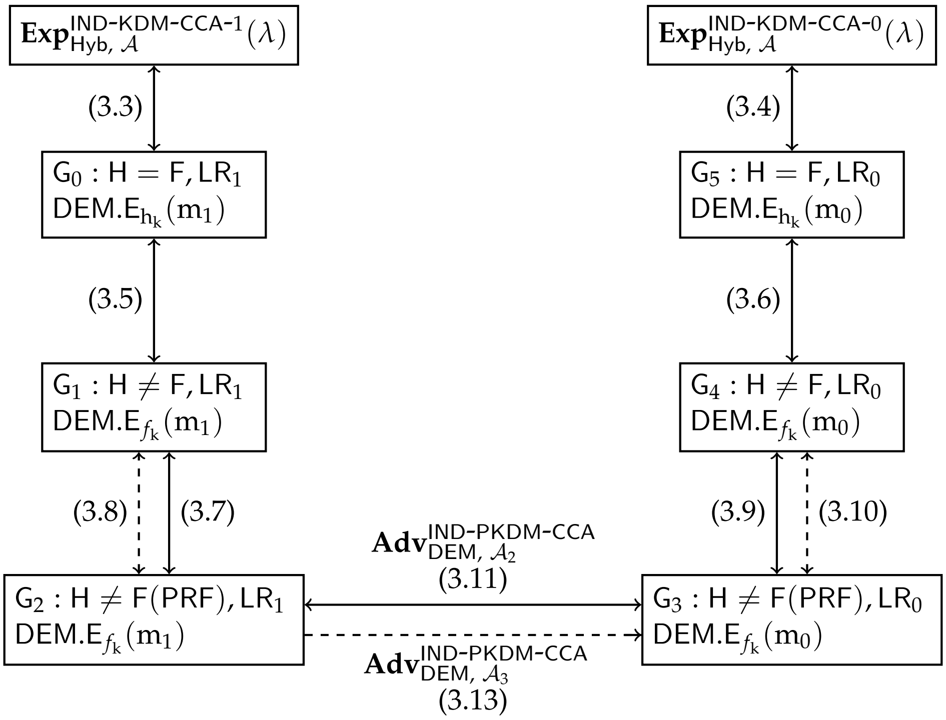
Figure 3.8: *The* IND-KDM-CCA *indistinguishability experiment made explicit for multi-key hybrid encryption in the random oracle model.*

the ephemeral key for the DEM, where knowledge of the private KEM key leads to immediate recovery of the ephemeral key. Since we regard Hyb as a public key encryption scheme in the context of key-dependent messages, it follows from Fig. 2.5 that it is on the private key of the KEM that key-dependent messages (that are input to the DEM) will depend.

We show that any KEM/DEM system that has a TYPE-1 μOW-CCA KEM and an IND-CCA DEM gives an IND-KDM-CCA[Ψ]-secure hybrid encryption scheme provided that the key derivation function KDF is modelled as a random oracle. In particular we allow the functions in Ψ to call the random oracle. By this we mean that when modelled as circuits, $\psi \in \Psi$ can have gates that explicitly call the random oracle. The μ indicates that there is a choice of multiple targets (KEM ciphertexts) to invert. Recall that our modelling of functions in $\psi \in \Psi$ as circuits implicitly implies that ψ is length-regular, meaning that given pk and ψ, one can uniquely determine the length of $\psi(\mathrm{sk})$ (this is the same restriction as made by Black et al. [65] and Backes et al. [20]). This result is formalised in Theorem 3.2. In Theorem 3.4 we provide an analogous, but significantly less tight result for TYPE-2 KEMs; in the security proof when an adversary makes a query $\mathsf{H}(\mathrm{k})$ to its random oracle, the checking oracle allows the reduction to determine whether this k corresponds to some challenge encapsulation ω.

The event $\mathsf{Coll}_{\mathsf{KEM}}(q_{\mathsf{LR}}, \lambda)$, parameterised by the number of oracle queries the adversary makes and the security parameter, implies a collision in the ephemeral key output by the KEM, which is extremely unlikely to occur (if it were, this would also adversely affect the KEM's one-wayness).

Proof intuition. In our proof we make use of the game-playing technique [183, 51] and introduce a sequence of games, as described in Fig. 3.9, and the games themselves are specified in Fig. 3.11. Apart from the simple, syntactical transitions (3.3) and (3.4), there are five game-hops to bound $\mathcal{A}$'s advantage distinguishing $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}1}}_{\mathsf{Hyb},\, \mathcal{A}}(\lambda)$ and $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}0}}_{\mathsf{Hyb},\, \mathcal{A}}(\lambda)$. These are denoted with solid lines. Here (3.5) and (3.6) are identical-until-bad hops. We define bad to be the event that the adversary queries the random oracle on a protokey k previously used by the left-or-right oracle.



$\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}1}}_{\mathsf{Hyb},\, \mathcal{A}}(\lambda)$
$\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}0}}_{\mathsf{Hyb},\, \mathcal{A}}(\lambda)$
(3.3)
(3.4)
$\mathsf{G}_0 : \mathsf{H} = \mathsf{F}, \mathsf{LR}_1$
$\mathsf{DEM.E}_{h_k}(m_1)$
$\mathsf{G}_5 : \mathsf{H} = \mathsf{F}, \mathsf{LR}_0$
$\mathsf{DEM.E}_{h_k}(m_0)$
(3.5)
(3.6)
$\mathsf{G}_1 : \mathsf{H} \neq \mathsf{F}, \mathsf{LR}_1$
$\mathsf{DEM.E}_{f_k}(m_1)$
$\mathsf{G}_4 : \mathsf{H} \neq \mathsf{F}, \mathsf{LR}_0$
$\mathsf{DEM.E}_{f_k}(m_0)$
(3.8)
(3.7)
(3.9)
(3.10)
$\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\, \mathcal{A}_2}$
(3.11)
$\mathsf{G}_2 : \mathsf{H} \neq \mathsf{F(PRF)}, \mathsf{LR}_1$
$\mathsf{DEM.E}_{f_k}(m_1)$
$\mathsf{G}_3 : \mathsf{H} \neq \mathsf{F(PRF)}, \mathsf{LR}_0$
$\mathsf{DEM.E}_{f_k}(m_0)$
$\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\, \mathcal{A}_3}$
(3.13)

Figure 3.9: *Diagrammatic overview of game hops used to prove Eqn. 3.2*

So far we have utilised standard tools: use the security of the KEM to decouple the key encapsulated by the KEM and the one used by the DEM (where Dent [99] used the same bad event in his analysis of IND-CCA secure KEMs), followed by a straightforward indistinguishability hop to the DEM. Unfortunately, with the introduction of key-dependent messages the latter hop has become considerably more challenging; moreover bounding the bad event in the presence of key-dependent messages is troublesome. To overcome these challenges, our proof uses a number of techniques.

To invoke the DEM's indistinguishability, the standard reduction would pick all the KEM key pairs and use these to simulate the KEM part of the hybrid encryption scheme (to run the adversary against the entire PKE). Since the reduction itself is playing the DEM indistinguishability game, it can use its DEM oracles to complete the DEM part (as the protokey encapsulated by the KEM and the ephemeral key used by the DEM are decoupled at this point). However if an adversary (against the PKE) may make queries with KDM functions that call the random oracle, it could in principle submit functions that decrypt past key encapsulations and, with the help of the random oracle, turn them into past DEM keys (effectively, the KDM function can cause the event that would normally have triggered bad). Since the reduction does not know the actual DEM keys being used, it suddenly finds itself in a tight spot and a direct hybrid argument (to get rid of past DEM keys) does not seem to work.

Our solution is to leverage the newly introduced $\mathsf{IND\text{-}PKDM\text{-}CCA}$ notion (Section 3.5.1.1). Since we model the KDM functions as circuits, it turns out to be possible to describe a compiler that turns a KDM function against the PKE into one against the DEM. There is however one further complication. For the public key scheme, we model the hash function as a random oracle and *the KDM function has access to the random oracle*. Yet, for the DEM scheme there is no random oracle present, which would suggest that the KDM function in the DEM world should not depend on one either. Moreover, it is not possible to predict on which values the KDM function would call the random oracle. Thus, when the random oracle is implemented by the reduction using lazy sampling, though it could hard-code the hash list so far into the circuit, the simulation might fail once fresh values are requested. To handle this, we (partly) model the random oracle as a pseudorandom function (rather than using lazy sampling). This provides the reduction a succinct description of the *entire* random oracle and it can safely embed the key to the pseudorandom function in the circuit used in the $\mathsf{IND\text{-}PKDM\text{-}CCA}$ game. The introduction of a PRF requires two additional hops, in equations (3.7) and (3.9).

The bounding of event bad is relatively easy on the m_0-side of the diagram, as one does not need to know the KEM's private key sk in order to simulate the data encapsulations: bad is bounded in G_3 by $\mathbf{Adv}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{A}_1}$. However, on the m_1-side of the diagram it is less obvious how to bound the bad event, since it is not possible to simulate the key-dependent values. The solution is to move the bad event from the m_1-side to the m_0-side using the separate hop (3.13), which bounds the difference between $\mathbf{Pr}[\mathsf{bad}]$ in games G_2 and G_3. This incurs a second $\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}_2}$ term to the bound.

Game	Oracle	Oracle Model	Message
$\mathbf{Exp}^1$	H	LS	m_1
G_0	$\boxed{\mathsf{H} = \mathsf{F}}$	LS	m_1
G_1	$\boxed{\mathsf{H} \neq \mathsf{F}}$	LS	m_1
G_2	$\mathsf{H} \neq \mathsf{F}$	$\boxed{\text{PRF}}$	m_1
G_3	$\mathsf{H} \neq \mathsf{F}$	PRF	$\boxed{\mathrm{m}_0}$
G_4	$\mathsf{H} \neq \mathsf{F}$	$\boxed{\text{LS}}$	m_0
G_5	$\boxed{\mathsf{H} = \mathsf{F}}$	LS	m_0
$\mathbf{Exp}^0$	$\boxed{\mathsf{H}}$	LS	m_0

Figure 3.10: *Description of game hops used to prove Eqn. 3.2.*

Bounding of the bad event breaks down if distinct queries to the LR oracle made identical KDF queries, meaning that there are collisions in the KEM. We bound this event by the separate quantity $\mathsf{Coll}_{\mathsf{KEM}}(q_{\mathsf{LR}}, \lambda)$. It might be possible to avoid this technicality by changing the scheme so it hashes $\mathsf{H}(\omega, \mathrm{k})$ instead of just $\mathsf{H}(\mathrm{k})$.

It may be of assistance to envision this game-hopping progression as a tree rather than the U-shaped left-or-right depiction in Fig. 3.9. From the root node representing $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}}b}_{\mathsf{Hyb},\,\mathcal{A}}$ there is one child representing G_0 and G_5, then this node has one child representing G_1 and G_4.From here this node fans out to two child nodes: one representing the hop from modelling the RO as lazy sampling to PRF, and another hop to depict the tracking of the event bad. The hop described in equation 3.13 coalesces these two child nodes into one, completing the indistinguishability argument.

Interpretation. When it comes to hybrid schemes, our result is very general. Indeed, it even generalises the work by Dent [99] (restricted to IND-CPA-security) as we can deal with key encapsulation schemes where the protokey is derived from the randomness in a hard-to-invert fashion. For instance, if $\mathbb{G}_\mathrm{p}$ is a cyclic group of order p with generator g, an obvious Diffie-Hellman-inspired KEM would pick private key $\mathrm{x} \in \mathbb{Z}_\mathrm{p}$, set public key g^x and compute a key encapsulation by generating a random $\mathrm{r} \in \mathbb{Z}_\mathrm{p}$, releasing g^r as the encapsulation of $\mathrm{k} = \mathrm{g}^{\mathrm{rx}}$. Our theorems can deal with this situation (where the KEM is TYPE-1 iff DDH is easy in $\mathbb{G}_\mathrm{p}$), but it is not covered by the KEMs given by Dent.

Black et al. [65] suggest the use of a variant of TDP-KEM combined with a one-time pad as a KDM-secure public key scheme in the random oracle model. Here TDP-KEM is shorthand for trapdoor-permutation-KEM, where the public and private key of the KEM match that of the trapdoor permutation and key encapsulation takes a random k in the domain of the trapdoor permutation, applies the permutation to encapsulate and outputs $\mathsf{H}(\mathsf{k})$ as ephemeral key, or, in the hybrid model with explicit key derivation function (Fig. 3.8) the KEM would output k as ephemeral protokey.

As a result of our theorem, if we restrict this scheme to any fixed-size message length, security is guaranteed. Strictly speaking, for arbitrary length messages, we would need to allow signalling of (an upper bound on) the message length to the random oracle so it can output the required number of bits. This is primarily a syntactical issue that we did not feel sufficiently important to incorporate into our main framework. Since TDP-KEM has an obvious checking oracle, we regard our Theorem 3.2 settling the problem left open by Black et al.

Theorem 3.2. *Let* Hyb *be a hybrid PKE scheme (Fig. 2.5) with a* TYPE-1 *KEM, with the key derivation function modelled by a random oracle. Let* Ψ *be any set of functions, including those which have random oracle access. Let* F *be an arbitrary family of pseudorandom functions. Then for any adversary* $\mathcal{A}$ *calling* LR *at most* q_{LR} *times, there exists algorithms* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *(of comparable computational complexity) such that*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Psi]}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda) \leq\ & 2\cdot\mathbf{Adv}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{A}_1}(\lambda) + 2\cdot\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}_2}(\lambda)\\ & + 2\cdot\mathsf{Coll}_{\mathsf{KEM}}(q_{\mathsf{LR}},\lambda) + 4\cdot\mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda)\ .\end{aligned}$$

This theorem, combined with Theorem 3.1, yields the following corollary relating to standard definitions.

Corollary 3.3. *As above, and let* n *be the number of DEM keys in the system, then:*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Psi]}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda) \leq\ & 2\cdot\mathbf{Adv}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{A}_1}(\lambda) + 2\mathsf{n}\cdot\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}_2}(\lambda)\\ & + 2\cdot\mathsf{Coll}_{\mathsf{KEM}}(q_{\mathsf{LR}},\lambda) + 4\cdot\mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda)\ .\end{aligned}$$

Proof: [of Theorem 3.2]

Fig. 3.8 contains a description of the security games $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}}b}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda)$ that are obtained by specifying the general PKE IND-KDM-CCA games for hybrid encryption where the key derivation function is modelled by a random oracle H. Certain lines are only applicable in some of the games, and this is indicated in the figure. For simplicity, we omit explicit mention of the class Ψ in the description of the security experiments. As is customary, we use lazy sampling to define H's behaviour, maintaining a list $\mathsf{H}_{\mathsf{list}}$ of query pairs $(\mathsf{k}, \mathsf{h}_{\mathsf{k}})$ produced by H so far.

```
$\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Psi]\text{-}b}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda)$:
    $\mathrm{pp} \leftarrow \mathsf{PGen}(\lambda)$
    $t \leftarrow 0$
    $\mathbf{sk} \leftarrow ()$
    $\boxed{\mathrm{x} \xleftarrow{\$} \{0,1\}^\lambda}$
    $\mathsf{H}_{\mathrm{list}}, \mathsf{F}_{\mathrm{list}}, \mathrm{FL} \leftarrow \emptyset$
    $b' \leftarrow \mathcal{A}^{\mathsf{New},\mathsf{H},\mathsf{LR}_b,\mathsf{Dec}}(\mathrm{pp})$
return $b'$

$\mathsf{New}()$:
    $t \leftarrow t+1$
    $(\mathrm{pk}_t, \mathrm{sk}_t) \leftarrow \mathsf{KGen}(\mathrm{pp})$
    append $\mathrm{sk}_t$ to $\mathbf{sk}$
    return $\mathrm{pk}_t$

$\mathsf{LR}_b(\psi^{\mathsf{H}}, i)$:
    if $\psi^{\mathsf{H}} \notin \Psi(\mathrm{pp}, \mathbf{pk}, i)$ then
        return $\lightning$
    $\mathrm{m}_1 \leftarrow \psi(\mathbf{sk})$
    $\mathrm{m}_0 \leftarrow 0^{|\mathrm{m}_1|}$
    $(\omega, \mathrm{k}) \leftarrow \mathsf{KEM.Encap}_{\mathrm{pk}_i}()$
    $\mathrm{h_k} \leftarrow \mathsf{F}(\mathrm{k})$
    $\mathrm{C}_b \leftarrow \mathsf{DEM.E}_{\mathrm{h_k}}(\mathrm{m}_b)$
    $\mathrm{FL} \leftarrow \mathrm{FL} \cup \{(\omega, \mathrm{C}_b, i)\}$
    return $(\omega, \mathrm{C}_b)$

$\mathsf{H}(\mathrm{k})$:
    if $(\mathrm{k}, \mathrm{h_k}) \in \mathsf{H}_{\mathrm{list}}$
        return $\mathrm{h_k}$
    if $(\mathrm{k}, \mathrm{h_k}) \in \mathsf{F}_{\mathrm{list}}$
        set bad $\leftarrow$ true
        $\boxed{\boxed{\textbf{return } \mathrm{h_k}}}$
    $\dbox{\mathrm{h_k} \xleftarrow{\$} \{0,1\}^\lambda}$
    $\dbox{\mathsf{H}_{\mathrm{list}} \leftarrow \mathsf{H}_{\mathrm{list}} \cup \{(\mathrm{k}, \mathrm{h_k})\}}$
    $\boxed{\mathrm{h_k} \leftarrow \mathsf{PRF}_{\mathrm{x}}(\mathrm{k})}$
    $\boxed{\mathsf{H}_{\mathrm{list}} \leftarrow \mathsf{H}_{\mathrm{list}} \cup \{(\mathrm{k}, \mathrm{h_k})\}}$
    return $\mathrm{h_k}$

$\mathsf{F}(\mathrm{k})$:
    if $(\mathrm{k}, \mathrm{h_k}) \in \mathsf{F}_{\mathrm{list}}$
        return $\mathrm{h_k}$
    if $(\mathrm{k}, \mathrm{h_k}) \in \mathsf{H}_{\mathrm{list}}$
        set bad $\leftarrow$ true
        $\boxed{\boxed{\textbf{return } \mathrm{h_k}}}$
    $\mathrm{h_k} \xleftarrow{\$} \{0,1\}^\lambda$
    $\mathsf{F}_{\mathrm{list}} \leftarrow \mathsf{F}_{\mathrm{list}} \cup \{(\mathrm{k}, \mathrm{h_k})\}$
    return $\mathrm{h_k}$

$\mathsf{Dec}(\omega, \mathrm{C}, i)$:
    if $(\omega, \mathrm{C}, i) \in \mathrm{FL}$ then
        return $\lightning$
    call $\mathrm{k} \leftarrow \mathsf{Decap}(\omega, i)$
    if $\mathrm{k} = \bot$ then
        return $\bot_{\mathsf{KEM}}$
    $\mathrm{h_k} \leftarrow \mathsf{HF}(\mathrm{k})$
    $\mathrm{m} \leftarrow \mathsf{DEM.D}_{\mathrm{h_k}}(\mathrm{C})$
    if $\mathrm{m} = \bot$ then
        return $\bot_{\mathsf{DEM}}$
    return $\mathrm{m}$

$\mathsf{HF}(\mathrm{k})$:
    if $(\mathrm{k}, \mathrm{h_k}) \in \mathsf{F}_{\mathrm{list}}$
        return $\mathrm{h_k}$
    if $(\mathrm{k}, \mathrm{h_k}) \in \mathsf{H}_{\mathrm{list}}$
        return $\mathrm{h_k}$
    $\dbox{\mathrm{h_k} \xleftarrow{\$} \{0,1\}^\lambda}$
    $\boxed{\mathrm{h_k} \leftarrow \mathsf{PRF}_{\mathrm{x}}(\mathrm{k})}$
    $\mathsf{H}_{\mathrm{list}} \leftarrow \mathsf{H}_{\mathrm{list}} \cup \{(\mathrm{k}, \mathrm{h_k})\}$
    return $\mathrm{h_k}$
```

Figure 3.11: *Security games used for proof Theorem 3.2. Games G_0 and G_5 imply that $\mathsf{H} = \mathsf{F}$ (as far as I/O behaviour is concerned). Games G_1–G_4 have $\mathsf{H} \neq \mathsf{F}$ as two independently sampled random oracles. Games G_2 and G_3 model the random oracle as a PRF, rather than using lazy sampling. Games G_0, G_1 and G_2 correspond to $b = 1$, whereas Games G_3, G_4 and G_5 correspond to $b = 0$. $\boxed{\textit{Items within thin boxes}}$ refer to code that is only used in G_2 and G_3, $\boxed{\boxed{\textit{items within thick boxes}}}$ are only for G_0 and G_5, and $\dbox{\textit{dashed boxed items}}$ refer to games G_0, G_1, G_4 and G_5.*

G_0 and G_5: Re-writing the security game. In the game there are four distinct places where queries to H could be made. Firstly, the adversary $\mathcal{A}$ can make direct H queries; any query to the oracle LR_b will require one 'direct' call to H for the key derivation and may include a number of indirect calls as part of the specified function ψ; and finally as a decryption query for key derivation. For the purpose of our game-hopping approach, we need to be able to make a clear distinction between these cases. To this end, we introduce two additional oracles: F and HF. We make a syntactical change so that LR_b always uses F for its key derivation, and Dec always uses HF. Oracle HF synchronises with items that are added to lists for both H and F. By ensuring that F, H and HF implement the same random oracle (i.e. are functionally equivalent, exhibiting exactly the same input/output behaviour), the changed games are equivalent to the original security experiments.

In Fig. 3.11, G_0 corresponds to such a modified, yet equivalent game, in this case for $b = 1$. The $b = 0$ sibling game is called G_5. In both of these games the oracles H and F each maintain their own list, H_{list}, respectively F_{list}, yet control code ensures *(a)* that these two lists can not contain k overlap in the sense that no triple $(\mathsf{k}, \mathsf{h}_\mathsf{k}, \mathsf{h}'_\mathsf{k})$ can exist for which both $(\mathsf{k}, \mathsf{h}_\mathsf{k}) \in \mathsf{H}_{\text{list}}$ and $(\mathsf{k}, \mathsf{h}'_\mathsf{k}) \in \mathsf{F}_{\text{list}}$ and *(b)* that the oracles H and F will look up elements from the other oracle's list, thus ensuring synchronisation. As a result of this design, F and H are functionally equivalent to each other in the games G_0 and G_5, implying that from an adversary's point of view G_0 is equivalent to $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}1}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda)$, or

$$\mathbf{Pr}\left[\mathsf{G}_0{}^{\mathcal{A}} = 1\right] = \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}1}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda) = 1\right] . \tag{3.3}$$

Similarly we claim that G_5 is equivalent to $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}0}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda)$, so

$$\mathbf{Pr}\left[\mathsf{G}_5{}^{\mathcal{A}} = 1\right] = \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}0}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda) = 1\right] . \tag{3.4}$$

G_1 **and** G_4**: Decoupling the Oracles.** We proceed by a more interesting hop, where we make F and H independent. The oracles F and H are modified such that when a query is made to one oracle (say H) that has previously been queried to the other (F) then a fresh value is still created (and added to H_{list}). Moreover, in this case the flag bad is set to `true` first. This is described in Fig. 3.11, where the new G_1 corresponds to the $b = 1$ case and G_4 to the $b = 0$ case. By syntactical inspection, G_0 and G_1 are identical up to the point at which the flag is set, enabling application of the fundamental lemma of game-hopping (see Section 2.1.1):

$$\left|\mathbf{Pr}\left[\mathsf{G}_0{}^{\mathcal{A}} = 1\right] \; - \; \mathbf{Pr}\left[\mathsf{G}_1{}^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_1\right] \tag{3.5}$$

and in a similar vein G_4 and G_5 are identical until bad, so

$$\left|\mathbf{Pr}\left[\mathsf{G}_4{}^{\mathcal{A}} = 1\right] \; - \; \mathbf{Pr}\left[\mathsf{G}_5{}^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_4\right] \; . \tag{3.6}$$

(To bound the difference between games G_4 and G_5 a standard hop involving the KEM's IND-CCA advantage is an alternative.)

The hop between the key-dependent scenario and the non-key-dependent world will be problematic later on due to the fact that if ψ calls the random oracle, the simulation cannot correctly answer these queries. This is because it does not know the values of the DEM keys in the system, only their indices. To counter this we add two additional hops in which we use a PRF rather than lazy sampling to model our random oracle. As stated earlier, we regard the ψ that acts on **sk** (of the KEM) as a circuit, with some gates that call the RO. There is a (one-to-one) mapping from ψ circuits (which act on **sk**) to ϑ circuits (that act on the DEM keys). We assume that there is some kind of 'safe storage' of all DEM keys. In this manner it is possible to track the past RO queries that are made by these ϑ functions. These H gates will have some inputs, and will check if the input string corresponds to some H_{list} entry, or an F_{list} entry. If it is an F query (i.e. made by an LR call), then assign a k_i to some of the output wires (since the game does not know the k_i but it can use them). However the issue is that if $\mathcal{A}$ gives a circuit ψ that makes an H query in a gate, and subsequently makes another H query, then the H_{list} lists will not be synchronised.

G_2 **and** G_3 **Using a PRF to model the Random Oracle.** To counter this, consider H as a pseudorandom function PRF : $\{0,1\}^\lambda \times \mathcal{K}_{\mathsf{DEM}} \rightarrow \{0,1\}^\lambda$ chosen from some PRF-secure function family F, parameterised by some seed $\mathsf{x} \in \{0,1\}^\lambda$, rather than using lazy sampling. Denote $\mathsf{PRF}_{\mathsf{x}}(\mathsf{k})$ as being the PRF applied to input k with seed x. The gates for H now store the F_{list}, and when calls to F are made we can wire up the corresponding k_i values. When the function makes H calls, we simply implement the PRF on the given input. To make this subtle change, we need to implement another two (symmetrical) game hops in which we change the way we model the random oracle from lazy sampling (LS) to using a PRF. The difference between $\mathcal{A}$'s advantage against G_1 and its advantage against G_2 is bounded by $\mathcal{A}$'s advantage in breaking the PRF:[4]

$$\mathbf{Pr}\left[\mathsf{G}_1{}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G}_2{}^{\mathcal{A}} = 1\right] \;\; \leq \;\; \mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) \tag{3.7}$$

$$\mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_1\right] - \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_2\right] \;\; \leq \;\; \mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) \tag{3.8}$$

[4] The more usual hop in a proof would be to replace a pseudorandom function by a perfectly random function, whereas here the perfect object is substituted by a computational approximation—for bounding the difference between the two worlds the 'direction' is irrelevant.

and likewise the difference between $\mathcal{A}$'s advantage against G_3 and its advantage against G_4 is bounded by the PRF advantage:

$$\begin{aligned} \mathbf{Pr}\left[\mathsf{G}_3{}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G}_4{}^{\mathcal{A}} = 1\right] &\leq \mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) & (3.9) \\ \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_4\right] - \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_3\right] &\leq \mathbf{Adv}^{\mathsf{PRF}}_{F,\,\mathcal{A}}(\lambda) & (3.10) \end{aligned}$$

Now we are in a position to consider the hop between games G_2 and G_3, i.e. from the key-dependent scenario to the key-independent side. In game G_2 the response from the left-or-right oracle is given to the adversary by LR_1, resulting to an encryption of $\mathrm{m}_1 = \psi(\mathbf{sk})$ in $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}1}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda)$, whereas in game G_3, the left-or-right oracle is implemented by LR_0, leading to an encryption of $\mathrm{m}_0 = 0^{|\psi(\mathbf{sk})|}$ (as in $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}0}}_{\mathsf{Hyb},\,\mathcal{A}}(\lambda)$). To show that games G_2 and G_3 are distinguishable only with small probability we introduce an adversary $\mathcal{A}_2$ that attacks the IND-PKDM-CCA property of the DEM, and show that as long as the DEM is secure in this respect, then the output of the games is indistinguishable. More precisely,

$$\mathbf{Pr}\left[\mathsf{G}_2{}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G}_3{}^{\mathcal{A}} = 1\right] \leq \mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}_2}(\lambda) \qquad (3.11)$$

The consequence of F and H being independently sampled oracles is that in games G_2 and G_3 the encapsulated key and the key used for the DEM are effectively decoupled (as the adversary has no direct access to F). This decoupling allows us to use a DEM hop to prove equation (3.11), and this reduction is detailed in Fig. 3.12. In the game that $\mathcal{A}_2$ plays, it runs $\mathcal{A}$ as a black-box that returns a valid ψ, then $\mathcal{A}_2$ creates messages m_0 and m_1 in the same way that the LR_b oracle does in the other games. However, where in the games G_2 and G_3 there was an explicit oracle F that provided linkage between a protokey k output by the KEM and its corresponding ephemeral key $\mathrm{h_k}$ actually used by the DEM, in the simulation $\mathcal{A}_2$ uses its own oracles to create the keys $\mathrm{h_k}$ in the IND-PKDM-CCA experiment it itself is playing. To get this to go through we need to move the function ψ that acts on the KEM secret keys to the function ϑ, that acts upon DEM keys. The set k_ϑ contains all the DEM keys that are currently in the system. To simulate the DEM hop we need to make sure that the ϑ circuit in the IND-PKDM-CCA game is consistent with the circuit that acts on all of the DEM keys in the system in the PKE game. Every time $\mathcal{A}$ makes an F query in its PKE game we need to add that key to the set of keys that ϑ can act upon.

In this decoupled scenario, reduction $\mathcal{A}_2$ generates the $(\mathrm{pk}, \mathrm{sk})$ pairs itself. The seed of the PRF is then 'hardwired' into the gates of ϑ so when $\mathcal{A}$'s KDM function makes a RO call, it is dealt with by this setup. This allows the simulation to go through without $\mathcal{A}_2$ actually knowing which

```
$\mathcal{A}_2$ playing $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}}b}_{\mathsf{DEM},\,\mathcal{A}_2}(\lambda)$:
    $\mathrm{pp} \leftarrow \mathsf{KEM.PGen}(\lambda)$
    $t \leftarrow 0$
    $\mathbf{sk} \leftarrow ()$
    $\mathrm{x} \xleftarrow{\$} \{0,1\}^{\lambda}$
    $\mathsf{H}_{\text{list}}, \mathsf{F}_{\text{list}}, \mathrm{FL} \leftarrow \emptyset$
    $b' \leftarrow \mathcal{A}^{\mathsf{New},\mathsf{H},\mathsf{LR}_b,\mathsf{Dec}}(\mathrm{pp})$
    return $b'$

$\mathsf{New}()$:
    $t \leftarrow t+1$
    $(\mathrm{pk}_t, \mathrm{sk}_t) \leftarrow \mathsf{KGen}(\mathrm{pp})$
    append $\mathrm{sk}_t$ to $\mathbf{sk}$
    return $\mathrm{pk}_t$

$\mathsf{H}(\mathrm{k})$:
    if $(\mathrm{k}, \mathrm{h}_{\mathrm{k}}) \in \mathsf{H}_{\text{list}}$
        return $\mathrm{h}_{\mathrm{k}}$
    $\mathrm{h}_{\mathrm{k}} \leftarrow \mathsf{PRF}_{\mathrm{x}}(\mathrm{k})$
    $\mathsf{H}_{\text{list}} \leftarrow \mathsf{H}_{\text{list}} \cup \{(\mathrm{k}, \mathrm{h}_{\mathrm{k}})\}$
    return $\mathrm{h}_{\mathrm{k}}$
```

```
$\mathsf{LR}_b(\psi^{\mathsf{H}}, i)$:
    $\psi^{\mathsf{H}} \to \vartheta$:
        for RO query k:
            if $(\mathrm{k}, j) \in \mathsf{F}_{\text{list}}$
                Incorporate $\mathrm{k}_j$ into $\vartheta$
            else
                $\mathrm{h} \leftarrow \mathsf{PRF}_{\mathrm{x}}(\mathrm{k})$
    $(\omega, \mathrm{k}) \leftarrow \mathsf{KEM.Encap}_{\mathrm{pk}_i}()$
    if $(\mathrm{k}, \mathrm{h}_{\mathrm{k}}) \in \mathsf{H}_{\text{list}}$ then
        ABORT
    if $(\mathrm{k}, j) \in \mathsf{F}_{\text{list}}$ then
        call $\mathsf{C} \leftarrow \mathsf{LR}(j, \vartheta)$
    else
        call $j \leftarrow \mathsf{New}()$
        $\mathsf{F}_{\text{list}} \leftarrow \mathsf{F}_{\text{list}} \cup \{(\mathrm{k}, j)\}$
        call $\mathsf{C} \leftarrow \mathsf{LR}(j, \vartheta)$
    $\mathrm{FL} \leftarrow \mathrm{FL} \cup \{(\omega, \mathsf{C}, i)\}$
    return $(\omega, \mathsf{C})$

$\mathsf{Dec}(\omega, \mathsf{C}, i)$:
    if $(\omega, \mathsf{C}, i) \in \mathrm{FL}$ then
        return $\lightning$
    $\mathrm{k} \leftarrow \mathsf{Decap}_{\mathrm{sk}_i}(\omega)$
    if $(\mathrm{k}, j) \in \mathsf{F}_{\text{list}}$
        call $\mathrm{m} \leftarrow \mathsf{D}(\mathsf{C}, j)$
    else
        $\mathrm{h}_{\mathrm{k}} \leftarrow \mathsf{H}(\mathrm{k})$
        $\mathrm{m} \leftarrow \mathsf{DEM.D}_{\mathrm{h}_{\mathrm{k}}}(\mathsf{C})$
    return m
```

Figure 3.12: *Description of reduction $\mathcal{A}_2$ used to prove* (3.11)*. When $\mathcal{A}_2$ runs $\mathcal{A}$, it needs to create an environment* $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA}}_{\mathsf{Hyb},\,\mathcal{A}}$*. It makes* New *queries and specifies the public key index i in its* LR_b *queries. The messages* m_0 *and* m_1 *and also ω and* k *are 'created' just as they are in normal* LR_b*, whereas* h_{k} *is virtually set to whatever value is used in the game $\mathcal{A}_2$ itself is playing by $\mathcal{A}_2$'s calls to* New, LR *and* D *(from* $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\,\mathcal{A}_2}$*). Note that $\mathcal{A}_2$ need not know* h_{k} *for this simulation.*

values k_i are queried to the RO. The messages m_1 and m_0 are then 'created' just as they are in $\mathcal{A}$'s LR queries. Now $\mathcal{A}_2$ calls its own oracles LR, New and Dec (in the IND-PKDM-CCA game) and returns a pair (ω, C_b) as $\mathcal{A}$ would have expected.

The LR oracle in the simulation translates the ψ into a ϑ. If this function makes an oracle call k_i, the simulation checks H_{list} for an entry containing k_i, and if present returns the corresponding h_{k}. If the value is on F_{list} then the simulation will know the index of the key but not the value itself, and thus a PRF gate can be called to retrieve the corresponding h_{k}. If it is on neither list, simply initiate PRF on k_i.

Since the adversary $\mathcal{A}$ has no direct access to F this indirect simulation of F is perfect. As a result, if $\mathcal{A}_2$ is in $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}1}}_{\mathsf{DEM},\,\mathcal{A}_2}(\lambda)$ then $\mathcal{A}$ will behave towards $\mathcal{A}_2$ exactly as it would do in G_2, and similarly if $\mathcal{A}_2$ is in $\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}0}}_{\mathsf{DEM},\,\mathcal{A}_2}(\lambda)$ then $\mathcal{A}$ will behave as in G_3, proving (3.11).

All that remains is bounding the probability of the bad event in games G_2 and G_3, followed by a collection of the various terms into a single bound on the advantage.

G_3**: Analysing the event** bad**.** The analysis of the bad event in game G_3 is easiest, as here the adversary is given an encryption of a zero string which is clearly not key-dependent (since the adversary directly specifies its length). By simple code inspection, it emerges that $\mathcal{A}$ can set the flag bad to `true` in two places in G_3: either in a direct oracle query to H on a k that has already been queried to F by LR_0; or if LR_0 calls F on a k that has previously been queried to H directly by $\mathcal{A}$. Intuitively, the former constitutes a break against the one-wayness of the KEM, and the latter should just be very unlikely (although we actually bound it by a break as well to avoid the need for an additional assumption on the way k as output by KEM is distributed).

Fig. 3.13 details reduction $\mathcal{A}_1$, for which

$$\mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_3\right] \leq \mathbf{Adv}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{A}_1}(\lambda) + \mathsf{Coll}_{\mathsf{KEM}}(q_{\mathsf{LR}}, \lambda). \tag{3.12}$$

First we observe that if Enc (internally) creates a pair (ω, k) and (ω', k') satisfying $\mathrm{k} = \mathrm{k}'$ yet $\omega \neq \omega'$ the simulation will with high probability produce $\mathsf{F}(\mathrm{k}) \neq \mathsf{F}(\mathrm{k}')$, indicating that in that case it is not perfect. However, the event that such a pair is created by a KEM ought to be small. We define $\mathsf{Coll}_{\mathsf{KEM}}(q, \lambda)$ as the probability this happens in q queries to the encapsulation oracle.

In order to simulate correctly, we require that the reductions can make as many New calls as $\mathcal{A}$ can. To do this we can simply set an upper bound on the number of New calls that $\mathcal{A}$ makes, and then restrict the number of calls $\mathcal{A}_1$ can make by this figure. The value z labels the DEM keys that are brought into the system by $\mathcal{A}$'s calls to its Encap oracle.

If a collision as above does not happen then $\mathcal{A}_1$ creates a perfect simulation of G_3 as long as bad is not set. Moreover, at the very point a query is made that would have caused bad to be set in G_3, the reduction $\mathcal{A}_1$ uses its KEM-checking oracle KEM.Chk to detect that bad was set and retrieves the corresponding key k, plus the index of the Enc query this key belongs to.

As a technical aside, to simulate G_3 the reduction needs to answer the adversary $\mathcal{A}$'s LR_0 queries. Since $\mathcal{A}$ gives out ψ and expects an encryption of $0^{|\psi(\mathbf{sk})|}$, it is necessary (in order to simulate correctly) for $\mathcal{A}_1$ to learn $|\psi(\mathbf{sk})|$ without knowing $\mathbf{sk}$. Here the length regularity condition is required: given $\mathbf{pk}$ and ψ, we can determine $|\psi(\mathbf{sk})|$ and thus simulate LR_0.

```
$\mathcal{A}_1$ playing $\mathbf{Exp}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\,\mathcal{A}_1}(\lambda)$:
    receive pp
    $z \leftarrow 0$
    $\mathsf{H}_{\text{list}}, \mathsf{F}_{\text{list}}, \mathrm{FL} \leftarrow \emptyset$
    $b' \leftarrow \mathcal{A}^{\mathsf{New},\mathsf{H},\mathsf{LR},\mathsf{Dec}}(\mathrm{pp})$
    return $\perp$

$\mathsf{New}()$:
    call $\mathrm{pk}_t \leftarrow \mathsf{New}()$
    $\mathbf{pk} \leftarrow \mathbf{pk} \cup \mathrm{pk}_t$
    return $\mathrm{pk}_t$

$\mathsf{H}(\mathrm{k})$:
    if $(\mathrm{k}, \mathrm{h_k}) \in \mathsf{H}_{\text{list}}$
        return $\mathrm{h_k}$
    for $(\omega, \mathrm{h}, j) \in \mathsf{F}_{\text{list}}$
        if $\mathsf{KEM.Chk}(\omega, \mathrm{k}) = \mathtt{true}$ then
            exit $(j, \mathrm{k})$
    $\mathrm{h_k} \xleftarrow{\$} \{0,1\}^\lambda$
    $\mathsf{H}_{\text{list}} \leftarrow \mathsf{H}_{\text{list}} \cup \{(\mathrm{k}, \mathrm{h_k})\}$
    return $\mathrm{h_k}$
```

```
$\mathsf{LR}(\psi, i)$:
    $\mathrm{m}_0 \leftarrow 0^{|\psi(\mathbf{sk})|}$
    $z \leftarrow z + 1$
    call $\omega \leftarrow \mathsf{Encap}(i)$
    $\mathrm{h_k} \leftarrow \mathsf{F_{SIM}}(\omega, i)$
    $\mathrm{C} \leftarrow \mathsf{DEM.E}_{\mathrm{h_k}}(\mathrm{m}_0)$
    $\mathrm{FL} \leftarrow \mathrm{FL} \cup \{(\omega, \mathrm{C}, i)\}$
    return $(\omega, \mathrm{C})$

$\mathsf{F_{SIM}}(\omega, i)$:
    for $(\mathrm{k}, \mathrm{h_k}) \in \mathsf{H}_{\text{list}}$ do
        if $\mathsf{KEM.Chk}(\omega, \mathrm{k}) = \mathtt{true}$ then
            exit $(z, \mathrm{k})$
    if $(\omega, \mathrm{h}, j) \in \mathsf{F}_{\text{list}}$
        $\mathrm{h_k} \leftarrow \mathrm{h}$
    else
        $\mathrm{h_k} \xleftarrow{\$} \{0,1\}^\lambda$
    $\mathsf{F}_{\text{list}} \leftarrow \mathsf{F}_{\text{list}} \cup \{(\omega, \mathrm{h_k}, z)\}$
    return $\mathrm{h_k}$

$\mathsf{Dec}(\omega, \mathrm{C}, i)$:
    if $(\omega, \mathrm{C}, i) \in \mathrm{FL}$ then
        return $\lightning$
    call $\mathrm{k} \leftarrow \mathsf{Decap}(\omega, i)$
    if $\mathrm{k} = \perp$ then
        return $\perp_{\mathsf{KEM}}$
    $\mathrm{h_k} \leftarrow \mathsf{H}(\mathrm{k})$
    $\mathrm{m} \leftarrow \mathsf{DEM.D}_{\mathrm{h_k}}(\mathrm{C})$
    if $\mathrm{m} = \perp$ then
        return $\perp_{\mathsf{DEM}}$
    return m
```

Figure 3.13: *Description of reduction $\mathcal{A}_1$ used to prove* (3.12). *When $\mathcal{A}_1$ runs $\mathcal{A}$, it needs to create an environment $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA}}_{\mathsf{Hyb},\,\mathcal{A}}$. The line "**exit** (j, k)" indicates that $\mathcal{A}_1$ at that point terminates running $\mathcal{A}$ and returns (j, k) to its own environment (as guess for k_j). As long as* Enc *(internally) does not create a pair (ω, k) and (ω', k') with* $\mathrm{k} = \mathrm{k}'$ *yet $\omega \neq \omega'$ the simulation is perfect.*

G_2 **and** G_3**: Deferring analysis of the event** bad. The analysis of the bad event in G_2 is considerably more challenging and a direct approach (as done for G_3) does not work. Instead, we take inspiration from the "deferred analysis" technique of Gennaro and Shoup [116]. Rather than analysing the bad events in G_2, we defer the analysis to G_3 (for which we already have a bound). However, it is not at all evident that in the hop G_2 to G_3 the probability the bad flag is set stays the same (as was the case for the deferred analysis by Gennaro and Shoup). Indeed, it is unlikely to be the case, however we are able to show that the difference between the two bad events from occurring is bound by IND-PKDM-CCA advantage of an adversary $\mathcal{A}_3$ (as described in Fig. 3.14) against the DEM, so

$$\mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_2\right] - \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_3\right] \leq \mathbf{Adv}^{\text{IND-PKDM-CCA}}_{\text{DEM}, \mathcal{A}_3}(\lambda)\ . \tag{3.13}$$

```
$\mathcal{A}_3$ playing $\mathbf{Exp}^{\text{IND-PKDM-CCA-}b}_{\text{DEM}, \mathcal{A}_3}(\lambda)$:
    pp $\leftarrow$ $\mathsf{PGen}(\lambda)$
    $t \leftarrow 0$
    **sk** $\leftarrow ()$
    $\mathrm{x} \xleftarrow{\$} \{0,1\}^{\lambda}$
    $\mathsf{H}_{\text{list}}, \mathsf{F}_{\text{list}}, \mathrm{FL} \leftarrow \emptyset$
    $b' \leftarrow \mathcal{A}^{\mathsf{New},\mathsf{H},\mathsf{LR}_b,\mathsf{Dec}}(\mathrm{pp})$
    **if** an ABORToccurs **then**
        **return** 1
    **return** 0

$\mathsf{New}()$:
    $t \leftarrow t + 1$
    $(\mathrm{pk}_t, \mathrm{sk}_t) \leftarrow \mathsf{KGen}(\mathrm{pp})$
    Append $\mathrm{sk}_t$ to **sk**
    **return** $\mathrm{pk}_t$

$\mathsf{H}(\mathrm{k})$:
    **if** $(\mathrm{k}, \mathrm{h_k}) \in \mathsf{H}_{\text{list}}$ **then**
        **return** $\mathrm{h_k}$
    **if** $(\mathrm{k}, *) \in \mathsf{F}_{\text{list}}$ **then**
        ABORT
    $\mathrm{h_k} \leftarrow \mathsf{PRF}_{\mathrm{x}}(\mathrm{k})$
    $\mathsf{H}_{\text{list}} \leftarrow \mathsf{H}_{\text{list}} \cup \{(\mathrm{k}, \mathrm{h_k})\}$
    **return** $\mathrm{h_k}$
```

```
$\mathsf{LR}_b(\psi^{\mathsf{H}}, i)$:
    $\psi^{\mathsf{H}} \rightarrow \vartheta$
        **for RO query** k:
            **if** $(\mathrm{k}, j) \in \mathsf{F}_{\text{list}}$
                Incorporate $\mathrm{k}_j$ into $\vartheta$
            **else**
                $\mathrm{h} \leftarrow \mathsf{PRF}_{\mathrm{x}}(\mathrm{k})$
    $(\omega, \mathrm{k}) \leftarrow \mathsf{KEM.Encap}_{\mathrm{pk}_i}()$
    **if** $(\mathrm{k}, *) \in \mathsf{H}_{\text{list}}$ **then**
        ABORT
    **if** $(\mathrm{k}, j) \in \mathsf{F}_{\text{list}}$ **then**
        **call** $\mathrm{C}_b \leftarrow \mathsf{LR}(j, \vartheta)$
    **else**
        **call** $j \leftarrow \mathsf{New}()$
        $\mathsf{F}_{\text{list}} \leftarrow \mathsf{F}_{\text{list}} \cup \{(\mathrm{k}, j)\}$
        **call** $\mathrm{C}_b \leftarrow \mathsf{LR}(j, \vartheta)$
    $\mathrm{FL} \leftarrow \mathrm{FL} \cup \{(j, \mathrm{C})\}$
    **return** $(\omega, \mathrm{C}_b)$

$\mathsf{Dec}(\omega, \mathrm{C}, i)$:
    **if** $(\omega, \mathrm{C}, i) \in \mathrm{FL}$ **then**
        **return** $\lightning$
    $\mathrm{k} \leftarrow \mathsf{Decap}_{\mathrm{sk}_i}(\omega)$
    **if** $(\mathrm{k}, j) \in \mathsf{F}_{\text{list}}$
        **call** $\mathrm{m} \leftarrow \mathsf{D}(\mathrm{C}, j)$
    **else**
        $\mathrm{h_k} \leftarrow \mathsf{H}(\mathrm{k})$
        $\mathrm{m} \leftarrow \mathsf{DEM.D}_{\mathrm{h_k}}(\mathrm{C})$
    **return** m
```

Figure 3.14: *Description of reduction $\mathcal{A}_3$ used to prove* (3.13). *When $\mathcal{A}_3$ runs $\mathcal{A}$, it needs to create an environment $\mathbf{Exp}^{\text{IND-KDM-CCA}}_{\text{Hyb}, \mathcal{A}}$. The messages m_0 and m_1 and also ω and k are 'created' just as they are in normal LR_b, whereas $\mathrm{h_k}$ is virtually set to whatever value is used in the game $\mathcal{A}_3$ itself is playing by $\mathcal{A}_3$'s calls to* New, LR *and* D *(from $\mathbf{Exp}^{\text{IND-PKDM-CCA}}_{\text{DEM}, \mathcal{A}_3}$). The number of key pairs $\mathcal{A}_3$ can ask for is upper-bounded by the number of* New *queries $\mathcal{A}$ makes. Note that $\mathcal{A}_3$ need not know $\mathrm{h_k}$ for this simulation.*

Similarly to the analysis of (3.11), it is necessary to translate the function ψ into a ϑ, and align the simulated queries correctly. We set this up so that the bad event in the security games corresponds to $\mathcal{A}_3$ causing an ABORT in the reduction.

1. If $\mathcal{A}_3$ is in game IND-PKDM-CCA-1 then, unless ABORT occurs, this is a perfect simulation of G_2 for $\mathcal{A}$.

2. If $\mathcal{A}_3$ is in game $\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}0}$ then, unless ABORT occurs, this is a perfect simulation of G_3 for $\mathcal{A}$.

3. $\mathcal{A}_3$ will ABORT iff the event bad occurs in (either) G_2 (or G_3).

Consequently we have

$$\begin{aligned}
\mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_2\right] &= \mathbf{Pr}\left[\mathcal{A}_3 \text{ ksees ABORT in } \mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}1}}\right] \\
\mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_3\right] &= \mathbf{Pr}\left[\mathcal{A}_3 \text{ sees ABORT in } \mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}0}}\right] .
\end{aligned}$$

Since by construction (and definition) we also have

$$\mathbf{Pr}\left[\mathcal{A}_3 \text{ sees ABORT in } \mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}}b}\right] = \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}PKDM\text{-}CCA\text{-}}b} = 1\right]$$

and so our claim (3.13) follows. Finally we put all of the terms together and arrive at the claimed bound.

$$\mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Psi]}_{\mathsf{Hyb},\, \mathcal{A}}(\lambda) = \left|(-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CCA\text{-}}b}_{\mathsf{Hyb},\, \mathcal{A}}(\lambda) = 1\right]\right|$$

$$\begin{aligned}
= \quad & \left|\mathbf{Pr}\left[\mathsf{G_0}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_5}^{\mathcal{A}} = 1\right]\right| && \text{[by (3.3),(3.4)]} \\
= \quad & \left|\mathbf{Pr}\left[\mathsf{G_0}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_1}^{\mathcal{A}} = 1\right] + \mathbf{Pr}\left[\mathsf{G_1}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_2}^{\mathcal{A}} = 1\right]\right. \\
& +\mathbf{Pr}\left[\mathsf{G_2}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_3}^{\mathcal{A}} = 1\right] + \mathbf{Pr}\left[\mathsf{G_3}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_4}^{\mathcal{A}} = 1\right] \\
& \left.+\mathbf{Pr}\left[\mathsf{G_4}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_5}^{\mathcal{A}} = 1\right]\right| \\
\leq \quad & \left|\mathbf{Pr}\left[\mathsf{G_0}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_1}^{\mathcal{A}} = 1\right] + \mathbf{Adv}^{\mathsf{PRF}}_{F,\, \mathcal{A}}(\lambda) + \mathbf{Pr}\left[\mathsf{G_2}^{\mathcal{A}} = 1\right]\right. \\
& \left.-\mathbf{Pr}\left[\mathsf{G_3}^{\mathcal{A}} = 1\right] + \mathbf{Adv}^{\mathsf{PRF}}_{F,\, \mathcal{A}}(\lambda) + \mathbf{Pr}\left[\mathsf{G_4}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_5}^{\mathcal{A}} = 1\right]\right| && \text{[by (3.7), (3.9)]} \\
\leq \quad & \left|\mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_1\right]\right| + \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_4\right] + && \text{[by (3.5), (3.6)]} \\
& +\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\, \mathcal{A}_2}(\lambda) + 2 \cdot \mathbf{Adv}^{\mathsf{PRF}}_{F,\, \mathcal{A}}(\lambda) && \text{[by (3.11)]} \\
\leq \quad & \left|\mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_2\right] + \mathbf{Adv}^{\mathsf{PRF}}_{F,\, \mathcal{A}}(\lambda)\right| + \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_3\right] && \text{[by (3.8), (3.10)]} \\
& +\mathbf{Adv}^{\mathsf{PRF}}_{F,\, \mathcal{A}}(\lambda) + \mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\, \mathcal{A}_2}(\lambda) + 2 \cdot \mathbf{Adv}^{\mathsf{PRF}}_{F,\, \mathcal{A}}(\lambda) \\
\leq \quad & 2 \cdot \mathbf{Pr}\left[\mathcal{A} \text{ sets bad in } \mathsf{G}_3\right] + \mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\, \mathcal{A}_2}(\lambda) \\
& +\mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\, \mathcal{A}_3}(\lambda) + 4 \cdot \mathbf{Adv}^{\mathsf{PRF}}_{F,\, \mathcal{A}}(\lambda) && \text{[by (3.13)]} \\
\leq \quad & 2 \cdot \mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\, \mathcal{A}_2}(\lambda) + 2 \cdot \mathbf{Adv}^{\mu\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\, \mathcal{A}_1}(\lambda) \\
& +2 \cdot \mathsf{Coll}_{\mathsf{KEM}}(q_{\mathsf{LR}}, \lambda) + 4 \cdot \mathbf{Adv}^{\mathsf{PRF}}_{F,\, \mathcal{A}}(\lambda) . && \text{[by (3.12)]}
\end{aligned}$$

3.5.2.1 IND-KDM Security of TYPE-2 Hybrid Encryption

Recall that in TYPE-2 KEMs there is no efficient oracle that takes a protokey k and a key encapsulation ω and informs the querying entity whether or not ω is an encapsulation of k or not.

Theorem 3.4. *Let* Hyb *be a hybrid PKE scheme as defined above that comprises a* TYPE-2 *key encapsulation mechanism* KEM *and a data encapsulation mechanism* DEM*. For any adversary $\mathcal{A}$ that asks at most q oracle queries (encryption queries + direct RO queries + indirect RO queries), and for all length-regular $\psi \in \Psi$, there exists algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ such that*

$$\mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CCA}[\Psi]}_{\mathsf{Hyb},\ \mathcal{A}}(\lambda) \leq 2q \cdot \mathbf{Adv}^{\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\ \mathcal{A}_1}(\lambda) \ + \ 2 \cdot \mathbf{Adv}^{\mathsf{IND\text{-}PKDM\text{-}CCA}}_{\mathsf{DEM},\ \mathcal{A}_2}(\lambda) + 4 \cdot \mathbf{Adv}^{\mathsf{PRF}}_{F,\ \mathcal{A}}(\lambda)$$

Proof: The proof is identical to the proof of Theorem 3.2 up until equation (3.12), and instead we have:

$$\mathsf{Pr}[\mathcal{A} \text{ sets bad in } \mathsf{G}_3] \leq q \cdot \mathbf{Adv}^{\mathsf{OW\text{-}CCA}}_{\mathsf{KEM},\ \mathcal{A}_1}(\lambda) \tag{3.14}$$

In TYPE-2 KEMs an adversary playing the OW-CCA game cannot check if each query is equal to the challenge. As a result, in the simulation of LR_0 the probability that we can win the KEM game when $\mathcal{A}$ sets bad in G_2 is $\frac{1}{q}$, where q is the total number of queries that $\mathcal{A}$ makes. This factor carries through in the term collection at the end of the proof.

3.6 Separation Results for KDM Security in the Hybrid Framework

We now detail two schemes that are secure under standard security definitions, yet are insecure when the adversary has access to a key cycle of length 2. We recall that security against 2-cycles is referred to as 2-circular security. We cast the two schemes, both reliant on the SXDH assumption [19] and containing somewhat 'artificial properties,' in the hybrid framework. Since there is no need to make anything other than cosmetic changes to the schemes to make them hybrid, the security and insecurity results relating the full schemes follow from the original papers (and we refer to the original papers for full details and proofs).

By inspection, it also easy to see that both KEMs are IND-CPA secure under the SXDH assumption and that the DEMs are one-time IND-CPA secure, assuming a key derivation function is pseudorandom. The reason the schemes nonetheless fail to fall under our framework, is that the key derivation function is only applied to *part* of the key. Indeed, it is the other part of the key (and how it is used by the DEM) that has been carefully crafted by the original authors to ensure 2-circular insecurity. As a result, these existing counterexamples also serve as proof that the KEM-DEM framework *per se* does not provide any 'leverage' when it comes to increasing resistance against key-dependent message attacks.

3.6.1 Writing Acar et al.'s es_2 as a KEM and DEM

Acar et al. [6] present schemes that are $\mathsf{IND\text{-}R}$-secure (indistinguishable from random) if the SXDH problem is hard, yet not 2-circular secure (see Section 3.3.3). Let GS be a group scheme for which SXDH is hard. The original paper shows $\mathsf{es}_2 = (\mathsf{es}_2.\mathsf{PGen}, \mathsf{es}_2.\mathsf{KGen}, \mathsf{es}_2.\mathsf{Enc}, \mathsf{es}_2.\mathsf{Dec}, \mathsf{es}_2.\mathsf{MsgR}, \mathsf{es}_2.\mathsf{CtxtR})$, an asymmetric scheme. The paper also presents a symmetric scheme but our focus will be on es_2. The decryption keys are in the message space.

The authors showed, in lemmas 3 and 4 of [6], that this scheme is $\mathsf{IND\text{-}R}$-secure but not 2-circular secure. We will now cast this scheme in a hybrid framework and comment on the security properties. Fig. 3.15 details hybrid scheme $\mathsf{Hyb}_1 = (\mathsf{Hyb}_1.\mathsf{PGen}, \mathsf{Hyb}_1.\mathsf{KGen}, \mathsf{Hyb}_1.\mathsf{KEM.Encap}, \mathsf{Hyb}_1.\mathsf{DEM.E}, \mathsf{Hyb}_1.\mathsf{KEM.Decap}, \mathsf{Hyb}_1.\mathsf{DEM.D})$ which is a straightforward adaptation of the es_2 scheme. The KEM part generates keys that are uniform over the keyspace, however it is important to note that the keyspace of the DEM part depends on pp, so we must allow some joint parameter generation between the KEM and the DEM. The changes from the original scheme are only cosmetic, and as a result the security properties of the original scheme—namely $\mathsf{IND\text{-}R}$-secure yet KDM- and 2-circular-insecure—still hold.

The KEM as described in Fig. 3.15 is $\mathsf{IND\text{-}CPA}$ secure under the SXDH assumption: parameters, public key, key encapsulation and encapsulated key form two independent DDH instances in each of the groups, so replacing any of these (in particular the keys) with unrelated random elements from the same group will go unnoticed (assuming SXDH). The DEM is deterministic, and hence it can clearly not be multi-time secure. However, it is one-time secure: For every T_1 and m_2, there is a unique Ω_1 such that $T_1 = \Omega_1^{m_2}$, indicating that m_2 is information-theoretically hidden given T_1 and similarly for m_1 and T_2; moreover, if H is a balanced function, then $m_1 + H(\ldots, Z_1)$ operates as a perfect one-time pad. Thus the DEM is perfectly one-time secure if H is a balanced function. If H is a pseudorandom generator, security degrades to one-time $\mathsf{IND\text{-}CPA}$ security.

As an aside, while Acar et al. noted the scheme was insecure against 2-cycles, they did not remark on 1-cycles. It is easy to see that security also breaks down in this case: If $\psi(\mathrm{sk}) = \psi(x_1, x_2) = (x_1, x_2)$ then an adversary can disinguish the encryption of $(m_1, m_2) = (x_1, x_2)$: $T_1 = g_1^{x_1 u_1 / x_2}$ and $T_2 = g_2^{x_2 u_2 / x_1}$ so $\mathbf{e}(T_1, T_2) = \mathbf{e}(g_1, g_2)^{x_1 u_1 x_2 u_2 / x_1 x_2} = \mathbf{e}(g_1, g_2)^{u_1 u_2} = \mathbf{e}(U_1, U_2)$ so the adversary can simply check if these values coincide.

```
$\mathsf{Hyb}_1.\mathsf{PGen}(\lambda)$ :
    $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2) \xleftarrow{\$} \mathsf{GS}(\lambda)$
    $pp \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$
    return $pp$

$\mathsf{Hyb}_1.\mathsf{KGen}(pp)$ :
    $x_1, x_2 \xleftarrow{\$} \mathbb{Z}_p^*$
    $X_1 \leftarrow g_1^{x_1}$ ; $X_2 \leftarrow g_2^{x_2}$
    $sk \leftarrow (x_1, x_2)$ ; $pk \leftarrow (X_1, X_2)$
    return $(pk, sk)$

$\mathsf{Hyb}_1.\mathsf{KEM.Encap}(pp, pk)$ :
    $(X_1, X_2) \leftarrow pk$
    $y_1, y_2, u_1, u_2 \xleftarrow{\$} \mathbb{Z}_p^*$
    $Y_1 \leftarrow g_1^{y_1}$ ; $U_1 \leftarrow g_1^{u_1}$ ; $Z_1 \leftarrow X_1^{y_1}$
    $Y_2 \leftarrow g_2^{y_2}$ ; $U_2 \leftarrow g_2^{u_2}$ ; $Z_2 \leftarrow X_2^{y_2}$
    $\Omega_1 \leftarrow X_1^{u_1}$ ; $\Omega_2 \leftarrow X_2^{u_2}$
    $\omega \leftarrow (Y_1, Y_2, U_1, U_2)$
    $k \leftarrow (Z_1, Z_2, \Omega_1, \Omega_2)$
    return $(\omega, k)$
```

```
$\mathsf{Hyb}_1.\mathsf{DEM.E}(pp, pk, k, (m_1, m_2))$
    $(Z_1, Z_2, \Omega_1, \Omega_2) \leftarrow k$
    $T_1 \leftarrow \Omega_1^{1/m_2}$ ; $T_2 \leftarrow \Omega_2^{1/m_1}$
    $c_1 \leftarrow m_1 + H(pp, 1, Z_1)$
    $c_2 \leftarrow m_2 + H(pp, 2, Z_2)$
    $C \leftarrow (T_1, T_2, c_1, c_2)$
    return $C$

$\mathsf{Hyb}_1.\mathsf{KEM.Decap}(pp, sk, \omega)$
    $(x_1, x_2) \leftarrow sk$
    $(Y_1, Y_2, U_1, U_2) \leftarrow \omega$
    $Z_1 \leftarrow Y_1^{x_1}$ ; $Z_2 \leftarrow Y_2^{x_2}$
    $\Omega_1 \leftarrow U_1^{x_1}$ ; $\Omega_2 \leftarrow U_2^{x_2}$
    $k \leftarrow (Z_1, Z_2, \Omega_1, \Omega_2)$
    return $k$

$\mathsf{Hyb}_1.\mathsf{DEM.D}(pp, k, C)$
    $(T_1, T_2, c_1, c_2) \leftarrow C$
    $(Z_1, Z_2, \Omega_1, \Omega_2) \leftarrow k$
    $m_1 \leftarrow c_1 - H(pp, 1, Z_1)$
    $m_2 \leftarrow c_2 - H(pp, 2, Z_2)$
    return $(m_1, m_2)$
```

Figure 3.15: *Hybrid scheme* Hyb_1 *based on* es_2 *of [6]. KEM is* IND-CPA *if DDH in* $\mathbb{G}_1$ *holds.*

3.6.2 Writing Cash et al.'s Π_{CPA} as a KEM and DEM

The encryption scheme Π_{CPA} of Cash et al. [87] also uses asymmetric bilinear groups $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ of prime order p, and assumes that $\mathbb{G}_1$ and $\mathbb{G}_2$ are distinct and that the DDH assumption holds in both (i.e. the SXDH assumption). The scheme includes functions $\mathsf{encode} : \mathcal{M} \mapsto \{0,1\}^{l(\lambda)}$ and $\mathsf{decode} : \{0,1\}^{l(\lambda)} \mapsto \mathcal{M}$, which denote an invertible encoding scheme, where $l(\lambda)$ is the polynomial length of the encoded message. Let $F : \mathbb{G}_T \mapsto \{0,1\}^{l(\lambda)}$ be a pseudorandom generator.

The authors show that this scheme is IND-CPA secure, however when given a circular encryption of two keys, an adversary can distinguish another ciphertext with probability 1/2. In fact, with probability 1/2 over the coins used in key generation, the adversary can recover both secret keys. In the appendix of the full version of their paper, the authors give another scheme that is IND-CPA secure without using the 'group-switching' technique, and experiences catastrophic collapse (meaning key recovery) in the presence of a 2-cycle, with even higher adversarial success probability.

Fig. 3.16 details scheme Hyb_2 which is a straightforward casting of the Π_{CPA} scheme in the hybrid framework. Again, the changes from the original scheme are only cosmetic: the (in)security properties of the original scheme, namely IND-CPA secure yet 2-circular insecure carry over without reserve. Here we consider the security of the KEM and the DEM as defined by us. The DEM is secure for arguments similar to those used for the Acar et al. scheme: it is one-time IND-CPA secure provided that F is a pseudorandom generator.

```
$\mathsf{Hyb}_2.\mathsf{PGen}(\lambda)$ :
    $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2) \xleftarrow{\$} \mathsf{GS}(\lambda)$
    $pp \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$
    return $pp$

$\mathsf{Hyb}_2.\mathsf{KGen}(pp)$ :
    $x_1, x_2 \xleftarrow{\$} \mathbb{Z}_p$
    $\beta \xleftarrow{\$} \{0,1\}$
    $Y_1 \leftarrow \mathbf{e}(g_1, g_2)^{x_1}$
    if $\beta = 0$ then
        $Y_2 \leftarrow g_1^{x_2}$
    if $\beta = 1$ then
        $Y_2 \leftarrow g_2^{x_2}$
    $sk \leftarrow (\beta, x_1, x_2)$
    $pk \leftarrow (\beta, Y_1, Y_2)$
    return $(pk, sk)$

$\mathsf{Hyb}_2.\mathsf{KEM.Encap}(pp, pk)$ :
    $(\beta, Y_1, Y_2) \leftarrow pk$
    $r \xleftarrow{\$} \mathbb{Z}_p$ ; $R \xleftarrow{\$} \mathbb{G}_T$
    $c_2 \leftarrow R.Y_1^r$ ; $k_1 \leftarrow R$
    $k_2 \leftarrow Y_2^r$
    if $\beta = 0$ then
        $c_1 \leftarrow g_1^r$
    if $\beta = 1$ then
        $c_1 \leftarrow g_2^r$
    $\omega \leftarrow (c_1, c_2)$
    $k \leftarrow (k_1, k_2)$
    return $(\omega, k)$
```

```
$\mathsf{Hyb}_2.\mathsf{DEM.E}(pp, pk, k, (m_1, m_2))$
    $(\beta, Y_1, Y_2) \leftarrow pk$
    $(\alpha, m_1, m_2) \leftarrow M$
    $(k_1, k_2) \leftarrow k$
    $I \leftarrow F(k_1) \oplus \mathsf{encode}(M)$
    $c_4 \leftarrow I$
    if $\beta = 0$ then
        $c_3 \leftarrow k_2^{m_2}.g_1^{m_2}$
    if $\beta = 1$ then
        $c_3 \leftarrow k_2^{m_2}$
    $C \leftarrow (c_3, c_4)$
    return $C$

$\mathsf{Hyb}_2.\mathsf{KEM.Decap}(pp, sk, \omega)$
    $(\beta, x_1, x_2) \leftarrow sk$
    $(c_1, c_2) \leftarrow \omega$
    if $\beta = 0$ then
        $R \leftarrow (c_2 / \mathbf{e}(c_1, g_2)^{x_1})$
        $k_2 \leftarrow c_1^{x_2}$
    if $\beta = 1$ then
        $R \leftarrow (c_2 / \mathbf{e}(g_1, c_1)^{x_1})$
        $k_2 \leftarrow c_1^{x_2}$
    $k_1 \leftarrow R$
    $k \leftarrow (k_1, k_2)$
    return $k$

$\mathsf{Hyb}_2.\mathsf{DEM.D}(pp, k, C)$
    $(c_3, c_4) \leftarrow C$
    $(k_1, k_2) \leftarrow k$
    $M' \leftarrow F(k_1) \oplus c_4$
    $M \leftarrow \mathsf{decode}(M')$
    return $M$
```

Figure 3.16: *Asymmetric scheme* Hyb_2 *based on* Π_{CPA} *of Cash et al. [87]. Message space is* $\mathcal{M} = \{0,1\} \times \mathbb{Z}_p \times \mathbb{Z}_p$ *thus* m_1 *and* m_2 *must be non-zero, these values can be included in the message space by proper encoding. The ciphertext space is* $\mathbb{G}_{\beta+1} \times \mathbb{G}_T \times \mathbb{G}_{\beta+1} \times \{0,1\}^{l(\lambda)}$*. Note that* Y_2 *may be either in* $\mathbb{G}_1$ *or* $\mathbb{G}_2$ *depending on the structure of the public key.*

By construction, if $\beta = 0$ the KEM part takes place primarily in $\mathbb{G}_1$, whereas if $\beta = 1$ there is a symmetric move to $\mathbb{G}_2$. Consequently, we only need to analyse the KEM security for $\beta = 0$ and the $\beta = 1$ follows by symmetry. For $\beta = 0$, we have that (g_1, c_1, Y_2, k_2) forms a DDH tuple;

if we substitute $\vec{Y_1} = g_1^{x_1}$ for Y_1 and inverse R through the pairing resulting in $\vec{R} = \mathbf{e}(R, g_2)$ and let $\tilde{c_2} = \vec{R}.\vec{Y_1}^{r}$ (all this only makes the adversary's life easier) then the same holds for the tuple $(g_1, \vec{Y_1}, \vec{R}, \tilde{c}_2/\vec{R})$. We conclude that the KEM (for $\beta = 0$) is one-way secure if the CDH assumption holds in $\mathbb{G}_1$ and and $\mathsf{IND\text{-}CPA}$ if the DDH assumption holds (in $\mathbb{G}_1$).

We observe that the scheme Π_{CPA} is not trivially insecure under 1-cycles. In fact, if the square decision Diffie Hellman (SDDH) [24] assumption holds in both groups the scheme seems 1-cycle secure and we conjecture that the scheme is actually fully single-key $\mathsf{IND\text{-}KDM}[\Psi]$ secure in the Generic Group Model [181] (adapted to the asymmetric pairing setting).

3.7 Conclusions

This chapter has detailed how to construct a hybrid encryption scheme that is secure in the presence of an adversary that has access to encryptions that depend on the secret key. The construction, proven in the random oracle model, demonstrates that a key encapsulation mechanism that is $\mu\mathsf{OW\text{-}CCA}$ secure combined with a data encapsulation mechanism that is $\mathsf{IND\text{-}CCA}$ secure yields an $\mathsf{IND\text{-}KDM\text{-}CCA}$ secure hybrid encryption scheme. Further, known separation results for circular security are extended to the hybrid framework to demonstrate that hybrid encryption does not provide any leverage when it comes to mitigating key-dependent message attacks.

The main open problem stemming from this work regards instantiating the random oracle. As mentioned above, Chang et al. [88] showed that the Kurosawa-Desmedt hybrid encryption scheme is KDM secure in the standard model with respect to a very limited function class. While this is an interesting result, this function class only appears to be useful in the anonymous credential framework (where hybrid encryption is not particularly beneficial). A generic standard model result for achieving KDM security for hybrid encryption remains a significant challenge, with the major sticking point being the key derivation function. An open problem is to investigate whether in general instantiating the KDF is possible using the Universal Computational Extractor [37, 38, 36] paradigm, or if one can show this to be impossible using indistinguishability obfuscation with results reminiscent of those of Brzuska, Farshim and Mittelbach [78, 79] and Brzuska and Mittelbach [80, 81].

CHAPTER 4

Related-Key Attack Security

Contents

4.1 Overview and Motivation **63**

4.2 Related Literature **64**

4.3 Existing Security Models and Formalism **67**

4.3.1 RKA Security for PRFs and PRPs 68

4.3.2 RKA Security for Symmetric Encryption 69

4.4 RKA-KDM Security **70**

4.4.1 Security Definition 71

4.4.2 A Generic Construction 72

4.4.3 A Retraction 75

4.4.4 Instantiations 79

4.5 Conclusions **97**

4.1 Overview and Motivation

The work in this chapter is largely based on ***Encryption Schemes Secure under Related-Key and Key-Dependent Message Attacks*** *by Böhl, Davies and Hofheinz [68] published at PKC 2014. The full version is available via ePrint [67].*

The study of related-key attack (RKA) security attempts to model the scenario where an adversary can manipulate the execution of a cryptographic protocol or primitive in such a way that a modified key replaces the 'honest' key. This is at odds with the standard assumption in security definitions that the adversary gets only black-box access to the primitive(s) in question. This is a real-world concern: consider an adversary that can tamper with a device so that certain bits of a secret key of a block cipher are flipped, or inject a fault into a certificate authority so that signing is performed under a modified key [71, 57].

The design of cryptographic primitives that are robust in the presence of related-key attacks has become a desirable security goal, with block ciphers receiving particular attention. This effort has been complemented by a significant effort by the theoretical research community to understand the limits of the RKA framework, in terms of secure constructions and impossibility results.

The modification applied to the secret key is assumed to be provided by the adversary, and as a result security definitions are parameterised by a set of functions that act on the keyspace. Naturally it is desirable to achieve RKA security for particular primitives with respect to a rich function class, however certain natural function classes yield trivial attacks. In the related literature section we detail existing results regarding the robustness of primitives such as PRFs, block ciphers and SKE/PKE in the presence of RKAs.

A natural question arises: what is the link between related-key attack security and key-dependent message security? Both models consider an adversary that has capabilities beyond the scope of standard security definitions, so it is reasonable to assume that an adversary may be able to force encryptions under related keys of messages that could depend on the keys. A joint notion of security combining related-key and KDM attacks was given by Applebaum at TCC 2013 [15]; the work in this chapter will describe a generic framework for constructing schemes that are secure in this joint model. Intuitively, our framework states that a symmetric encryption scheme is RKA-KDM secure if it achieves KDM security and there exists an efficient *RKA transformer* that produces encryptions under related keys without knowledge of the underlying key. To demonstrate the effectiveness of the framework, we present instantiations that meet the requirements to yield schemes secure under a number of computational assumptions (namely DDH, DCR, QR and LWE).

The published version [68] and early editions of the full version [67] contain a stronger claim which turns out to be false. The stronger claim was that a symmetric encryption scheme is RKA-KDM secure if the following simultaneously hold: standard IND-CPA security, the existence of a transformer that produces encryptions under related keys, and the existence of a transformer that produces key-dependent messages. A corollary of this claim was that the combination of IND-CPA security and the KDM transformer implied KDM-CPA security, and Section 4.4.2 will detail why this is not the case.

4.2 Related Literature

Block Cipher Cryptanalysis. Initial research efforts in related-key cryptography were geared towards block cipher cryptanalysis; Knudsen [143] and Biham [53] independently developed chosen-plaintext attacks against LOKI91 [77]. Related-key cryptanalysis of numerous ciphers including Triple-DES was given by Kelsey, Schneier and Wagner [138, 139]. Iwata and Kohno [135] looked at two crucial components of the 3GPP security architecture [1, 2] used in mobile communications: the symmetric encryption scheme $f8$ (variant of OFB mode of operation) and the message authentication code $f9$ (variant of CBC-MAC) and showed how to prove both secure if the underlying block cipher is an RKA-secure PRP (for XOR-induced related-key attacks).

A number of papers presenting related-key cryptanalysis of AES [190, 142, 121, 61, 60, 59] and other practical block ciphers [54, 55, 56] encouraged the theoretical side of RKA security research we refer to these works and the references therein for a more complete overview of this vast research topic.

Theoretical Models. In 1997 Boneh et al. [71] (PKE setting) and Biham and Shamir [57] (symmetric encryption setting) showed how tampering could allow an adversary to mount what we now know as related-key attacks. A more formal security treatment was provided by Bellare and Kohno [44], who gave the first definition of related-key deriving functions and further study of related-key-secure cryptography with a focus on block ciphers. Soon after, Lucks [158] gave a construction that is secure when restricting the adversary to only modifying part of the key; the construction hashes the secret key before use and then models the hash function as a random oracle.

Gennaro et al. [115] gave a theoretical framework attempting to capture side-channel analysis, and reached many of the same impossibility results as Bellare and Kohno[1]. Ishai, Prabhakaran, Sahai and Wagner [133], building upon [134, 115], investigate so-called *private circuits* which are circuits that carry out a specific functionality while protecting internal secret information even when an adversary can invoke an unbounded number of faults. Albrecht et al. [9] gave an idealised notion of computation where the key may depend on the ideal primitive itself.

RKA-Secure Primitives. In 2010 Bellare and Cash [28] gave PRPs and PRFs that are RKA secure in the standard model and based on DDH and DLIN (see Section 4.3.1). Not long afterwards Bellare, Cash and Miller [31] gave definitions of RKA security for a number of primitives (see Sections 4.3.1 and 4.3.2) and looked at how one could give RKA-secure constructions of symmetric encryption, identity-based encryption (IBE) and signatures from RKA-secure PRFs (for linear functions since no constructions secure against richer classes were available at the time). Bellare, Meiklejohn and Thomson [45] show how to build RKA-secure signatures from RKA-secure one-way functions, and show that many intuitive one-way functions are already RKA secure (all in

[1] As pointed out by Abdalla et al. [5], Gennaro et al. [115] did not seem to be aware of the work of Bellare and Kohno [44].

the context of linear functions, also see Section 3.2 for details of this paper's work on KDM-secure storage). Instantiations of IBE and other primitives, RKA-secure for affine and polynomial functions, were given by Bellare, Paterson and Thomson [47], and Wee [186] presented RKA-secure PKE schemes (for linear functions). Abdalla, Benhamouda, Passelègue and Paterson [5] gave RKA-secure PRFs for affine and polynomial functions (a weaker result for these function classes was given in the concurrent work of Lewi et al. [154] based on the PRFs of Boneh et al. [73], and Applebaum and Widder [18] gave linear PRFs for the scenario where the keyspace is bit-strings rather than group elements). Abdalla, Benhamouda and Passelègue [4] generalised the framework of [5] and gave PRFs that are RKA secure with respect to arbitrary permutations of polynomials.

Other Approaches. To emphasise the diversity of the RKA literature, we remark that use of one-time RKA-secure schemes as a building block for fuzzy extractors robust against adversarial modification was a short-lived hot topic [104, 92, 137], and later Goldenberg and Liskov [118] considered related-key secure PRFs and block ciphers in terms of pseudorandom bits, showing that a 1-bit RKA-secure PRG is sufficient to build RKA-secure PRPs. Applebaum, Harnik and Ishai [17] presented LPN and LWE-based chosen-plaintext attack-secure symmetric encryption schemes resistant to linear RKAs, and Pietrzak [170] showed how these schemes were actually secure against affine RKAs. Bitansky and Canetti [63] looked at point obfuscation and gave RKA-secure constructions, Dziembowski, Pietrzak and Wichs [109] focused on an information-theoretic approach for tamper prevention, and Goyal, O'Neill and Rao [122] gave a general primitive called a *correlated input secure hash function*, which has applications beyond RKA security.

Topics Linked to RKA Security. Related-key attack security can be seen as a component in the theoretical study of attacks on real systems, with two other main areas in this subfield being KDM security (see Chapter 3) and leakage resilient cryptography (see [134, 161, 108, 64, 106] and references therein). Other adversarial models include resistance against reset attacks [189] (adversary can rewind an honest party to an earlier state), randomness-dependent messages [58, 126] and the recent work on related-randomness attacks [168, 36] (adversary can force a scheme to reuse (functions of) past random values).

RKA-KDM Security. In work that inspired the results in this chapter, Applebaum [15] gave the first joint notion of security for encryption that is secure against related-key attacks and key-dependent message attacks, in the context of garbled circuits. The RKA-KDM-secure construction given by Applebaum uses an RKA-KDM secure symmetric encryption scheme to garble XOR gates 'for free' (i.e. no explicit encryption is required). Applebaum highlighted the strength of this joint notion by giving a scheme that is RKA secure and KDM secure, yet suffers a key-recovery attack in the presence of an adversary with RKA-KDM query capability. Intuitively the scheme (see [14] Section 5.3, full version of [15]) consists of a double encryption where the inner encryption is RKA secure (for linear RKA functions) and an outer encryption that is KDM secure (for linear KDM functions).

4.3 Existing Security Models and Formalism

We now formalise the ability of an adversary to tamper with the keys used in cryptographic systems. We refer to $\Phi : \mathcal{K} \to \mathcal{K}$ as the set of *related-key deriving* (RKD) functions that the adversary is allowed to query. Creation of primitives secure with respect to rich and meaningful function classes has proved to be a considerable challenge for the cryptographic community, and while the literature is now diverse it took a long time for such candidate schemes to appear.

Using the notation of [47], we write Φ^{C} for the set of constant functions. If $\mathcal{K}$ is a group under operation $*$ then Φ^{lin} is the set of linear functions on $\mathcal{K}$, and if $\mathcal{K}$ is a ring then Φ^{aff} denotes affine functions and $\Phi^{\mathrm{poly}(d)}$ denotes polynomial functions over $\mathcal{K}$ of degree at most d:

$$\begin{aligned}
\Phi^{\mathrm{C}} &:= \{\varphi_C : \mathcal{K} \to \mathcal{K},\ \mathrm{k} \mapsto C : C \in \mathcal{K}\} \\
\Phi^{\mathrm{lin}} &:= \{\varphi_a : \mathcal{K} \to \mathcal{K},\ \mathrm{k} \mapsto a * \mathrm{k} : a \in \mathcal{K}\} \\
\Phi^{\mathrm{aff}} &:= \{\varphi_{a,b} : \mathcal{K} \to \mathcal{K},\ \mathrm{k} \mapsto a \cdot \mathrm{k} + b : a, b \in \mathcal{K}\} \\
\Phi^{\mathrm{poly}(d)} &:= \{\varphi_q : \mathcal{K} \to \mathcal{K},\ \mathrm{k} \mapsto q(\mathrm{k}) : q \in \mathcal{K}_d\}.
\end{aligned}$$

Naturally, the class of functions that we allow the adversary access to depends on the algebraic structure of the keyspace: for bitstrings it makes more sense to restrict attention to linear functions, where $*$ is XOR. A function class Φ is *claw-free* if for all distinct $\varphi \neq \varphi' \in \Phi$, then $\varphi(\mathrm{k}) \neq \varphi'(\mathrm{k})\ \forall \mathrm{k} \in \mathcal{K}$. Note that while constant and linear functions are inherently claw-free, affine and polynomial functions in general are not. In the literature many authors only consider $\mathcal{K} = \{0,1\}^\lambda$ for some security parameter λ, and only consider (linear) *XOR-induced related-key attacks*

$$\Phi^{\mathrm{lin}} \supset \Phi^{\mathrm{xrka}} := \{\varphi_\Delta : \{0,1\}^\lambda \to \{0,1\}^\lambda,\ \mathrm{k} \mapsto \Delta \oplus \mathrm{k} : \Delta \in \{0,1\}^\lambda\},$$

and this is a function class we will use later on. Note that in this class the adversary cannot fix key bits to certain values, and cannot swap the position of key bits.

4.3.1 RKA Security for PRFs and PRPs

We now detail the definition of RKA security for pseudorandom permutations given by Bellare and Kohno [44]. This definition is given for historical context and is not used later in the chapter, however the set up of the security model and constructions are indicative of the approach to RKA security throughout the literature. The notation $\mathsf{Perm}(\mathcal{K}, \mathcal{D})$ refers to the set of all block ciphers (permutations) with domain $\mathcal{D}$ and keyspace $\mathcal{K}$, so $\mathsf{G} \xleftarrow{\$} \mathsf{Perm}(\mathcal{K}, \mathcal{D})$ means selecting a random block cipher over $\mathcal{D}$.

```
$\mathbf{Exp}^{\mathsf{RKA\text{-}PRP}[\Phi]\text{-}b}_{\mathsf{E},\,\mathcal{A}}(\lambda)$ :
    $\mathsf{G} \xleftarrow{\$} \mathsf{Perm}(\mathcal{K}, \mathcal{D})$
    $\mathsf{k} \xleftarrow{\$} \mathcal{K}$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\lambda)$
    return $b'$
```

```
$\mathsf{LR}_b(\varphi, \mathsf{x})$ :
    if $\varphi \notin \Phi$ then
        return $\lightning$
    $\mathsf{c}_1 \leftarrow \mathsf{E}_{\varphi(\mathsf{k})}(\mathsf{x})$
    $\mathsf{c}_0 \leftarrow \mathsf{G}_{\varphi(\mathsf{k})}(\mathsf{x})$
    return $\mathsf{c}_b$
```

Figure 4.1: *The experiment defining* RKA-PRP[Φ] *security for block cipher* E.

Definition 4.1 (RKA-PRP[Φ] Security.). *Let $\mathsf{E} : \mathcal{K} \times \mathcal{D} \to \mathcal{D}$ be a family of functions where the domain is specified by a security parameter λ, and let Φ be a set of RKD functions over the keyspace $\mathcal{K}$.*

Then the RKA-PRP*[Φ] advantage for an adversary $\mathcal{A}$ against* E *is defined by*

$$\mathbf{Adv}^{\mathsf{RKA\text{-}PRP}[\Phi]}_{\mathsf{E},\,\mathcal{A}}(\lambda) \overset{def}{=} \left| \sum_{b \in \{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{RKA\text{-}PRP}[\Phi]\text{-}b}_{\mathsf{E},\,\mathcal{A}}(\lambda) = 1\right] \right|$$

where experiment $\mathbf{Exp}^{\mathsf{RKA\text{-}PRP}[\Phi]\text{-}b}_{\mathsf{E},\,\mathcal{A}}$ is given in Fig. 4.1.

An analogous experiment for RKA-PRF[Φ] security can be realised by setting the output $\mathcal{R}$ to be something not necessarily equal to the input domain $\mathcal{D}$, and choosing the function G from the set of all functions from $\mathcal{K} \times \mathcal{D} \to \mathcal{R}$.

As mentioned above, if the constant function $\varphi(\mathrm{k}) = C$ is allowed for some C known to the adversary, then RKA security (for any indistinguishability-style notion) for any deterministic primitive cannot be achieved since the adversary can make one query to the real-or-random oracle, calculate the primitive on the input under key C and check if it matches the oracle output.

Bellare and Kohno [44] showed that Naor and Reingold's [162] DDH-based PRF and Lewko and Waters' [155] DLIN-based PRF are insecure in the presence of related-key attacks. Lucks gave RKA[Φ^{lin}]-secure PRF constructions for $\mathcal{K} = \mathbb{Z}_\mathrm{M}$ (for certain composite M) where $*$ is addition modulo M, under novel interactive assumptions. Bellare and Cash [28] gave $\mathsf{RKA}[\Phi^{\mathrm{lin}}]$-secure PRFs (for claw-free classes Φ since linear functions are inherently claw-free) and used deterministic extractors to get $\mathsf{RKA}[\Phi^{\mathrm{lin}}]$-secure PRGs with bitstring outputs, then used those as a key-derivation function for a normal (i.e. not RKA-secure) PRP to get $\mathsf{RKA}[\Phi^{\mathrm{lin}}]$-secure PRPs (note the function class with respect to which these primitives provide security carries through from PRF to PRG to PRP). Their main construction is secure with respect to keyspace $(\mathbb{Z}_{\mathrm{p}}^{*})^{\mathrm{n}+1}$ under component-wise multiplication modulo p, with security based on the DDH assumption. They also gave an $\mathsf{RKA}[\Phi^{\mathrm{lin}}]$-secure PRF based on DLIN. The Bellare-Cash [28] framework relies on two properties: a *key-transformer* which is very similar to the RKA transformer described in Section 4.4.2, and so-called *key-fingerprinting* which means that the adversary's queries are appropriately separated prior to being processed by the PRF. The original framework has a bug, as described in the most recent iteration of the full version [29], meaning that the analysis of a third construction, an additive DDH-based $\mathsf{RKA}[\Phi^{\mathrm{lin}}]$-secure PRF (with exponential security reduction) no longer holds.

Abdalla et al. [5] give $\mathsf{RKA}[\Phi^{\mathrm{aff}}]$- and $\mathsf{RKA}[\Phi^{\mathrm{poly}(d)}]$-secure PRFs, recovering the withdrawn Bellare-Cash construction and removing the claw-free assumption for the wider classes. The construction for affine RKD functions is secure under DDH, and for polynomial functions the authors require the decisional d-Diffie-Hellman Inversion (d-DDHI) assumption that was introduced by Goyal et al. [122].

4.3.2 RKA Security for Symmetric Encryption

We now present the definition of RKA security in the context of symmetric encryption, and again follow the 'real-or-random' paradigm. The definition is parameterised by the RKD function class Φ. An adversary has access to an encryption oracle which on input a RKD function $\varphi \in \Phi$ and message $\mathrm{m} \in \mathcal{M}$ receives either an encryption of m under the related key $\mathsf{E}_{\varphi(\mathrm{k})}(\mathrm{m})$ or an encryption under the related key of a dummy message. In the previous section we referred to the $\mathsf{RKA\text{-}PRP}[\Phi]$ security for PRPs, however for encryption we aid exposition by referring to $\mathsf{RKA}[\Phi]$ security, dropping explicit mention of the primitive in question.

For this definition and for the rest of this chapter, we will assume that the message space $\mathcal{M}$ is finite and fixed a priori, reflecting the discussion in Section 3.3.1. While this restriction on the message space is at odds with the standard assumption of much of the symmetric encryption community that messages are arbitrary-length bitstrings, the analysis later on will utilise a number of techniques based in public key cryptography, including message spaces in algebraic groups. Note that the message space $\mathcal{M}$ may depend on the public parameters pp.

```
$\mathbf{Exp}^{\mathsf{RKA}[\Phi]\text{-}b}_{\Sigma,\,\mathcal{A}}(\lambda)$ :
  $\mathrm{pp} \leftarrow \mathsf{Pg}(\lambda)$
  $\mathrm{k} \leftarrow \mathsf{Kg}(\mathrm{pp})$
  $\mathrm{m}_0 \xleftarrow{\$} \mathcal{M}$
  $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathrm{pp})$
  return $b'$
```

```
$\mathsf{LR}_b(\varphi, \mathrm{m})$ :
  if $\varphi \notin \Phi$ then
    return $\lightning$
  $\mathrm{m}_1 \leftarrow \mathrm{m}$
  $\mathrm{c} \leftarrow \mathsf{E}_{\varphi(\mathrm{k})}(\mathrm{m}_b)$
  return c
```

Figure 4.2: *The experiment defining* RKA[Φ] *security for symmetric encryption.*

Definition 4.2 (RKA[Φ] Security.). *Let $\Sigma = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{E}, \mathsf{D})$ be a symmetric encryption scheme with message space $\mathcal{M}$. Adversary $\mathcal{A}$ can make encryption queries by submitting $(\varphi \in \Phi, \mathrm{m} \in \mathcal{M})$ and the responses it receives are detailed by the LR_b oracle in Fig. 4.2.*

Then the $\mathsf{RKA}[\Phi]$ advantage for an adversary $\mathcal{A}$ against Σ is defined by

$$\mathbf{Adv}^{\mathsf{RKA}[\Phi]}_{\Sigma,\,\mathcal{A}}(\lambda) \overset{def}{=} \left| \sum_{b \in \{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{RKA}[\Phi]\text{-}b}_{\Sigma,\,\mathcal{A}}(\lambda) = 1\right] \right|$$

where experiment $\mathbf{Exp}^{\mathsf{RKA}[\Phi]\text{-}b}_{\Sigma,\,\mathcal{A}}$ is given in Fig. 4.2.

As mentioned earlier, Bellare, Cash and Miller's [31] transform turned a $\mathsf{RKA}[\Phi^{\mathrm{lin}}]$-secure PRF (e.g. from Bellare and Cash [28]) into a (CPA) $\mathsf{RKA}[\Phi^{\mathrm{lin}}]$-secure symmetric encryption scheme. Goyal, O'Neill and Rao [122] gave a (CPA) $\mathsf{RKA}[\Phi^{\mathrm{poly}(d)}]$-secure symmetric encryption scheme, and the work of Bellare et al. [47] gives $\mathsf{RKA}[\Phi^{\mathrm{poly}(d)}]$-secure IBE, signature, symmetric encryption and public key encryption schemes.

4.4 RKA-KDM Security

As noted earlier, the concept of a joint notion of security for related-key attacks and key-dependent message attacks was first considered by Applebaum [15] (he denoted it RK-KDM security). Applebaum made the assumption that messages were arbitrary-length bitstrings, and that messages of equal length are encrypted to ciphertexts of equal length. We now present a variant of Applebaum's definition, where the dummy message m_0 is chosen randomly from message space $\mathcal{M}$ at the start of the security experiment rather than being set to $0^{|\mathrm{m}_1|}$ (i.e. the length of the 'real' message), since we no longer restrict $\mathcal{M}$ to be only bitstrings. For the scenario where bitstrings are of fixed length, these definitions are equivalent.

4.4.1 Security Definition

The definition we will use for RKA-KDM security is the single-key, multi-query scenario. While definitions for KDM security of public key encryption are normally multi-key (to include key cycles in the admissable class of KDM functions), in the symmetric setting the literature is divided on this choice. While Black et al. [65] regarded a vector of symmetric keys (see Def. 3.1), Halevi and Krawczyk [125] gave a single-key definition. Applebaum's [15] RK-KDM definition is also in the single-key, multi-query scenario. For the purposes of simplicity this chapter will deal with the single-key setting and leave the multi-key scenario as an open problem: a definition is straightforward however secure constructions pose a considerable challenge.

The adversary has access to what we refer to as an RKA-KDM oracle: the adversary sends a related-key deriving function $\varphi \in \Phi$ and a key-dependent message function $\psi \in \Psi$, and will receive an encryption under related key $\varphi(\mathrm{k})$ of either key-dependent message $\psi(\mathrm{k})$ or a random element of the message space. The concern regarding length-regularity of KDM functions (see Section 3.3) is dealt with by the previously mentioned restriction of an a priori fixed (and finite) message space.

Definition 4.3 (RKA-KDM[Φ, Ψ] Security.). *Let* $\Sigma = (\textsf{Pg}, \textsf{Kg}, \textsf{E}, \textsf{D})$ *be a symmetric encryption scheme. Adversary* $\mathcal{A}$ *can make encryption queries by submitting* $(\varphi \in \Phi, \psi \in \Psi)$ *and the responses it receives are detailed by the* $\textsf{LR}_b$ *oracle in Fig. 4.3.*

Then the RKA-KDM*[*Φ, Ψ*] advantage for an adversary* $\mathcal{A}$ *against* Σ *is defined by*

$$\mathbf{Adv}^{\textsf{RKA-KDM}[\Phi,\Psi]}_{\Sigma,\,\mathcal{A}}(\lambda) \overset{def}{=} \left|\sum_{b\in\{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\textsf{RKA-KDM}[\Phi,\Psi]\text{-}b}_{\Sigma,\,\mathcal{A}}(\lambda) = 1\right]\right|$$

where experiment $\mathbf{Exp}^{\textsf{RKA-KDM}[\Phi,\Psi]\text{-}b}_{\Sigma,\,\mathcal{A}}$ *is given in Fig. 4.3.*

```
$\mathbf{Exp}^{\textsf{RKA-KDM}[\Phi,\Psi]\text{-}b}_{\Sigma,\,\mathcal{A}}(\lambda)$ :
    $\mathrm{pp} \leftarrow \textsf{Pg}(\lambda)$
    $\mathrm{k} \leftarrow \textsf{Kg}(\mathrm{pp})$
    $\mathrm{m}_0 \overset{\$}{\leftarrow} \mathcal{M}$
    $b' \leftarrow \mathcal{A}^{\textsf{LR}_b}(\mathrm{pp})$
    return $b'$
```

```
$\textsf{LR}_b(\varphi, \psi)$ :
    if $\varphi \notin \Phi$ or $\psi \notin \Psi$ then
        return $\lightning$
    $\mathrm{m}_1 \leftarrow \psi(\mathrm{k})$
    $\mathrm{c} \leftarrow \textsf{E}_{\varphi(\mathrm{k})}(\mathrm{m}_b)$
    return c
```

Figure 4.3: *The experiment defining* RKA-KDM*[*Φ, Ψ*] security for symmetric encryption.*

If the above definition considers only constant functions as the class of KDM functions Ψ then we get the well-established definition of RKA security (Def. 4.2), and restricting the RKA function class Φ to the identity function yields standard KDM security. If we apply both restrictions at once we get real-or-random IND-CPA security. In fact we need to ensure that the constant functions in Ψ span the entire message space, i.e. $\psi_{\mathrm{m}} = \mathrm{m} \in \Psi$ for all $\mathrm{m} \in \mathcal{M}$.

Applebaum [15] gives a construction that is 'LIN RK-KDM' secure for $\mathcal{M} = \{0,1\}^*$ and $\mathcal{K} = \{0,1\}^\lambda$, meaning for

$$\Phi^{\mathrm{xrka}} := \{\varphi_\Delta : \mathcal{K} \to \mathcal{K},\ \mathrm{k} \mapsto \Delta \oplus \mathrm{k} : \Delta \in \mathcal{K}\}$$
$$\Psi^{\mathrm{lin}} := \{\psi_{\mathrm{m},b} : \mathcal{K} \to \mathcal{M},\ \mathrm{k} \mapsto \mathrm{m} \oplus (b \cdot \mathrm{k}) : \mathrm{m} \in \mathcal{M}, b \in \{0,1\}\}$$

with a note to say that if m is longer than the length of the key then $b \cdot \mathrm{k}$ is padded with zeros, and if it is shorter then m is padded.

4.4.2 A Generic Construction

In Thm. 4.1 we will prove that an SKE scheme Σ is $\mathsf{RKA\text{-}KDM}[\Phi, \Psi]$ secure if the following two conditions are simultaneously met:

- Σ is $\mathsf{IND\text{-}KDM\text{-}CPA}$ secure,
- there is what we call an $\mathsf{RKA}[\Phi]$ transformer[2] (defined below in Def. 4.4) for Σ that, when given $\mathsf{E}_{\mathrm{k}}(\mathrm{m})$ and RKA function $\varphi \in \Phi$ as input, returns a value that is indistinguishable from $\mathsf{E}_{\varphi(\mathrm{k})}(\mathrm{m})$ without knowledge of the encryption key k.

This means that the RKA transformer operates without the encryption key k, and that indistinguishability should hold even for an adversary that is given k. An adversary playing the $\mathsf{RKA}[\Phi]$ transformer game is given the encryption key and is asked to distinguish the output of the transformer from a genuine encryption performed under a related key.

These security definitions for the RKA transformer represents a *known-key attack* since the adversary attempting to distinguish the output in each side of the security game is given the encryption key k. There are two alternate notions that we will not elaborate on: a *chosen-key attack* where the adversary selects the encryption key and an *unknown-key attack* where the adversary is not given the key.

Informally, we say that the function $\mathcal{F}_{\mathsf{RKA}[\Phi]}(\varphi, \cdot)$ is an $\mathsf{RKA}[\Phi]$ transformer for Σ iff an adversary $\mathcal{A}$ cannot distinguish the output of this function from the output of $\mathsf{E}_{\varphi(\mathrm{k})}(\cdot)$. Note that a

[2] While the term *transformer* in this description and forthcoming definitions is somewhat imprecise, the terms *simulator*, *oracle* and *machine* are even less desirable. The construct acts like a simulator in the sense that it simulates related-key outputs without the secret key, however using the term simulator *and* giving proofs that involve reductions that simulate an adversarial environment yields some very confusing passages. The work upon which this chapter is based [68, 67] uses the term *oracle*.

valid $\mathsf{RKA}[\Phi]$ transformer does not take k as input, it takes φ and $\mathsf{E}_{\mathsf{k}}(\mathsf{m})$ as input and its output is indistinguishable from $\mathsf{E}_{\varphi(\mathsf{k})}(\mathsf{m})$.

Definition 4.4 ($\mathsf{RKA}[\Phi]$ transformer)**.** *Let $\Sigma = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{E}, \mathsf{D})$ be a secret key encryption scheme. Then the* RKA-$\mathsf{transformer}[\Phi]$ *advantage for an adversary $\mathcal{A}$ against $\mathcal{F}_{\mathsf{RKA}[\Phi]}$ and Σ is defined by*

$$\mathbf{Adv}^{\mathsf{RKA\text{-}transformer}[\Phi]}_{\Sigma,\, \mathcal{F}_{\mathsf{RKA}[\Phi]},\, \mathcal{A}}(\lambda) \overset{def}{=} \left| \sum_{b \in \{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{RKA\text{-}transformer}[\Phi]\text{-}b}_{\Sigma,\, \mathcal{F}_{\mathsf{RKA}[\Phi]},\, \mathcal{A}}(\lambda) = 1\right] \right|$$

where experiment $\mathbf{Exp}^{\mathsf{RKA\text{-}transformer}[\Phi]\text{-}b}_{\Sigma,\, \mathcal{F}_{\mathsf{RKA}[\Phi]},\, \mathcal{A}}$ is given in Fig. 4.4.

```
$\mathbf{Exp}^{\mathsf{RKA\text{-}transformer}[\Phi]\text{-}b}_{\Sigma,\, \mathcal{F}_{\mathsf{RKA}[\Phi]},\, \mathcal{A}}(\lambda)$ :
    $\mathsf{pp} \leftarrow \mathsf{Pg}(\lambda)$
    $\mathsf{k} \leftarrow \mathsf{Kg}(\mathsf{pp})$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathsf{pp}, \mathsf{k})$
    return $b'$
```

```
$\mathsf{LR}_b(\varphi, \mathsf{m})$ :
    if $\varphi \notin \Phi$ then
        return $\lightning$
    $\tilde{\mathsf{c}} \leftarrow \mathsf{E}_{\mathsf{k}}(\mathsf{m})$
    $\mathsf{c}_1 \leftarrow \mathcal{F}_{\mathsf{RKA}[\Phi]}(\varphi, \tilde{\mathsf{c}})$
    $\mathsf{c}_0 \leftarrow \mathsf{E}_{\varphi(\mathsf{k})}(\mathsf{m})$
    return $\mathsf{c}_b$
```

Figure 4.4: *The experiment defining what it means for $\mathcal{F}_{\mathsf{RKA}[\Phi]}$ to be an* RKA-$\mathsf{transformer}[\Phi]$ *for symmetric encryption scheme Σ and RKD function class Φ.*

We now state the main theorem of this chapter. The IND-KDM-CPA security experiment we use here is the 'alternate' notion described in Section 3.3.1 for fixed message spaces.

Theorem 4.1. *Let Σ be an SKE scheme and let $\mathcal{F}_{\mathsf{RKA}[\Phi]}$ be an* $\mathsf{RKA}[\Phi]$ *transformer for Σ. Then the advantage of an adversary $\mathcal{A}$ against* RKA-$\mathsf{KDM}[\Phi, \Psi]$ *security of Σ is*

$$\mathbf{Adv}^{\mathsf{RKA\text{-}KDM}[\Phi,\Psi]}_{\Sigma,\, \mathcal{A}}(\lambda) \leq 2 \cdot \mathbf{Adv}^{\mathsf{RKA\text{-}transformer}[\Phi]}_{\Sigma,\, \mathcal{F}_{\mathsf{RKA}[\Phi]},\, \mathcal{A}_1}(\lambda) + \mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]}_{\Sigma,\, \mathcal{A}_2}(\lambda).$$

where $\mathcal{A}_1$ and $\mathcal{A}_2$ have resources comparable to $\mathcal{A}$, and functions in Φ and Ψ are efficiently computable.

Proof: The proof of this theorem is by a sequence of games, which are detailed in Fig. 4.5.

$\underline{\mathsf{G}_0 :}$ In G_0 the adversary$\mathcal{A}$ plays against $\mathbf{Exp}^{\mathsf{RKA\text{-}KDM}[\Phi,\Psi]\text{-}1}_{\Sigma,\, \mathcal{A}}$, the 'real' side of the RKA-KDM security experiment. Consequently

$$\mathbf{Pr}\left[\mathsf{G}_0{}^{\mathcal{A}} = 1\right] = \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{RKA\text{-}KDM}[\Phi,\Psi]\text{-}1}_{\Sigma,\, \mathcal{A}}(\lambda) = 1\right] . \tag{4.1}$$

$\underline{\mathsf{G}_1 :}$ In G_1, instead of computing $\mathsf{E}_{\varphi(\mathsf{k})}(\psi(\mathsf{k}))$ the experiment (recall that the experiment generates k) computes $\mathsf{c}_{\mathsf{KDM}} \leftarrow \mathsf{E}_{\mathsf{k}}(\psi(\mathsf{k}))$ and outputs $\mathcal{F}_{\mathsf{RKA}[\Phi]}(\varphi, \mathsf{c}_{\mathsf{KDM}})$ to the adversary. This game is indistinguishable from G_0 due to the indistinguishability of $\mathcal{F}_{\mathsf{RKA}[\Phi]}$ (see Definition 4.4). More

```
$\mathbf{Exp}^{\mathsf{RKA\text{-}KDM}[\Phi,\Psi]\text{-}b}_{\Sigma,\,\mathcal{A}}(\lambda)$ :
    $\mathsf{pp} \leftarrow \mathsf{Pg}(\lambda)$
    $\mathsf{k} \leftarrow \mathsf{Kg}(\mathsf{pp})$
    $\mathsf{m}_0 \xleftarrow{\$} \mathcal{M}$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathsf{pp})$
    return $b'$

$\mathsf{LR}_b(\varphi, \psi)$ :
    if $\varphi \notin \Phi$ or $\psi \notin \Psi$ then
        return $\lightning$
    $\boxed{\mathsf{m}_1 \leftarrow \psi(\mathsf{k})}$
    $\boxed{\mathsf{c} \leftarrow \mathsf{E}_{\varphi(\mathsf{k})}(\mathsf{m}_b)}$
    return $\mathsf{c}$
```

```
$\mathsf{G}_0$ :
    $\mathsf{m}_1 \leftarrow \psi(\mathsf{k})$
    $\mathsf{c} \leftarrow \mathsf{E}_{\varphi(\mathsf{k})}(\mathsf{m}_1)$

$\mathsf{G}_1$ :
    $\mathsf{c}_{\mathsf{KDM}} \leftarrow \mathsf{E}_{\mathsf{k}}(\psi(\mathsf{k}))$
    $\mathsf{c} \leftarrow \mathcal{F}_{\mathsf{RKA}[\Phi]}(\varphi, \mathsf{c}_{\mathsf{KDM}})$

$\mathsf{G}_2$ :
    $\mathsf{c}_{\mathsf{KDM}} \leftarrow \mathsf{E}_{\mathsf{k}}(\mathsf{m}_0)$
    $\mathsf{c} \leftarrow \mathcal{F}_{\mathsf{RKA}[\Phi]}(\varphi, \mathsf{c}_{\mathsf{KDM}})$

$\mathsf{G}_3$ :
    $\mathsf{c} \leftarrow \mathsf{E}_{\varphi(\mathsf{k})}(\mathsf{m}_0)$
```

Figure 4.5: *The security games used in the proof of Thm. 4.1. The code for each game G_i is the RKA-KDM security experiment above with the* $\boxed{\text{boxed}}$ *code replaced by the appropriate game code.*

formally, there exists an adversary $\mathcal{A}_1$, detailed in Fig. 4.6 such that

$$\left|\mathbf{Pr}\left[\mathsf{G}_0{}^{\mathcal{A}} = 1\right] \;-\; \mathbf{Pr}\left[\mathsf{G}_1{}^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Adv}^{\mathsf{RKA\text{-}transformer}[\Phi]}_{\Sigma,\,\mathcal{F}_{\mathsf{RKA}[\Phi]},\,\mathcal{A}_1}(\lambda). \tag{4.2}$$

```
$\mathcal{A}_1$ playing $\mathbf{Exp}^{\mathsf{RKA\text{-}transformer}[\Phi]\text{-}b}_{\Sigma,\,\mathcal{F}_{\mathsf{RKA}[\Phi]},\,\mathcal{A}_1}(\lambda)$:
    receive $(\mathsf{pp}, \mathsf{k})$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathsf{pp})$
    return $b'$

$\mathsf{LR}_b(\varphi, \psi)$ :
    $\mathsf{m} \leftarrow \psi(\mathsf{k})$
    call $\mathsf{c}_b \leftarrow \mathsf{LR}_b(\varphi, \mathsf{m})$
    return $\mathsf{c}_b$
```

Figure 4.6: *Description of reduction $\mathcal{A}_1$ used to prove Eqn. 4.2. $\mathcal{A}_1$ runs $\mathcal{A}$ and needs to create an environment $\mathbf{Exp}^{\mathsf{RKA\text{-}KDM}[\Phi,\Psi]\text{-}b}_{\Sigma,\,\mathcal{A}}$ that mimics the hop between G_0 and G_1. Note that for this reduction to be efficient, we need to insist that computing $\psi(\cdot)$ is also efficient.*

$\underline{\mathsf{G}_2 :}$ For G_2, instead of computing $\mathsf{E}_{\mathsf{k}}(\psi(\mathsf{k}))$, the experiment computes $\mathsf{E}_{\mathsf{k}}(\mathsf{m}_0)$. This is the IND-KDM-CPA hop (single-key version of Def. 3.1 in Section 3.3.1), and the reduction $\mathcal{A}_2$ to the IND-KDM-CPA game is detailed in Fig. 4.7.

$$\left|\mathbf{Pr}\left[\mathsf{G}_1{}^{\mathcal{A}} = 1\right] \;-\; \mathbf{Pr}\left[\mathsf{G}_2{}^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]}_{\Sigma,\,\mathcal{A}_2}(\lambda). \tag{4.3}$$

$\underline{\mathsf{G}_3 :}$ In G_3 the experiment computes $\mathsf{E}_{\varphi(\mathsf{k})}(\mathsf{m}_0)$, which is indistinguishable from G_2 due to the indistinguishability of $\mathcal{F}_{\mathsf{RKA}[\Phi]}$. The reduction works in a very similar manner to that of Fig. 4.6,

```
$\mathcal{A}_2$ playing $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA\text{-}}b}_{\Sigma,\ \mathcal{A}_2}(\lambda)$:
    receive pp
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathrm{pp})$
    return $b'$

$\mathsf{LR}_b(\varphi, \psi)$ :
    call $\mathsf{c}_{\mathsf{KDM}} \leftarrow \mathsf{LR}_b(\psi)$
    $\mathsf{c}_b \leftarrow \mathcal{F}_{\mathsf{RKA}[\Phi]}(\varphi, \mathsf{c}_{\mathsf{KDM}})$
    return $\mathsf{c}_b$
```

Figure 4.7: *Description of reduction $\mathcal{A}_2$ used to prove Eqn. 4.3. $\mathcal{A}_2$ runs $\mathcal{A}$ and needs to create an environment $\mathbf{Exp}^{\mathsf{RKA\text{-}KDM}[\Phi,\Psi]\text{-}b}_{\Sigma,\ \mathcal{A}}$ that mimics the hop between G_1 and G_2.*

so we also name this reduction $\mathcal{A}_1$ and detail it in Fig. 4.8. Consequently,

$$\left|\mathbf{Pr}\left[\mathsf{G}_2{}^{\mathcal{A}} = 1\right] \ - \ \mathbf{Pr}\left[\mathsf{G}_3{}^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Adv}^{\mathsf{RKA\text{-}transformer}[\Phi]}_{\Sigma,\ \mathcal{F}_{\mathsf{RKA}[\Phi]},\ \mathcal{A}_1}(\lambda). \tag{4.4}$$

```
$\mathcal{A}_1$ playing $\mathbf{Exp}^{\mathsf{RKA\text{-}transformer}[\Phi]\text{-}b}_{\Sigma,\ \mathcal{F}_{\mathsf{RKA}[\Phi]},\ \mathcal{A}_1}(\lambda)$:
    receive $(\mathrm{pp}, \mathrm{k})$
    $\mathrm{m}_0 \xleftarrow{\$} \mathcal{M}$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathrm{pp})$
    return $b'$

$\mathsf{LR}_b(\varphi, \psi)$ :
    $\mathrm{m} \leftarrow \mathrm{m}_0$
    call $\mathsf{c}_b \leftarrow \mathsf{LR}_b(\varphi, \mathrm{m})$
    return $\mathsf{c}_b$
```

Figure 4.8: *Description of reduction $\mathcal{A}_1$ used to prove Eqn. 4.4. $\mathcal{A}_1$ runs $\mathcal{A}$ and needs to create an environment $\mathbf{Exp}^{\mathsf{RKA\text{-}KDM}[\Phi,\Psi]\text{-}b}_{\Sigma,\ \mathcal{A}}$ that mimics the hop between G_2 and G_3. Again, for this reduction to be efficient we need to insist that computing $\psi(\cdot)$ is also efficient.*

We now note that G_3 represents $\mathbf{Exp}^{\mathsf{RKA\text{-}KDM}[\Phi,\Psi]\text{-}0}_{\Sigma,\ \mathcal{A}}$, the 'fake' side of the RKA-KDM security experiment, i.e.

$$\mathbf{Pr}\left[\mathsf{G}_3{}^{\mathcal{A}} = 1\right] = \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{RKA\text{-}KDM}[\Phi,\Psi]\text{-}0}_{\Sigma,\ \mathcal{A}}(\lambda) = 1\right] \ . \tag{4.5}$$

4.4.3 A Retraction

The original paper [68] made a stronger claim than the one given in Theorem 4.1, and the claim was that the existence of a so-called *KDM transformer* and standard IND-CPA security, in addition to the existence of the RKA transformer, yields RKA-KDM security. The definition of a KDM transformer is given below.

Definition 4.5 ($\mathsf{KDM}[\Psi]$ transformer). *Let $\Sigma = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{E}, \mathsf{D})$ be a secret key encryption scheme. Then the* $\mathsf{KDM\text{-}transformer}[\Psi]$ *advantage for an adversary $\mathcal{A}$ against $\mathcal{F}_{\mathsf{KDM}[\Psi]}$ and Σ is defined by*

$$\mathbf{Adv}^{\mathsf{KDM\text{-}transformer}[\Psi]}_{\Sigma,\ \mathcal{F}_{\mathsf{KDM}[\Psi]},\ \mathcal{A}}(\lambda) \overset{def}{=} \left|\sum_{b\in\{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{KDM\text{-}transformer}[\Psi]\text{-}b}_{\Sigma,\ \mathcal{F}_{\mathsf{KDM}[\Psi]},\ \mathcal{A}}(\lambda) = 1\right]\right|$$

where experiment $\mathbf{Exp}^{\mathsf{KDM\text{-}transformer}[\Psi]\text{-}b}_{\Sigma,\ \mathcal{F}_{\mathsf{KDM}[\Psi]},\ \mathcal{A}}$ is given in Fig. 4.9.

```
$\mathbf{Exp}^{\mathsf{KDM\text{-}transformer}[\Psi]\text{-}b}_{\Sigma,\,\mathcal{F}_{\mathsf{KDM}[\Psi]},\,\mathcal{A}}(\lambda)$ :
    $\mathsf{pp} \leftarrow \mathsf{Pg}(\lambda)$
    $\mathsf{k} \leftarrow \mathsf{Kg}(\mathsf{pp})$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathsf{pp}, \mathsf{k})$
    return $b'$
```

```
$\mathsf{LR}_b(\psi)$ :
    if $\psi \notin \Psi$ then
        return $\lightning$
    $\tilde{\mathsf{c}} \leftarrow \mathsf{E}_{\mathsf{k}}(\psi(0))$
    $\mathsf{c}_1 \leftarrow \mathcal{F}_{\mathsf{KDM}[\Psi]}(\psi, \tilde{\mathsf{c}})$
    $\mathsf{c}_0 \leftarrow \mathsf{E}_{\mathsf{k}}(\psi(\mathsf{k}))$
    return $\mathsf{c}_b$
```

Figure 4.9: The experiment defining what it means for $\mathcal{F}_{\mathsf{KDM}[\Psi]}$ to be a $\mathsf{KDM}[\Psi]$ transformer for symmetric encryption scheme Σ and KDM function class Ψ.

Note that a valid $\mathsf{KDM}[\Psi]$ transformer does not take k as input, it takes ψ and $\mathsf{E}_{\mathsf{k}}(\psi(0))$ as input and its output is indistinguishable from $\mathsf{E}_{\mathsf{k}}(\psi(\mathsf{k}))$. Also note that for constant functions $\psi \in \Psi$ a sufficient behaviour of $\mathcal{F}_{\mathsf{KDM}[\Psi]}$ is to output the ciphertext it received without changes. All $\mathsf{KDM}[\Psi]$ transformers presented henceforth implicitly adopt this behaviour.

The theorem given in the published version, presented here in the concrete security framework, is as follows:

Theorem 4.2 (Retracted). *Let Σ be an SKE scheme that is IND-CPA secure, $\mathcal{F}_{\mathsf{RKA}[\Phi]}$ be an $\mathsf{RKA}[\Phi]$ transformer for Σ and $\mathcal{F}_{\mathsf{KDM}[\Psi]}$ be a $\mathsf{KDM}[\Psi]$ transformer for Σ. Then the advantage of an adversary $\mathcal{A}$ against $\mathsf{RKA\text{-}KDM}[\Phi,\Psi]$ security of Σ is*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{RKA\text{-}KDM}[\Phi,\Psi]}_{\Sigma,\,\mathcal{A}}(\lambda) \quad &\leq \quad 2\cdot\mathbf{Adv}^{\mathsf{RKA\text{-}transformer}[\Phi]}_{\Sigma,\,\mathcal{F}_{\mathsf{RKA}[\Phi]},\,\mathcal{A}_1}(\lambda) + 2\cdot\mathbf{Adv}^{\mathsf{KDM\text{-}transformer}[\Psi]}_{\Sigma,\,\mathcal{F}_{\mathsf{KDM}[\Psi]},\,\mathcal{A}_2}(\lambda)\\ &\quad + \mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Sigma,\,\mathcal{A}_3}(\lambda).\end{aligned}$$

where $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ have resources comparable to $\mathcal{A}$, and functions in Φ and Ψ are efficiently computable.

Initially the problem appeared to be simply with the proof method, however this is not the case: the statement does not hold and this is demonstrated by the following counter-example. Consider $\Phi := \{\varphi_{\mathsf{id}} : \mathcal{K} \to \mathcal{K}, \mathsf{k} \mapsto \mathsf{k}\}$ and $\Psi := \{\psi_{\mathsf{id}} : \mathcal{K} \to \mathcal{M}, \mathsf{k} \mapsto \mathsf{k}\} \cup \{\psi_C : \mathcal{K} \to \mathcal{M}, \mathsf{k} \mapsto C, C \in \mathcal{M}\}$. Let $\Sigma = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{E}, \mathsf{D})$ be some IND-CPA secure encryption scheme where all ciphertexts are of equal length. Define E' as

$$\mathsf{E}'_{\mathsf{k}}(\mathrm{m}) = \begin{cases}\mathsf{E}_k(\mathrm{m})||0 & \text{if } \mathrm{m} \neq \mathrm{k},\\ \mathsf{E}_k(0)||1 & \text{of } \mathrm{m} = \mathrm{k}.\end{cases}$$

Decryption D' is still possible (if ciphertext ends in 0, ignore final bit and apply D, otherwise output k), and $\Sigma' = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{E}', \mathsf{D}')$ is IND-CPA secure. Since the RKA function class Φ is just the identity function, a valid RKA transformer simply outputs whatever it is given as input. A valid KDM transformer takes as input ψ and a valid encryption of $\psi(0)$. For the identity function

$\psi(\mathrm{k}) = \mathrm{k}$ this input is $\mathsf{E}'_{\mathrm{k}}(0) = \mathsf{E}_k(0)||0$ so the transformer just flips the last bit to 1. If ψ is a constant function $\psi(\mathrm{k}) = C$, then this value is $\mathsf{E}'_{\mathrm{k}}(C) = \mathsf{E}_k(C)||0$, and the transformer simply outputs this value. Scheme Σ' is not RKA-KDM secure: an adversary $\mathcal{A}$ sends $(\varphi, \psi) = (\varphi_{\mathsf{id}}, \psi_{\mathsf{id}})$ to the LR oracle and receives $\mathsf{E}'_{\varphi(\mathrm{k})}(\psi(\mathrm{k})) = \mathsf{E}'_{\mathrm{k}}(\mathrm{k}) = \mathsf{E}_{\mathrm{k}}(0)||1$ if $b = 1$ or $\mathsf{E}'_{\varphi(\mathrm{k})}(\mathrm{m}_0) = \mathsf{E}'_{\mathrm{k}}(\mathrm{m}_0) = \mathsf{E}_{\mathrm{k}}(\mathrm{m}_0)||0$ so can use the final bit to distinguish with just one query.

To pinpoint the problem, if we restrict the RKD function class to identity functions in Thm. 4.2 we get the following corollary:

Corollary 4.3 (Retracted). *Let Σ be an SKE scheme that is IND-CPA secure and $\mathcal{F}_{\mathsf{KDM}[\Psi]}$ be a valid* $\mathsf{KDM}[\Psi]$ *transformer for Σ. Then Σ is* $\mathsf{IND\text{-}KDM\text{-}CPA}$ *secure.*

$$\mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]}_{\Sigma,\,\mathcal{A}} \;\leq\; 2 \cdot \mathbf{Adv}^{\mathsf{KDM\text{-}transformer}[\Psi]}_{\Sigma,\,\mathcal{F}_{\mathsf{KDM}[\Psi]},\,\mathcal{A}_1}(\lambda) + \mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Sigma,\,\mathcal{A}_2}(\lambda).$$

Proof: The attempted proof of this corollary is by a sequence of games, which are detailed in Fig. 4.10. The proof breaks down in the hop between G_2 and G_3.

```
$\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]\text{-}b}_{\Sigma,\,\mathcal{A}}(\lambda)$ :
  $\mathrm{pp} \leftarrow \mathsf{Pg}(\lambda)$
  $\mathrm{k} \leftarrow \mathsf{Kg}(\mathrm{pp})$
  $\mathrm{m}_0 \xleftarrow{\$} \mathcal{M}$
  $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathrm{pp})$
  return $b'$

$\mathsf{LR}_b(\psi)$ :
  if $\psi \notin \Psi$ then
    return $\lightning$
  $\boxed{\begin{array}{l} \mathrm{m}_1 \leftarrow \psi(\mathrm{k}) \\ \mathrm{c} \leftarrow \mathsf{E}_{\mathrm{k}}(\mathrm{m}_b) \end{array}}$
  return c
```

```
$\mathsf{G}_0$ :
  $\mathrm{m}_1 \leftarrow \psi(\mathrm{k})$
  $\mathrm{c} \leftarrow \mathsf{E}_{\mathrm{k}}(\mathrm{m}_1)$

$\mathsf{G}_1$ :
  $\tilde{\mathrm{c}} \leftarrow \mathsf{E}_{\mathrm{k}}(\psi(0))$
  $\mathrm{c} \leftarrow \mathcal{F}_{\mathsf{KDM}[\Psi]}(\psi, \tilde{\mathrm{c}})$

$\mathsf{G}_2$ :
  $\tilde{\mathrm{c}} \leftarrow \mathsf{E}_{\mathrm{k}}(\mathrm{m}_0)$
  $\mathrm{c} \leftarrow \mathcal{F}_{\mathsf{KDM}[\Psi]}(\psi, \tilde{\mathrm{c}})$
$\mathsf{G}_3$ :
  $\mathrm{c} \leftarrow \mathsf{E}_{\mathrm{k}}(\mathrm{m}_0)$
```

Figure 4.10: *The security games used in the attempted proof of Corollary 4.3. The code for each game G_i is the* $\mathsf{IND\text{-}KDM\text{-}CPA}$ *security experiment above with the* $\boxed{\text{boxed}}$ *code replaced by the appropriate game code.*

$\underline{\mathsf{G}_0 :}$ In G_0 the adversary$\mathcal{A}$ plays against $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]\text{-}1}_{\Sigma,\,\mathcal{A}}$, the 'real' side of the $\mathsf{IND\text{-}KDM\text{-}CPA}$ security experiment. Consequently

$$\mathbf{Pr}\left[\mathsf{G}_0{}^{\mathcal{A}} = 1\right] = \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]\text{-}b}_{\Sigma,\,\mathcal{A}}(\lambda) = 1\right] . \tag{4.6}$$

$\underline{\mathsf{G}_1 :}$ For G_1, instead of computing $\mathsf{E}_{\mathrm{k}}(\psi(\mathrm{k}))$, the experiment computes $\tilde{\mathrm{c}} \leftarrow \mathsf{E}_{\mathrm{k}}(\psi(0))$ and sets

$c \leftarrow \mathcal{F}_{\mathsf{KDM}[\Psi]}(\psi, \tilde{c})$. Thus there is an adversary $\mathcal{A}_1$ described in Fig. 4.11 such that

$$\left|\mathbf{Pr}\left[\mathsf{G_0}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_1}^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Adv}^{\mathsf{KDM\text{-}transformer}[\Psi]}_{\Sigma,\, \mathcal{F}_{\mathsf{KDM}[\Psi]},\, \mathcal{A}_1}(\lambda). \tag{4.7}$$

```
$\mathcal{A}_1$ playing $\mathbf{Exp}^{\mathsf{KDM\text{-}transformer}[\Psi]\text{-}b}_{\Sigma,\, \mathcal{F}_{\mathsf{KDM}[\Psi]},\, \mathcal{A}_1}(\lambda)$:
    receive $(\mathrm{pp}, \mathrm{k})$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathrm{pp})$
    return $b'$

$\mathsf{LR}_b(\psi)$ :
    call $\mathrm{c}_b \leftarrow \mathsf{LR}_b(\psi)$
    return $\mathrm{c}_b$
```

Figure 4.11: Description of reduction $\mathcal{A}_1$ used to prove Eqn. 4.7. $\mathcal{A}_1$ runs $\mathcal{A}$ and needs to create an environment $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]}_{\Sigma,\, \mathcal{A}}$ that mimics the hop between G_0 and G_1. Note that the reduction $\mathcal{A}_1$ does not need the key k, *meaning that an unknown-key variant of the* KDM-transformer *experiment suffices.*

<u>G_2 :</u> In G_2 we replace $\tilde{c} \leftarrow \mathsf{E}_{\mathrm{k}}(\psi(0))$ by $\tilde{c} \leftarrow \mathsf{E}_{\mathrm{k}}(\mathrm{m}_0)$. This is the IND-CPA hop. To see this, note that an adversary distinguishing between this game and the previous game is trying to distinguish between the encryption of a constant value (picked by the adversary) and the encryption of a random element of $\mathcal{M}$. This is the 'real-or-random' flavour of the IND-CPA game presented in Def. 2.1 of Section 2.2.1. As a result, there exists an adversary $\mathcal{A}_2$ described in Fig. 4.12 such that

$$\left|\mathbf{Pr}\left[\mathsf{G_1}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_2}^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Sigma,\, \mathcal{A}_2}(\lambda). \tag{4.8}$$

```
$\mathcal{A}_2$ playing $\mathbf{Exp}^{\mathsf{IND\text{-}CPA}\text{-}b}_{\Sigma,\, \mathcal{A}_2}(\lambda)$:
    receive pp
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathrm{pp})$
    return $b'$

$\mathsf{LR}_b(\psi)$ :
    $\mathrm{m} \leftarrow \psi(0)$
    call $\tilde{c} \leftarrow \mathsf{LR}_b(\mathrm{m})$
    $\mathrm{c}_b \leftarrow \mathcal{F}_{\mathsf{KDM}[\Psi]}(\psi, \tilde{c})$
    return $\mathrm{c}_b$
```

Figure 4.12: Description of reduction $\mathcal{A}_2$ used to prove Eqn. 4.8. $\mathcal{A}_2$ runs $\mathcal{A}$ and needs to create an environment $\mathbf{Exp}^{\mathsf{RKA\text{-}KDM}[\Phi,\Psi]\text{-}b}_{\Sigma,\, \mathcal{A}}$ that mimics the hop between G_1 *and* G_2.

<u>G_3 :</u> In G_3 the value $\mathcal{F}_{\mathsf{KDM}[\Psi]}(\psi, \mathsf{E}_{\mathrm{k}}(\mathrm{m}_0))$ is replaced by $\mathsf{E}_{\mathrm{k}}(\mathrm{m}_0)$. *This is the phase where the proof breaks down. $\mathcal{A}$ sends ψ to its* LR *oracle and expects either $\mathcal{F}_{\mathsf{KDM}[\Psi]}(\psi, \mathsf{E}_{\mathrm{k}}(\mathrm{m}_0))$ or $\mathsf{E}_{\mathrm{k}}(\mathrm{m}_0)$. If $\mathcal{A}_1$ sends* m_0 *to its own* LR *oracle it will receive either $\mathcal{F}_{\mathsf{KDM}[\Psi]}(\mathrm{m}_0, \mathsf{E}_{\mathrm{k}}(\mathrm{m}_0))$ or $\mathsf{E}_{\mathrm{k}}(\mathrm{m}_0)$.* The reduction $\mathcal{A}_1$ is detailed in Fig. 4.13 which attempts in vain to show

$$\left|\mathbf{Pr}\left[\mathsf{G_2}^{\mathcal{A}} = 1\right] - \mathbf{Pr}\left[\mathsf{G_3}^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Adv}^{\mathsf{KDM\text{-}transformer}[\Psi]}_{\Sigma,\, \mathcal{F}_{\mathsf{KDM}[\Psi]},\, \mathcal{A}_1}(\lambda). \tag{4.9}$$

```
$\mathcal{A}_1$ playing $\mathbf{Exp}^{\mathsf{KDM\text{-}transformer}[\Psi]\text{-}b}_{\Sigma,\,\mathcal{F}_{\mathsf{KDM}[\Psi]},\,\mathcal{A}_1}(\lambda)$:
    receive $(\mathrm{pp}, \mathrm{k})$
    $\mathrm{m}_0 \xleftarrow{\$} \mathcal{M}$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathrm{pp})$
    return $b'$
```

```
$\mathsf{LR}_b(\psi)$ :
    call $\mathrm{c}_b \leftarrow \mathsf{LR}_b(\mathrm{m}_0)$
    return $\mathrm{c}_b$
```

Figure 4.13: *Description of reduction $\mathcal{A}_1$ used in attempted proof of Eqn. 4.9. $\mathcal{A}_1$ runs $\mathcal{A}$ and needs to create an environment $\mathbf{Exp}^{\mathsf{RKA\text{-}KDM}[\Phi,\Psi]\text{-}b}_{\Sigma,\,\mathcal{A}}$ that mimics the hop between G_3 and G_4.*

Now note that G_3 represents $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]\text{-}0}_{\Sigma,\,\mathcal{A}}$, the 'fake' side of the IND-KDM-CPA security experiment, i.e.

$$\mathbf{Pr}\left[\mathsf{G}_3{}^{\mathcal{A}} = 1\right] = \mathbf{Pr}\left[\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]\text{-}0}_{\Sigma,\,\mathcal{A}}(\lambda) = 1\right] . \tag{4.10}$$

While this flaw removes some of the novelty of the result, the constructions detailed later in this chapter are largely unaffected.

4.4.4 Instantiations

We now detail specific instantiations of RKA-KDM-secure encryption schemes. The general approach is to start with a KDM-secure PKE scheme, adapt to the symmetric setting (to yield a clean and rich RKA function class), extract a suitable key-homomorphic property that allows functionality of the RKA transformer, then prove IND-KDM-CPA security of the adapted scheme. Our main contribution in all of these instantiations is the addition (or realisation) of RKA security. The adaptations generally involve modifying the randomness used in encryption to allow construction of a suitable RKA transformer, meaning that the IND-KDM-CPA proofs require either a reduction to the original scheme or in the case of our Malkin et al. construction, a reduction to the interactive vector games used in their proofs.

Scheme	Assm.	$\mathcal{M}$	$\mathcal{K}$	RKA[Φ]	KDM[Ψ]	$\mathcal{C}$
BHHO [72]	DDH	$\{0,1\}^{\mathrm{t}}$	$\{0,1\}^{\mu}$	$\mathrm{k} \oplus \Delta$	Affine[*]	$\mathbb{G}^{\mathrm{t}(\mu+1)}$
BG [74]	QR	$\{0,1\}^{\mathrm{t}}$	$\{0,1\}^{\mu}$	$\mathrm{k} \oplus \Delta$	Affine[*]	$\mathbb{Z}_{\mathrm{N}}^{\mathrm{t}(\mu+1)}$
MTY [159]	DCR	$\mathbb{Z}_{\mathrm{N}}$	$\mathbb{Z}_{\mathrm{N}/4}$	$\mathrm{k} + \Delta$[†]	Affine[‡]	$(\mathbb{Z}_{\mathrm{N}^2}^*)^2$
ACPS [16]	LWE	$\mathbb{Z}_{\mathrm{p}}$	$\{0,1\}^{\mu}$	$\mathrm{k} \oplus \Delta$	$\mathrm{k}_i \oplus b$	$\mathbb{Z}_{\mathrm{q}}^{\mu} \times \mathbb{Z}_{\mathrm{q}}$

[*] $\mathrm{m} = \mathbf{A} \cdot \mathrm{k} \oplus \mathbf{b}$ where $\mathbf{A}$ is a binary $\mathrm{t} \times \mu$ matrix where each row has one 1 and $(\mu - 1)$ zeros, and $\mathbf{b}$ is a bit-vector, and the key is regarded as a bit-vector.

[†] mod $\varphi_{\mathrm{E}}(\mathrm{N})/4$

[‡] $\mathrm{m} = a \cdot \mathrm{k} + b \bmod \mathrm{N}$ where $a, b \in \mathbb{Z}_{\mathrm{N}}$.

Figure 4.14: *Table detailing* RKA-KDM*-secure instantiations. Here Assm. indicates the assumption used to prove* IND-KDM-CPA *security.*

Figure 4.14 describes our instantiations and their relevant properties. Note that for BHHO, BG and ACPS we have $\mathcal{K} = \{0,1\}^{\mu}$ so our (linear) related-key function class is Φ^{xrka}.

In addition to these schemes, in Section 4.4.4.3 we use the approach of Bellare et al. [39] to show that any scheme that has projection-KDM security (meaning KDM functions where each output bit depends on just one key bit) and a suitable RKA transformer achieves RKA-(bounded-)KDM security. By bounded-KDM security we mean functions that can be represented by circuits of some fixed size. This approach uses the amplification technique of Applebaum [13].

Throughout this section we will regard security parameter λ as the value traditionally seen as the security 'goal' for the encryption scheme. This means that if we want 128-bit security then we set our other parameters such as prime sizes and group orders in such a way that $\lambda = 128$.

4.4.4.1 Boneh et al. [72]

As discussed in Section 3.4, the PKE scheme of Boneh et al. [72] was the first provably KDM-secure construction under a standard assumption, namely DDH. We will employ a symmetric-key scheme that is inspired by the 'basic' version of their public key scheme. The class of RKA functions Φ that we obtain allows for XOR operations on the key while the class of KDM functions Ψ allows the adversary to specify an index i and a bit b and receive an encryption of $(\mathrm{k}_i \oplus b)$. There is a mapping from this class to the Boneh et al. scheme's class which is affine functions on the secret key. We construct an $\mathsf{RKA}[\Phi]$ transformer for the scheme, and the $\mathsf{IND\text{-}KDM\text{-}CPA}$ security is based on the DDH assumption.

Recall that Boneh et al.'s (public key) scheme, which we call Π_{BHHO}, has a secret key $(\mathrm{g}^{\mathrm{k}_1}, \ldots, \mathrm{g}^{\mathrm{k}_\mu})$ that is a bitstring encoded as a vector of group elements and a public key consisting of generators $\mathrm{g}_1, \ldots, \mathrm{g}_\mu$ along with $\mathrm{h} \leftarrow (\mathrm{g}_1^{\mathrm{k}_1} \ldots \mathrm{g}_\mu^{\mathrm{k}_\mu})^{-1}$ and uses just one random element $\mathrm{r} \xleftarrow{\$} \mathbb{Z}_\mathrm{p}$ in encryption to give ciphertexts of the form $(\mathrm{g}_1^{\mathrm{r}}, \ldots, \mathrm{g}_\mu^{\mathrm{r}}, \mathrm{m} \cdot \mathrm{h}^{\mathrm{r}})$. Our scheme makes the generators $\mathrm{g}_1, \ldots, \mathrm{g}_\mu$ part of the public parameters , and regards the secret key simply as a bitstring (this will assist construction of our RKA transformer). Our scheme uses μ random elements in $\mathbb{Z}_\mathrm{p}$ for each encryption, which means we need to replace h^{r} as used in Π_{BHHO} by the value $\mathrm{g}_0 \leftarrow \prod_{i \in [\mu]} (\mathrm{g}_i^{\mathrm{r}_i})^{-\mathrm{k}_i}$ to allow the decryption algorithm to 'cancel out' these exponents. Our plaintext messages are 'in the exponent' so we use a message space of bitstrings rather than group elements, meaning our final ciphertext component is $\mathrm{g}_0 \cdot \mathrm{g}^{\mathrm{m}}$ rather than $\mathrm{h}^{\mathrm{r}} \cdot \mathrm{m}$. For the sake of readability we initially introduce the scheme Σ'_{BHHO} with message space $\{0,1\}$. Canonical concatenation, described at the end of this section, will yield the scheme Σ_{BHHO} with message space $\{0,1\}^{\mathrm{t}}$ for some $\mathrm{t} \in \mathbb{Z}$.

Let $\mathbb{G}$ be a group of prime order p and let g be a generator of $\mathbb{G}$. Our scheme Σ'_{BHHO} for

$m \in \{0,1\}$ and ciphertext space $\mathbb{G}^{\mu+1}$ is defined in Fig. 4.15, where $\mu = \lceil 3\log_2 p \rceil$. Since the scheme is discrete-log based, attacks run in approximately $\sqrt{p}$ so for 128-bit security we need $\mu \approx 768$.

```
$\Sigma'_{\mathsf{BHHO}}.\mathsf{Pg}(\lambda)$ :
    $g_1, \ldots, g_\mu \xleftarrow{\$} \mathbb{G} \setminus \{1\}$
    $pp \leftarrow (\mathbb{G}, p, g, g_1, \ldots, g_\mu)$
    return $pp$

$\Sigma'_{\mathsf{BHHO}}.\mathsf{Kg}(pp)$ :
    $k \xleftarrow{\$} \{0,1\}^\mu$
    return $k$
```

```
$\Sigma'_{\mathsf{BHHO}}.\mathsf{E}(m, k)$ :
    $r_1, \ldots, r_\mu \xleftarrow{\$} \mathbb{Z}_p$
    $g_0 \leftarrow \prod_{i \in [\mu]} (g_i^{r_i})^{-k_i}$
    $c \leftarrow (g_1^{r_1}, \ldots, g_\mu^{r_\mu}, g^m \cdot g_0)$
    return $c$

$\Sigma'_{\mathsf{BHHO}}.\mathsf{D}(c, k)$ :
    $(x_1, \ldots, x_\mu, y) \leftarrow c$
    $\tilde{m} \leftarrow y \cdot \prod_{i \in [\mu]} x_i^{k_i}$
    if $\tilde{m} = 1$ then
        return $0$
    if $\tilde{m} = g$ then
        return $1$
    else
        return $\perp$
```

Figure 4.15: Symmetric encryption scheme Σ'_{BHHO} for $\mathcal{M} = \{0,1\}$.

The $\mathsf{RKA}[\Phi]$ **transformer.** The class of RKA functions that we will consider is

$$\Phi := \{\varphi_\Delta : \{0,1\}^\mu \to \{0,1\}^\mu, k \mapsto k \oplus \Delta : \Delta \in \{0,1\}^\mu\}.$$

An $\mathsf{RKA}[\Phi]$ transformer $\mathcal{F}_{\mathsf{RKA}[\Phi]}$ for Σ'_{BHHO} works as follows: Given a ciphertext $c = (x_1, \ldots, x_\mu, y)$ and a function φ_Δ it outputs

$$c' := (x'_1, \ldots, x'_\mu, y') := (x_1^{(-1)^{\Delta_1}}, \ldots, x_\mu^{(-1)^{\Delta_\mu}}, y \cdot \prod_{i \in [\mu]} x_i^{\Delta_i})$$

We assume that ciphertext c is honestly generated—this is a requirement of the indistinguishability experiment for $\mathcal{F}_{\mathsf{RKA}[\Phi]}$. Then we have $y = g^m \cdot \prod_{i \in [\mu]} x_i^{-k_i}$. We observe

$$y' = g^m \cdot \prod_{i \in [\mu]} x_i^{-k_i} \cdot \prod_{i \in [\mu]} x_i^{\Delta_i} = g^m \cdot \prod_{i \in [\mu]} x_i'^{(-1)^{\Delta_i}(-k_i + \Delta_i)} \overset{(*)}{=} g^m \cdot \prod_{i \in [\mu]} x_i'^{-(k_i \oplus \Delta_i)}$$

and $(*)$ holds since

$$(-1)^{\Delta_i}(-k_i + \Delta_i) = \left\{ \begin{array}{r} -k_i \text{ if } \Delta_i = 0 \\ -(1 - k_i) \text{ if } \Delta_i = 1 \end{array} \right\} = -(k_i \oplus \Delta_i)$$

Therefore c' decrypts to m under key $\mathrm{k} \oplus \Delta$, as required.

Lemma 4.4. $\mathcal{F}_{\mathsf{RKA}[\Phi]}$ *is an* $\mathsf{RKA}[\Phi]$ *transformer in the sense of Def. 4.4.*

Proof: The distributions of $\mathcal{F}_{\mathsf{RKA}[\Phi]}(\varphi_\Delta, \mathsf{E}_\mathrm{k}(\mathrm{m}))$ and $\mathsf{E}_{\mathrm{k}\oplus\Delta}(\mathrm{m})$ are perfectly indistinguishable, even for someone knowing k and Δ: The x'_i occur as if $\mathrm{r}'_i = (-1)^{\Delta_i}\mathrm{r}_i$ was used as randomness for the ith component (which yields the same distribution) and we have $\mathrm{y}' = \mathrm{g}^\mathrm{m} \cdot \prod_{i\in[\mu]}(\mathrm{x}'_i)^{-(\mathrm{k}_i\oplus\Delta_i)}$.

KDM security of Σ'_{BHHO}. Our starting point was PKE scheme Π_{BHHO} that is $\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi_{PKE}]$ secure for affine Ψ_{PKE}, so the challenge here is to show that our modified scheme Σ'_{BHHO} retains security against an adversary with access to key-dependent messages of a different form. Intuitively, we first use the hardness of DDH over $\mathbb{G}$ to collapse the randomness used by the encryption oracle to one random exponent per ciphertext, so instead of $\mathrm{r}_1, \ldots, \mathrm{r}_\mu$ all generators are taken to the same random exponent r. This modified scheme resembles the 'basic' version of [72] with a smaller message space, and we then reduce security to the $\mathsf{IND\text{-}KDM\text{-}CPA}$ security of Boneh et al.'s scheme. A major challenge is that the function classes are different, so we require a mapping such that our reduction can convert queries $\psi \in \Psi_{PKE}$ in its own PKE game into $\psi' \in \Psi'_{SKE}$. The KDM function class we consider for Σ'_{BHHO} is

$$\Psi'_{SKE} := \{\psi_{i,b'} : \{0,1\}^\mu \to \{0,1\}, \mathrm{k} \mapsto \mathrm{k}_i \oplus b' : i \in [\mu], b' \in \{0,1\}\}$$

meaning that the adversary can select one key bit at index $i \in [\mu]$ and a bit b' that will be XORed with k_i.

The KDM function class used by Boneh et al., detailed here for single-key(pair) Π_{BHHO}, acts from the encoded secret key $\mathbf{sk} = (\mathrm{g}^{\mathrm{k}_1}, \ldots, \mathrm{g}^{\mathrm{k}_\mu}) \in \mathbb{G}^\mu$ to the message space $\mathbb{G}$

$$\Psi_{PKE} := \{\psi_{\mathbf{u},\mathrm{v}} : \mathbb{G}^\mu \to \mathbb{G}, \mathbf{sk} \mapsto \langle \mathbf{u}; \mathbf{sk}\rangle \cdot \mathrm{v} : \mathbf{u} \in \mathbb{Z}_\mathrm{p}^\mu, \mathrm{v} \in \mathbb{G}\}$$

representing an 'in-the-exponent' inner product[4] of $\mathbf{u}$ and secret key $\mathbf{sk}$, multiplied by a group element v. Note that in the multi-keypair version of Π_{BHHO} the vector $\mathbf{u}$ is allowed to act on a vector of secret keys, which yields circular security.

To avoid notational issues we specify that the adversary's KDM queries, which consist of an index and a bit, will be referred to as $\psi_{i,b'}$ for this proof, and the bit that specifies whether the adversary is interacting with the challenger's real or random world will be denoted b.

[4] Meaning that $\langle(a, -b, 0); (\mathrm{g}^{\mathrm{k}_1}, \mathrm{g}^{\mathrm{k}_2}, \mathrm{g}^{\mathrm{k}_3})\rangle = \mathrm{g}^{a\mathrm{k}_1} \cdot \mathrm{g}^{-b\mathrm{k}_2} \cdot \mathrm{g}^{0\mathrm{k}_3} = \mathrm{g}^{a\mathrm{k}_1 - b\mathrm{k}_2}$ where $a, b \in \mathbb{Z}_\mathrm{p}$ and $\mathrm{g}^0 = 1_\mathbb{G}$ is the neutral element of $\mathbb{G}$.

Lemma 4.5. *The SKE scheme Σ'_{BHHO} is* IND-KDM-CPA$[\Psi'_{SKE}]$ *secure for Ψ'_{SKE} defined above if DDH is hard over the underlying group $\mathbb{G}$. More formally the advantage of an adversary $\mathcal{A}$ against the* IND-KDM-CPA *security of Σ'_{BHHO} is*

$$\mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CPA}[SKE,\Psi'_{SKE}]}_{\Sigma'_{\mathsf{BHHO}},\,\mathcal{A}}(\lambda) \leq (\mu-1)\cdot\mathbf{Adv}^{DDH}_{\mathbb{G},\,\mathcal{A}_1}(\lambda) + \mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CPA}[PKE,\Psi_{PKE}]}_{\Pi_{\mathsf{BHHO}},\,\mathcal{A}_2}(\lambda) \tag{4.11}$$

where $\mathcal{A}_1$ and $\mathcal{A}_2$ have similar resources to $\mathcal{A}$, and $\mathcal{A}_2$ is playing against the PKE version of the IND-KDM-CPA *game.*

Proof: We prove the lemma with the following sequence of games. To aid readability we break from the strategy deployed in many parts of this thesis and define the base game to be $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}[SKE,\Psi'_{SKE}]\text{-}b}_{\Sigma'_{\mathsf{BHHO}},\,\mathcal{A}}$, meaning the challenge bit is selected at random by the challenger at the start of the experiment.

$\underline{\mathsf{G}_0:}$ In G_0, adversary $\mathcal{A}$ plays $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}[SKE,\Psi'_{SKE}]\text{-}b}_{\Sigma'_{\mathsf{BHHO}},\,\mathcal{A}}$, the symmetric key version of the IND-KDM-CPA experiment.

$\underline{\mathsf{G}_1 \text{ to } \mathsf{G}_{\mu-1}}$: Games G_1 to $\mathsf{G}_{\mu-1}$ form a hybrid argument to collapse the randomness used by the encryption oracle. In hybrid i ($i \in [\mu-1]$) we pick the same randomness for the first $i+1$ components of the ciphertext. This means that the format of a ciphertext output by the encryption oracle in game i is

$$\left(\mathrm{g}_1^{\mathrm{r}},\ldots,\mathrm{g}_{i+1}^{\mathrm{r}},\mathrm{g}_{i+2}^{\mathrm{r}_{i+2}},\ldots,\mathrm{g}_{\mu}^{\mathrm{r}_{\mu}},\mathrm{g}^{\mathrm{m}}\cdot\left(\prod_{i\in[i+1]}\mathrm{g}_i^{-\mathrm{r}\mathrm{k}_i}\right)\left(\prod_{i\in[\mu]\backslash[i+1]}\mathrm{g}_i^{-\mathrm{r}_i\mathrm{k}_i}\right)\right)$$

If $b = 1$ then we are in the 'real' side of the KDM experiment so $\mathrm{m} = \mathrm{k}_i \oplus b$, and if $b = 0$ we are in the 'random' side and $\mathrm{m} \xleftarrow{\$} \{0,1\}$.

Analysis. Each hop is indistinguishable due to the hardness of DDH over $\mathbb{G}$ (see Section 2.6.1.2). The reduction for a hop from G_{i-1} to G_i (for $i \in [\mu-1]$) is described in Fig. 4.16 and we insist the challenge bit in the DDH game that $\mathcal{A}_1$ is playing is the same as the challenge bit in the IND-KDM-CPA game that $\mathcal{A}$ is playing:

$$\left|\mathbf{Pr}\left[\mathsf{G}_{i-1}{}^{\mathcal{A}}=1\right] - \mathbf{Pr}\left[\mathsf{G}_i{}^{\mathcal{A}}=1\right]\right| \leq \mathbf{Adv}^{\mathrm{DDH}}_{\mathbb{G},\,\mathcal{A}_1}(\lambda). \tag{4.12}$$

If $\mathrm{Z} = \mathrm{g}^{\mathrm{z}}$, the output of $\mathcal{A}_1$ looks like that of game $i-1$, otherwise (for $\mathrm{Z} = \mathrm{g}^{\mathrm{xy}}$) it looks like that of game i. Any successful distinguisher between those games can thus be used to break DDH. Here we call $\mathcal{A}$'s output b'' to avoid notational conflict.

```
$\mathcal{A}_1$ playing $\mathbf{Exp}^{\mathsf{DDH}\text{-}b}_{\mathbb{G},\,\mathcal{A}_1}(\lambda)$:
    receive $(\mathrm{g}, \mathrm{X}, \mathrm{Y}, \mathrm{Z})$
    for $j \in [\mu] \setminus \{i+1\}$ do
        $\alpha_j \xleftarrow{\$} \mathbb{Z}_\mathrm{p}$
        $\mathrm{g}_j \leftarrow \mathrm{g}^{\alpha_j}$
    $\mathrm{g}_{i+1} \leftarrow \mathrm{X}$
    $\mathrm{k} \xleftarrow{\$} \{0,1\}^\mu$
    $b'' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathbb{G}, \mathrm{g}, \mathrm{g}_1, \ldots, \mathrm{g}_\mu)$
    return $b''$
```

```
$\mathsf{LR}_b(\psi_{i,b'})$ :
    $\mathrm{r}, \mathrm{r}_{i+2}, \ldots, \mathrm{r}_\mu, \mathrm{d}, \mathrm{e} \xleftarrow{\$} \mathbb{Z}_\mathrm{p}$
    $\hat{\mathrm{Y}} \leftarrow \mathrm{g}^\mathrm{d} \cdot \mathrm{Y}^\mathrm{e}$
    $\hat{\mathrm{Z}} \leftarrow \mathrm{X}^\mathrm{d} \cdot \mathrm{Z}^\mathrm{e}$
    $\mathrm{g}_0 \leftarrow \prod_{i \in [\mu]} (\mathrm{g}_i^{\mathrm{r}_i})^{-\mathrm{k}_i}$
    $\mathrm{c}_b \leftarrow (\hat{\mathrm{Y}}^{\mathrm{r}\alpha_1}, \ldots, \hat{\mathrm{Y}}^{\mathrm{r}\alpha_i}, \hat{\mathrm{Z}}^\mathrm{r}, \mathrm{g}_{i+2}^{\mathrm{r}_{i+2}}, \ldots, \mathrm{g}^\mathrm{m} \cdot \mathrm{g}_0)$
    return $\mathrm{c}_b$
```

Figure 4.16: *Description of reduction $\mathcal{A}_1$ used to prove Eqn. 4.12. $\mathcal{A}_1$ receives a DDH challenge $(\mathrm{g}, \mathrm{X}, \mathrm{Y}, \mathrm{Z})$ where $\mathrm{X} = \mathrm{g}^\mathrm{x}$, $\mathrm{Y} = \mathrm{g}^\mathrm{y}$ and either $\mathrm{Z} = \mathrm{g}^{\mathrm{xy}}$ or $\mathrm{Z} = \mathrm{g}^\mathrm{z}$. $\mathcal{A}_1$ runs $\mathcal{A}$ and needs to create an environment $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}\text{-}b}_{\Sigma'_{\mathsf{BHHO}},\,\mathcal{A}}$ that mimics the hop between G_{i-1} and G_i (for $i \in [\mu - 1]$).*

Finally, only one fresh random exponent is used for each ciphertext in game $\mu - 1$, meaning the ciphertext is $(\mathrm{g}_1^\mathrm{r}, \ldots, \mathrm{g}_\mu^\mathrm{r}, \mathrm{g}^\mathrm{m} \cdot (\prod_{i \in [\mu]} \mathrm{g}_i^{-\mathrm{rk}_i}))$. This output now looks like that of the public key BHHO cryptosystem with message space $\{\mathrm{g}^0, \mathrm{g}^1\}$.

$\underline{\mathsf{G}_{\mu-1}:}$ We now bound $\mathcal{A}$'s advantage in game $\mathsf{G}_{\mu-1}$ by that of an adversary $\mathcal{A}_2$ playing against the PKE version of the IND-KDM-CPA game, and the reduction is detailed in Fig. 4.17. Since the KDM functions act differently for our scheme compared to the original scheme, the reduction $\mathcal{A}_2$ needs to convert $\mathcal{A}$'s KDM queries, which take the form (i, b'), to a vector $\mathbf{u}$ and a group element v that it can send to its own KDM left-or-right oracle. This mapping is straightforward for queries of the form (i, b'): if $b' = 0$ then $\mathcal{A}_2$ selects $\mathbf{u} \leftarrow \mathbf{e}_i$ and sets $\mathrm{v} = 1$ (meaning that $\mathcal{A}$ just wants the encryption of k_i) and if $b' = 1$ then $\mathcal{A}_2$ sets $\mathbf{u} \leftarrow -\mathbf{e}_i$ and $\mathrm{v} = \mathrm{g}$ (since $\mathrm{k}_i \oplus 1 = 1 - \mathrm{k}_i$, this invokes a multiplication of $\mathrm{g}^{-\mathrm{k}_i}$ by g).

Again, we call $\mathcal{A}$'s output b'' to avoid notational conflict. Since the previous game hops collapsed the randomness, these replies give a perfect simulation of the SKE version of the IND-KDM-CPA game in $\mathsf{G}_{\mu-1}$ to give the following:

$$\mathbf{Adv}[\mathsf{G}_{\mu-1}](\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CPA}[PKE, \Psi_{PKE}]}_{\Pi_{\mathsf{BHHO}},\,\mathcal{A}_2}(\lambda). \tag{4.13}$$

```
$\mathcal{A}_2$ playing $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}[PKE,\Psi_{PKE}]\text{-}b}_{\Pi_{\mathsf{BHHO}},\,\mathcal{A}_2}(\lambda)$:
    receive $(\mathrm{pk}, \mathrm{pp})$
    $b'' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathrm{pp})$
    return $b''$
```

```
$\mathsf{LR}_b(\psi_{i,b'})$ :
    $\mathbf{u} \leftarrow (-1)^{b'} \mathbf{e}_i$
    $\mathrm{v} \leftarrow \mathrm{g}^{b'}$
    call $\mathrm{c}_b \leftarrow \mathsf{LR}_b(\mathbf{u}, \mathrm{v})$
    return $\mathrm{c}_b$
```

Figure 4.17: *Description of reduction $\mathcal{A}_2$ used to prove Eqn. 4.13. $\mathcal{A}_2$ runs $\mathcal{A}$ and needs to create an environment $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}\text{-}b}_{\Sigma_{\mathsf{BHHO}},\,\mathcal{A}}$ that mimics the game $\mathsf{G}_{\mu-1}$.*

The full scheme Σ_{BHHO}. Finally, we assemble the SKE scheme Σ_{BHHO} from t instances of Σ'_{BHHO} that use the same public parameters pp and the same key k. A ciphertext under Σ_{BHHO} is a matrix from $\mathbb{G}^{t\times(\mu+1)}$ where each row is an instance of Σ'_{BHHO} (using pp and key k). To encrypt a message $m \in \{0,1\}^t$ under key k we encrypt m_i in row i (while picking fresh randomness r_j, $j \in [\mu]$ for each row). Decryption also works row-wise.

For the $\mathsf{RKA}[\Phi]$ transformer we apply $\mathcal{F}_{\mathsf{RKA}[\Phi]}$ to each row. The class of KDM functions Ψ'_{SKE} changes to

$$\Psi_{SKE} := \{\psi_{\mathbf{i},\Delta} : \{0,1\}^{\mu} \to \{0,1\}^{t}, k \mapsto (k_{\mathbf{i}_1} \oplus \Delta_1, \ldots, k_{\mathbf{i}_t} \oplus \Delta_t) : \mathbf{i} \in [\mu]^t, \Delta \in \{0,1\}^t\}$$

meaning that each bit of the message can be an arbitrarily picked key bit. Since the RKA transformer works row-wise it is easy to check that the indistinguishability result from Lemma 4.4 and the proof of IND-KDM-CPA security carries over to the full scheme Σ_{BHHO}. Finally, by Thm. 4.1, we get:

Theorem 4.6. *The SKE scheme Σ_{BHHO} is $\mathsf{RKA\text{-}KDM}[\Phi, \Psi_{SKE}]$ secure (for Φ and Ψ as defined above in this section) if DDH is hard over the underlying group $\mathbb{G}$.*

4.4.4.2 Brakerski-Goldwasser [74]

We now move our attention to the encryption scheme of Brakerski and Goldwasser [74], modified to the symmetric setting. The KDM security of the original (public key) scheme relies on the hardness of deciding quadratic residuosity (see Section 2.6.1) in the group $\mathbb{Z}_N^*$, for Blum integer $N = p \cdot q$. To construct our SKE scheme Σ_{BG} so that it is resilient against related-key attacks, we additionally have to stipulate that DDH is hard over the subgroup of quadratic residues QR_N. We achieve security against the same class of KDM functions as for Σ_{BHHO} from Section 4.4.4.1. In the published paper that resulted from the work in this chapter [68], the novelty in this instantion was the KDM transformer, so now the novelty compared to the BHHO construction is the different hardness assumption.

The SKE scheme Σ'_{BG} is detailed in Fig. 4.18, and we define the scheme for messages $m \in \{0,1\}$ and ciphertext space $\mathbb{Z}_N^{\mu+1}$. We use $\mu(\lambda)$ for the length of the Blum integer since the IND-CPA security of Brakerski and Goldwasser's original scheme requires that N is substantially shorter than the number of components/key length μ, e.g., $\mu(\lambda) = \lambda/2$. Full details of this constraint can be found in [[74], Theorem 6.1]. Brakerski and Goldwasser show that KDM security of their scheme holds for $\mu = \log N + \omega(\log \lambda)$ where $\lambda = \log N$, and if we assume that for 128-bit

```
$\underline{\Sigma'_{\mathsf{BG}}.\mathsf{Pg}(\lambda)}$ :
    $N \xleftarrow{\$} \mathsf{Blum}[\mu(\lambda)]$
    $g_1, \ldots, g_\mu \xleftarrow{\$} \mathsf{QR}_N$
    $pp \leftarrow (N, g_1, \ldots, g_\mu)$
    return pp

$\underline{\Sigma'_{\mathsf{BG}}.\mathsf{Kg}(pp)}$ :
    $k \xleftarrow{\$} \{0,1\}^\mu$
    return k
```

```
$\underline{\Sigma'_{\mathsf{BG}}.\mathsf{E}(m, k)}$ :
    $r_1, \ldots, r_\mu \xleftarrow{\$} [N^2]$
    $g_0 \leftarrow \prod_{i \in [\mu]} (g_i^{r_i})^{-k_i}$
    $c \leftarrow (g_1^{r_1}, \ldots, g_\mu^{r_\mu}, (-1)^m \cdot g_0)$
    return c

$\underline{\Sigma'_{\mathsf{BG}}.\mathsf{D}(c, k)}$ :
    $(x_1, \ldots, x_\mu, y) \leftarrow c$
    $\tilde{m} \leftarrow y \cdot \prod_{i \in [\mu]} x_i^{k_i}$
    if $\tilde{m} = 1$ then
        return 0
    if $\tilde{m} = -1$ then
        return 1
    else
        return $\perp$
```

Figure 4.18: *Symmetric encryption scheme Σ'_{BG} for $\mathcal{M} = \{0, 1\}$.*

security we desire Blum integers of size 3072 bits then $\mu \geq 3072$ which is considerably higher than the BHHO scheme.

The construction Σ'_{BG} bears many similarities to Σ'_{BHHO}, our modified version of BHHO's scheme: the parameter generation selects μ group elements that will be used to mask the randomness used in encryption, and the encryption algorithms use μ random elements instead of just one random element in the original schemes. In Σ'_{BG} the message is represented as a power of (-1) meaning that the decryption algorithm follows a different output procedure.

The $\mathsf{RKA}[\Phi]$ **transformer.** The $\mathsf{RKA}[\Phi]$ transformer $\mathcal{F}_{\mathsf{RKA}[\Phi]}$ for Σ'_{BG} works exactly like the $\mathsf{RKA}[\Phi]$ transformer for Σ'_{BHHO} from Section 4.4.4.1, i.e., Φ allows for transformations of the secret key under XOR. However there is a very small sampling error since the random elements used in encryption are selected from $[N^2]$, and $[N^2] \bmod \frac{\varphi_{\mathsf{E}}(N)}{4} \neq [-N^2] \bmod \frac{\varphi_{\mathsf{E}}(N)}{4}$. This means that in the components affected by the RKA function the 'random' value is very slightly more likely to be at the latter end of the range $\{1, \ldots, \frac{\varphi_{\mathsf{E}}(N)}{4}\}$ and in the unaffected components the randomness is more likely to be a smaller value in this range. This behaviour was observed by Cramer and Shoup [96]. Analogously to Lemma 4.4 we have:

Lemma 4.7. *$\mathcal{F}_{\mathsf{RKA}[\Phi]}$ is an $\mathsf{RKA}[\Phi]$ transformer for Σ'_{BG} in the sense of Def. 4.4.*

KDM security of Σ'_{BG}. The argument for $\mathsf{IND\text{-}KDM\text{-}CPA}$ security of Σ'_{BG} is very similar to the argument used for Σ'_{BHHO} in the previous section; we state the following lemma and remove the scheme-specific KDM function classes for ease of exposition.

Lemma 4.8. *The SKE scheme* Σ'_{BG} *is* IND-KDM-CPA *secure if QR is hard over the underlying group* $\mathbb{Z}_{\mathrm{N}}^*$ *and DDH is hard over the subgroup of quadratic residues* QR_{N}. *More formally the advantage of an adversary* $\mathcal{A}$ *against the* IND-KDM-CPA *security of* Σ'_{BG} *is*

$$\mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CPA}[SKE]}_{\Sigma'_{\mathsf{BG}},\,\mathcal{A}} \leq (\mu - 1) \cdot \mathbf{Adv}^{DDH}_{\mathsf{QR}_{\mathrm{N}},\,\mathcal{A}_1}(\lambda) + \mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CPA}[PKE]}_{\Pi_{\mathsf{BG}},\,\mathcal{A}_2}(\lambda) \tag{4.14}$$

where $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *have similar resources to* $\mathcal{A}$, *and* $\mathcal{A}_2$ *is playing against the PKE version of the* IND-KDM-CPA *game.*

Proof: This proof is analogous to the IND-KDM-CPA proof for Σ'_{BHHO} (see Lemma 4.5). We first collapse the randomness to one random exponent per ciphertext, relying on the hardness of DDH over QR_{N}: this reduction is very similar to Fig. 4.16, sampling from QR_{N} for generators and $[\mathrm{N}^2]$ for exponents rather than $\mathbb{G}$ and $\mathbb{Z}_{\mathrm{p}}$. Subsequently we invoke the IND-KDM-CPA security of Brakerski and Goldwasser's original scheme, however we again need to take into account the discrepancy between the format of the KDM queries in our SKE scenario compared to the PKE setting of the original paper. When the SKE adversary against Σ'_{BG} sends query (i, b'), the reduction (denoted $\mathcal{A}_2$ for the BHHO scheme in Fig. 4.17) needs to compute $\mathbf{u} \leftarrow (-1)^{b'}\mathbf{e}_i$ and $\mathrm{v} \leftarrow (-1)^{b'}$ to form a query $(\mathbf{u}, \mathrm{v})$ that it can send to its own LR oracle. Note that KDM queries that we call $(\mathbf{u}, \mathrm{v})$ (to follow the BHHO notation) are denoted by $(\mathbf{a}, a_0)$ in Brakerski and Goldwasser's paper.

The full scheme Σ_{BG}. Analogously to the setting for BHHO (Section 4.4.4.1), we can canonically construct the full scheme Σ_{BG} for message space $\{0,1\}^{\mathrm{t}}$ from t instances of Σ'_{BG} using the same public parameters and the same key. The class of RKA functions remains the same, while the class of KDM functions automatically extends from Ψ' to

$$\Psi := \{\psi_{\mathbf{i},\Delta} : \{0,1\}^{\mu} \to \{0,1\}^{\mathrm{t}}, \mathrm{k} \mapsto (\mathrm{k}_{\mathbf{i}_1} \oplus \Delta_1, \ldots, \mathrm{k}_{\mathbf{i}_{\mathrm{t}}} \oplus \Delta_{\mathrm{t}}) : \mathbf{i} \in [\mu]^{\mathrm{t}}, \Delta \in \{0,1\}^{\mathrm{t}}\}$$

Since we can canonically transfer Lemmas 4.7 and 4.8 from Σ'_{BG} to Σ_{BG} by Thm 4.1 we get the following theorem.

Theorem 4.9. *The SKE scheme* Σ_{BG} *is* RKA-KDM$[\Phi, \Psi]$ *secure (for* Φ *and* Ψ *as defined above in this section) if QR is hard in the underlying group* $\mathbb{Z}_{\mathrm{N}}^*$ *and DDH is hard over the subgroup of quadratic residues* QR_{N}.

4.4.4.3 Bellare et al. [39]

Since Applebaum's work on KDM amplification [13], it is known that projection-KDM security implies bounded-KDM security. Projection-KDM security allows for KDM functions where each output bit depends only on one input bit (key bit). Bounded-KDM security means that the class of KDM functions is the set of all functions that can be represented by a circuit of bounded size L. We refer to this function class as $\Psi_{\mathsf{bnd}(L)}$ from now on. To our knowledge, currently the most efficient way to construct a bounded-KDM-secure scheme from a projection-KDM-secure one is the approach of Bellare, Hoang and Rogaway [39] (henceforth BHR). In this section we observe that their construction also maintains RKA security in our sense. Thus, we can plug all of our projection-KDM-secure schemes (i.e., Σ_{BG}, Σ_{ACPS} and Σ_{BHHO}) into their framework to get RKA-bounded-KDM-secure schemes. Obviously, this result holds for any projection-KDM-secure scheme that is RKA secure (with a suitable transformer in our sense).

(Projective) garbling schemes. What follows is a brief introduction to garbling schemes established by [39]. A *garbling scheme* is a tuple of algorithms

$$(\mathsf{GC}_{\mathsf{garble}}, \mathsf{GC}_{\mathsf{encode}}, \mathsf{GC}_{\mathsf{decode}}, \mathsf{GC}_{\mathsf{eval}}).$$

For simplicity we omit the additional evaluation function from [39] and restrict to inputs of length λ here. The algorithm $\mathsf{GC}_{\mathsf{garble}}$ is probabilistic while the remaining algorithms are deterministic. Given an encoding of the security parameter and a function f, $\mathsf{GC}_{\mathsf{garble}}(\lambda, f)$ outputs the description of a garbled circuit (F, e, d). Here, F is a function mapping garbled inputs to garbled outputs. For example, F could be a circuit in terms of gates and wires together with a garbled table for each gate. The outputs e and d contain information to encode and decode the input and output of F respectively. We say that a garbling scheme is *correct* if

$$\mathsf{GC}_{\mathsf{decode}}(d, \mathsf{GC}_{\mathsf{eval}}(F, \mathsf{GC}_{\mathsf{encode}}(\mathrm{m}, e))) = f(\mathrm{m})$$

for all functions f (from a certain class), inputs $\mathrm{m} \in \{0,1\}^\lambda$ and descriptions $(F, e, d) \leftarrow \mathsf{GC}_{\mathsf{garble}}(\lambda, f)$ of garbled circuits for f.

For our application we need so-called *projective* garbling schemes. Basically, a garbling scheme is *projective* if for all $\mathbf{x} := \mathsf{GC}_{\mathsf{encode}}(e, \mathrm{m})$ and $\mathbf{x}' := \mathsf{GC}_{\mathsf{encode}}(e, \mathrm{m}')$, we have $|\mathbf{x}_i| = |\mathbf{x}'_i|$ for $i \in [\lambda]$ and $\mathbf{x}_i = \mathbf{x}'_i$ for $i \in [\lambda]$ with $\mathrm{m}_i = \mathrm{m}'_i$. One well-known way to construct a projective garbling scheme is to assign a pair of keys to each wire corresponding to low and high voltage (0/1) respectively. Then e is a tuple of pairs of keys and $\mathsf{GC}_{\mathsf{encode}}(\mathrm{m}, e)$ picks the keys from e corresponding to the bits of m.

Furthermore, we say that a garbling scheme is *privacy preserving* if for any two (adversarially chosen) functions f_0, f_1 with the same circuit size and inputs $\mathrm{x}_0, \mathrm{x}_1$ of same length with $f_0(\mathrm{x}_0) = f_1(\mathrm{x}_1)$, no adversary can distinguish

$$(F_0, \mathsf{GC}_{\mathsf{encode}}(e_0, \mathrm{x}_0), d_0) \quad \text{from} \quad (F_1, \mathsf{GC}_{\mathsf{encode}}(e_1, \mathrm{x}_1), d_1)$$

where $(F_b, e_b, d_b) \leftarrow \mathsf{GC}_{\mathsf{garble}}(\lambda, f_b), b \in \{0, 1\}$. We refer to the full version of Bellare et al.'s paper [40] for a more detailed definition.

The construction of BHR. The construction creates a symmetric $\mathsf{KDM}[\Psi_{\mathsf{bnd}(L)}]$-secure encryption scheme $\Sigma_{\mathsf{BHR}} = (\mathsf{Pg}, \mathsf{Kg}, \mathsf{E}, \mathsf{D})$ from any projection-KDM-secure encryption scheme $\Sigma' = (\mathsf{Pg}', \mathsf{Kg}', \mathsf{E}', \mathsf{D}')$ and any privacy preserving projective garbling scheme ($\mathsf{GC}_{\mathsf{garble}}$, $\mathsf{GC}_{\mathsf{encode}}$, $\mathsf{GC}_{\mathsf{decode}}$, $\mathsf{GC}_{\mathsf{eval}}$) as follows.

- $\mathsf{Pg}(\lambda)$ returns $\mathsf{Pg}'(\lambda)$.
- $\mathsf{Kg}(\mathrm{pp})$ returns $\mathsf{Kg}'(\mathrm{pp})$.
- $\mathsf{E}_{\mathrm{k}}(\mathrm{m})$ first generates a garbled circuit for the identity function ID_λ on bitstrings of length λ: $(F, e, d) \leftarrow \mathsf{GC}_{\mathsf{garble}}(\lambda, \mathsf{ID}_\lambda)$. It then encodes the message $\mathbf{x} \leftarrow \mathsf{GC}_{\mathsf{encode}}(e, \mathrm{m})$ (w.l.o.g. $\mathbf{x} \in \{0, 1\}^{\lambda \times \lambda}$). Finally, it outputs the ciphertext $\mathrm{c} \leftarrow (F, d, \mathsf{E}'_{\mathrm{k}}(\mathbf{x}_i))$.
- $\mathsf{D}_{\mathrm{k}}((F, d, (\mathbf{c}_i)_{i \in [\lambda]}))$ first decrypts the keys for the input wires $\mathbf{x}_i \leftarrow \mathsf{D}'_{\mathrm{k}}(\mathbf{c}_i)$ and then evaluates the circuit to compute and output the message $\mathrm{m} \leftarrow \mathsf{GC}_{\mathsf{decode}}(d, \mathsf{GC}_{\mathsf{eval}}(F, \mathbf{x}))$.

An $\mathsf{RKA}[\Phi]$ transformer for Σ_{BHR}. Given an $\mathsf{RKA}[\Phi]$ transformer $\mathcal{F}'_{\mathsf{RKA}[\Phi]}$ for Σ', we can construct an $\mathsf{RKA}[\Phi]$ transformer $\mathcal{F}_{\mathsf{RKA}[\Phi]}$ for Σ_{BHR} (note that we maintain the class of RKA functions). Let $\mathrm{c} = (F, d, (\mathbf{c}_i)_{i \in [\lambda]})$ be an honestly generated ciphertext and $\varphi \in \Phi$ be an RKA function. We define $\mathcal{F}_{\mathsf{RKA}[\Phi]}(\mathrm{c}) := (F, d, (\mathcal{F}'_{\mathsf{RKA}[\Phi]}(\mathbf{c}_i))_{i \in [\lambda]})$. A straightforward hybrid argument over the $\mathbf{c}_i$, based on the indistinguishability of $\mathcal{F}'_{\mathsf{RKA}[\Phi]}$, shows the indistinguishability of $\mathcal{F}_{\mathsf{RKA}[\Phi]}(\mathrm{c})$ in the sense of Def. 4.4.

Theorem 4.10. *Let Σ' be an $\mathsf{RKA\text{-}KDM}[\Phi, \Psi]$-secure SKE scheme with an indistinguishable $\mathsf{RKA}[\Phi]$ transformer $\mathcal{F}_{\mathsf{RKA}[\Phi]}$. If Ψ covers projections, then Σ_{BHR}, as defined above, is an $\mathsf{RKA\text{-}KDM}[\Phi, \Psi_{\mathsf{bnd}(L)}]$-secure SKE for any arbitrary but fixed bound L.*

Proof: Since we are using KDM-secure schemes, the proof simply involves invoking a suitable RKA transformer. Our first game is the real side the original $\mathsf{RKA\text{-}KDM}[\Phi, \Psi]$ experiment (see Def. 4.3). In the next game, we no longer use the secret key itself to answer the RKA part of queries. More concretely, for a given RKA-KDM query (φ, ψ), we compute $\mathrm{c} \leftarrow \mathsf{E}_{\mathrm{k}}(\psi(\mathrm{k}))$ and output $\mathcal{F}_{\mathsf{RKA}[\Phi]}(\varphi, \mathrm{c})$ instead of directly returning $\mathsf{E}_{\varphi(\mathrm{k})}(\psi(\mathrm{k}))$. The indistinguishability of this game hop follows directly from the indistinguishability of the $\mathsf{RKA}[\Phi]$ transformer. Finally, we can simply follow the strategy from [40], Theorem 15, to compute c. This strategy requires that the garbling scheme used to construct Σ_{BHR} is privacy preserving and projective. We can then hop to the fake side of the $\mathsf{RKA\text{-}KDM}[\Phi, \Psi]$ experiment, again bounded by an adversary playing the game against the $\mathsf{RKA}[\Phi]$ transformer.

4.4.4.4 Malkin et al. [159]

We now turn our attention to the work of Malkin, Teranishi and Yung (henceforth MTY) [159], who provide an efficient PKE scheme which is KDM secure with respect to functions computable by polynomial-size modular arithmetic circuits (MACs). We present a symmetric version of their scheme that is $\mathsf{RKA\text{-}KDM}[\Phi, \Psi]$ with respect to the RKA function class of modular addition and affine KDM functions. The security of the scheme relies on the DCR assumption, see Section 2.6.1 for details.

We consider MTY's so-called Cascaded Paillier Elgamal scheme in the case where $\mathrm{s} = 2$ (where s is the exponent of N) and $d = 1$ (the maximum degree of the polynomials used as KDM queries) to reflect affine functions. In this case the scheme for messages $\mathrm{m} \in \mathbb{Z}_{\mathrm{N}}$ and ciphertext space $(\mathbb{Z}^*_{\mathrm{N}^2})^2$ is detailed in Fig. 4.19. The element g is required to have full order $\varphi_{\mathsf{E}}(\mathrm{N})/4$. The decryption algorithm computes $\mathrm{x}^{\mathrm{k}}\mathrm{y} \bmod \mathrm{N}^2 = (1 + \mathrm{N})^{\mathrm{m}}$ and recovers m using the efficient bijection described in the original paper, denoted here by bij.

In the original PKE scheme Π_{MTY} for $\mathrm{s} = 2$ and $d = 1$, the secret key is chosen from a slightly different range $\mathrm{k} \xleftarrow{\$} [2^{\xi} \cdot \lfloor \mathrm{N}/4 \rfloor]$ and the public key is $\mathrm{g}^{\mathrm{k}} \bmod \mathrm{N}^2$. In the $\mathrm{s} = 2$ and $d = 1$ case their encryption algorithm picks two random elements $\mathrm{r}_0, \mathrm{r}_1 \xleftarrow{\$} [\lfloor \mathrm{N}/4 \rfloor]$ rather than just one and computes $(\mathrm{g}^{-\mathrm{r}_1}, \mathrm{g}^{-\mathrm{r}_0} \cdot \mathrm{g}^{\mathrm{k}\mathrm{r}_1}, (1 + \mathrm{N})^{\mathrm{m}} \cdot \mathrm{g}^{\mathrm{k}\mathrm{r}_0})$.

We remark that the original paper [159] extends to a larger class of KDM functions (polynomials of bounded degree), however it does not appear possible to construct an RKA transformer for the larger class. The expanded d-cascaded Paillier ElGamal encryption of m is

$$\mathrm{c} = (\mathrm{c}_{d+1}, \mathrm{c}_d, \ldots, \mathrm{c}_0) = \left(\mathrm{g}^{-\mathrm{r}_d}, \mathrm{g}^{-\mathrm{r}_{d-1}} \cdot \mathrm{g}^{\mathrm{k}\mathrm{r}_d}, \mathrm{g}^{-\mathrm{r}_{d-2}} \cdot \mathrm{g}^{\mathrm{k}\mathrm{r}_{d-1}}, \ldots, (1 + \mathrm{N})^{\mathrm{m}} \cdot \mathrm{g}^{\mathrm{k}\mathrm{r}_0}\right)$$

```
$\Sigma_{\mathsf{MTY}}.\mathsf{Pg}(\lambda)$ :
    $p, q \xleftarrow{\$} \mathsf{SafePrimes}[\lfloor \lambda/2 \rfloor]$
    $N \leftarrow p \cdot q$
    do
        $g \xleftarrow{\$} \mathsf{CR}[N^2]$
    while $\mathsf{ord}(g) \neq \varphi_{\mathsf{E}}(N)/4$
    $pp \leftarrow (N, g)$
    return $pp$

$\Sigma_{\mathsf{MTY}}.\mathsf{Kg}(pp)$ :
    $k \xleftarrow{\$} [\lfloor N/4 \rfloor]$
    return $k$
```

```
$\Sigma_{\mathsf{MTY}}.\mathsf{E}(m, k)$ :
    $r \xleftarrow{\$} [\lfloor N/4 \rfloor]$
    $x \leftarrow g^{-r} \bmod N^2$
    $y \leftarrow (1+N)^m g^{rk} \bmod N^2$
    $c \leftarrow (x, y)$
    return $c$

$\Sigma_{\mathsf{MTY}}.\mathsf{D}(c, k)$ :
    $(x, y) \leftarrow c$
    $\tilde{m} \leftarrow x^k y \bmod N^2$
    $m \leftarrow \mathsf{bij}(\tilde{m})$
    return $m$
```

Figure 4.19: *Symmetric encryption scheme* Σ_{MTY}

and decryption calculates $c_0 c_1^{k} c_2^{k^2} \dots c_{d+1}^{k^{d+1}}$.

The $\mathsf{RKA}[\Phi]$ **transformer.** For the concrete class of RKA functions

$$\Phi := \{\varphi_\Delta(k) := k + \Delta \bmod \varphi_{\mathsf{E}}(N)/4 : \Delta \in \mathbb{Z}\}$$

we find an $\mathsf{RKA}[\Phi]$ transformer $\mathcal{F}_{\mathsf{RKA}[\Phi]}$ for Σ_{MTY} as follows: $\mathcal{F}_{\mathsf{RKA}[\Phi]}(\varphi_\Delta, c)$ parses c as (x, y) and computes $(x, y \cdot x^{-\Delta} \bmod N^2)$.

Lemma 4.11. $\mathcal{F}_{\mathsf{RKA}[\Phi]}$ *is an* $\mathsf{RKA}[\Phi]$ *transformer in the sense of Def. 4.4.*

Proof: Observe that

$$(x, y \cdot x^{-\Delta}) = (g^{-r}, (1+N)^m g^{rk} g^{r\Delta}) = (g^{-r}, (1+N)^m g^{r(k+\Delta)}).$$

Hence, given a valid encryption of m, the output of the transformer is the encryption of m under key $k + \Delta \bmod \varphi_{\mathsf{E}}(N)/4$ and randomness r. Note that the adversary does not know $\varphi_{\mathsf{E}}(N)$ and so cannot compute this function class directly.

The KDM security of our variant of the MTY scheme follows from the analysis in [159]; we provide a formal proof in our notation.

Interactive Vector Lemmas. In a similar manner to Section 4.4.4.2 we utilise the interactive vector lemmas of [74, 159], in particular the DCR case for $s = 2$ from Section 6.1 of the Malkin et al. paper [159].

Definition 4.6. *The advantage of an adversary* $\mathcal{A}$ *in breaking Interactive Vector Game* IV_i *for* $i = 1, 2$ *is* *defined by*

$$\mathbf{Adv}_{\mathcal{A}}^{\mathsf{IV}_i}(\lambda) \overset{def}{=} \left|\sum_{b\in\{0,1\}} (-1)^b \cdot \mathbf{Pr}\left[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{IV}_i\text{-}b}(\lambda) = 1\right]\right|$$

where $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{IV}_1\text{-}b}(\lambda)$ *is given in Fig. 4.20 and* $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{IV}_2\text{-}b}(\lambda)$ *is given in Fig. 4.21.*

```
$\mathbf{Exp}_{\mathcal{A}}^{\mathsf{IV}_1\text{-}b}(\lambda)$ :
    $\mathrm{p},\mathrm{q} \xleftarrow{\$} \mathsf{SafePrimes}[\lfloor\lambda/2\rfloor]$
    $\mathrm{N} \leftarrow \mathrm{p}\cdot\mathrm{q}$
    $\mathrm{g} \xleftarrow{\$} \mathsf{CR}[\mathrm{N}^2]$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathrm{N},\mathrm{g})$
    return $b'$

$\mathsf{LR}_b(\delta)$ :
    if $\delta \notin \mathbb{Z}_\mathrm{N}$ then
        return $\lightning$
    $\mathrm{r} \xleftarrow{\$} [\lfloor\mathrm{N}/4\rfloor]$
    if $b = 1$ then
        $\mathrm{u} \leftarrow (1+\mathrm{N})^{\delta}\mathrm{g}^{\mathrm{r}} \bmod \mathrm{N}^2$
    if $b = 0$ then
        $\mathrm{u} \leftarrow \mathrm{g}^{\mathrm{r}} \bmod \mathrm{N}^2$
    return $u$
```

Figure 4.20: *Interactive Vector Game* IV_1

```
$\mathbf{Exp}_{\mathcal{A}}^{\mathsf{IV}_2\text{-}b}(\lambda)$ :
    $\mathrm{p},\mathrm{q} \xleftarrow{\$} \mathsf{SafePrimes}[\lfloor\lambda/2\rfloor]$
    $\mathrm{N} \leftarrow \mathrm{p}\cdot\mathrm{q}$
    $\mathrm{g},\mathrm{h} \xleftarrow{\$} \mathsf{CR}[\mathrm{N}^2]$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathrm{N},\mathrm{g},\mathrm{h})$
    return $b'$

$\mathsf{LR}_b(\delta,\overline{\delta})$ :
    if $\delta$ or $\overline{\delta} \notin \mathbb{Z}_\mathrm{N}$ then
        return $\lightning$
    $\mathrm{r} \xleftarrow{\$} [\lfloor\mathrm{N}/4\rfloor]$
    if $b = 1$ then
        $\mathrm{u} \leftarrow (1+\mathrm{N})^{\delta}\mathrm{g}^{\mathrm{r}} \bmod \mathrm{N}^2$
        $\mathrm{v} \leftarrow (1+\mathrm{N})^{\overline{\delta}}\mathrm{h}^{\mathrm{r}} \bmod \mathrm{N}^2$
    if $b = 0$ then
        $\mathrm{u} \leftarrow \mathrm{g}^{\mathrm{r}} \bmod \mathrm{N}^2$
        $\mathrm{v} \leftarrow \mathrm{h}^{\mathrm{r}} \bmod \mathrm{N}^2$
    return $(\mathrm{u},\mathrm{v})$
```

Figure 4.21: *Interactive Vector Game* IV_2.

By Lemma 1 of [159], the advantage of any adversary guessing b in IV_k for $k = 1, 2$ is negligible under the DCR assumption. Note that in IV_1 in the original paper [159], g is picked from $\{\mathrm{t}^{4\mathrm{N}} \bmod \mathrm{N}^2 | \mathrm{t} \in \mathbb{Z}_{\mathrm{N}^2}\}$ rather than from $\mathsf{CR}[\mathrm{N}^2]$, which could lead to a bad event if g does not have the correct order. We observe that this only occurs with negligible probability.

KDM security of Σ_{MTY}. We now prove our modified scheme Σ_{MTY} to be KDM secure with respect to affine functions. A reduction to the DCR assumption using the interactive vector games is preferable to a reduction to the original scheme, as we have modified the cardinality of ciphertexts and the randomness used in the original scheme Π_{MTY}. We define affine functions as follows:

$$\Psi := \{\psi_{\alpha,\beta} : \mathbb{Z}_{\lfloor \mathrm{N}/4\rfloor} \to \mathbb{Z}_\mathrm{N}, \mathrm{k} \mapsto \alpha\mathrm{k} + \beta \bmod \mathrm{N} : \alpha, \beta \in \mathbb{Z}_\mathrm{N}\}$$

which gives rise to the following lemma:

Lemma 4.12. *The SKE scheme* Σ_{MTY} *is* IND-KDM-CPA$[\Psi]$ *secure for affine functions* Ψ *if DCR is hard in the underlying group* $\mathbb{Z}^*_{N^2}$. *More formally there exist adversaries* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *such that*

$$\mathbf{Adv}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]}_{\Sigma_{\mathsf{MTY}},\,\mathcal{A}} \leq \mathbf{Adv}^{\mathsf{IV}_1}_{\mathcal{A}_1}(\lambda) + \mathbf{Adv}^{\mathsf{IV}_2}_{\mathcal{A}_2}(\lambda). \tag{4.15}$$

Proof: In contrast to our strategy in Section 4.4.4.1 we will define our base game G_0 as the adversary playing against the $b = 1$ or 'real' side of the security experiment. The proof proceeds in two stages. Firstly we use IV_1 to split the key-dependent encryption $\mathsf{E}_k(\alpha k + \beta)$ across the two ciphertext components, the second will drop the key-dependent component α and replace β by the dummy message m_0 using IV_2.

$\underline{\mathsf{G}_0 :}$ In G_0 adversary $\mathcal{A}$ is playing against the real side of the IND-KDM-CPA security experiment $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]\text{-}1}_{\Sigma_{\mathsf{MTY}},\,\mathcal{A}}$, so $\mathcal{A}$ receives $\mathsf{E}_k(\alpha k + \beta)$ where $\alpha, \beta \in \mathbb{Z}_N$.

$\underline{\mathsf{G}_1 :}$ In G_1 we replace the encryption of the key-dependent message, i.e. $\mathsf{E}_k(\alpha k + \beta) = (g^{-r}, g^{rk}(1+N)^{\alpha k+\beta})$ by $(g^{-r}(1+N)^{\alpha}, g^{rk}(1+N)^{\beta})$, a value that will still decrypt to $\alpha k + \beta$. This is a similar approach to MTY's 'fake mode' of encryption, and is a hop that should go through even when the adversary is given the key. Reduction $\mathcal{A}_1$, detailed in Fig. 4.22, picks a key $k \xleftarrow{\$} [\lfloor N/4 \rfloor]$. For each query $\psi_{\alpha,\beta}$ by the adversary, $\mathcal{A}_1$ sends $-\alpha$ to IV_1 and receives the response $u \leftarrow g^r(1+N)^{-\alpha b}$ where $b \in \{0,1\}$ and $g \in \mathsf{CR}[N^2]$ are picked uniformly by IV_1 for all queries and r is some uniform randomness that is fresh for each query. $\mathcal{A}_1$ sends the ciphertext $c \leftarrow (u^{-1}, u^k \cdot (1+N)^{\alpha k+\beta;})$ to $\mathcal{A}$. For $b = 0$ this is $\left(g^{-r}, g^{rk}(1+N)^{\alpha k+\beta}\right)$ which is a response to the KDM query $\psi_{\alpha,\beta}$ in the real side of the IND-KDM-CPA game (so the same as G_0). For $b = 1$ the output is $\left(g^{-r}(1+N)^{a}, g^{rk}(1+N)^{\beta}\right)$ as desired.

$$\left|\mathbf{Pr}\left[\mathsf{G}_0{}^{\mathcal{A}} = 1\right] \; - \; \mathbf{Pr}\left[\mathsf{G}_1{}^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Adv}^{\mathsf{IV}_1}_{\mathcal{A}_1}(\lambda). \tag{4.16}$$

```
$\mathcal{A}_1$ playing $\mathbf{Exp}^{\mathsf{IV}_1\text{-}b}_{\mathcal{A}_1}(\lambda)$:
    receive $(N, g)$
    $k \xleftarrow{\$} [\lfloor N/4 \rfloor]$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_1}(N, g)$
    return $b'$

$\mathsf{LR}_1(\psi_{\alpha,\beta})$ :
    call $u \leftarrow \mathsf{IV}_1.\mathsf{LR}_b(-\alpha)$
    $c_b \leftarrow (u^{-1}, u^k \cdot (1+N)^{\alpha k+\beta})$
    return $c_b$
```

Figure 4.22: Description of reduction $\mathcal{A}_1$ *used to prove Eqn. 4.16.* $\mathcal{A}_1$ *runs* $\mathcal{A}$ *and needs to create an environment* $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA\text{-}1}}_{\Sigma_{\mathsf{MTY}},\,\mathcal{A}}$ *that mimics the hop between* G_0 *and* G_1.

Note that since $\mathcal{A}_1$ simulates $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA\text{-}1}}_{\Sigma_{\mathsf{MTY}},\,\mathcal{A}}$ for $\mathcal{A}$, reduction $\mathcal{A}_1$ needs to simulate the $b = 1$

case (i.e. key-dependent encryptions) of $\mathcal{A}$'s LR oracle in the $\mathsf{IND\text{-}KDM\text{-}CPA}$ game that $\mathcal{A}$ plays against. $\mathcal{A}_1$'s own LR oracle is hardwired with the challenge bit b in the IV_1 game, so we write $\mathsf{IV}_1.\mathsf{LR}_b$ to emphasise that this outcome depends on b.

$\underline{\mathsf{G}_2:}$ We now use IV_2 to remove the key-dependent component α altogether and replace β by m_0, meaning that in this game the adversary is playing against $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}[\Psi]\text{-}0}_{\Sigma_{\mathsf{MTY}},\mathcal{A}}$. When $\mathcal{A}$ queries an encryption of $\psi_{\alpha,\beta}$, the adversary $\mathcal{A}_2$ playing IV_2 sends $(-\alpha, \beta - \mathrm{m}_0)$ to its IV_2 oracle ($\mathcal{A}_2$ picks m_0 randomly from $\mathbb{Z}_\mathrm{N}$ at the start of the game). It receives the response $(\mathrm{u}, \mathrm{v}) \leftarrow \left(\mathrm{g}^\mathrm{r}(1+\mathrm{N})^{-\alpha b}, \mathrm{h}^\mathrm{r}(1+\mathrm{N})^{b(\beta-\mathrm{m}_0)}\right)$ where $b \in \{0,1\}$, $\mathrm{g}, \mathrm{h} \in \mathsf{CR}[\mathrm{N}^2]$ are picked uniformly by IV_2 for all queries and r is some uniform randomness that is fresh for each query. Since h is of full order $\varphi_\mathsf{E}(\mathrm{N})/4$ there is some $\mathrm{k} \in [\lfloor \mathrm{N}/4 \rfloor]$ such that $\mathrm{h} = \mathrm{g}^\mathrm{k}$.[5] $\mathcal{A}_2$ sends the ciphertext $(\mathrm{u}^{-1}, \mathrm{v}(1+\mathrm{N})^{\mathrm{m}_0})$ to the adversary. For $b = 1$ this is $\left(\mathrm{g}^{-\mathrm{r}}(1+\mathrm{N})^{\alpha}, \mathrm{h}^\mathrm{r}(1+\mathrm{N})^{\beta}\right)$ which is the ciphertext in G_1, and for $b = 0$ this is $\left(\mathrm{g}^{-\mathrm{r}}, \mathrm{h}^\mathrm{r}(1+\mathrm{N})^{\mathrm{m}_0}\right)$, which is a legitimate encryption of m_0. Reduction $\mathcal{A}_2$ is described in Fig. 4.23 and yields the following equation:

$$\left|\mathbf{Pr}\left[\mathsf{G}_1{}^{\mathcal{A}} = 1\right] \ - \ \mathbf{Pr}\left[\mathsf{G}_2{}^{\mathcal{A}} = 1\right]\right| \leq \mathbf{Adv}^{\mathsf{IV}_2}_{\mathcal{A}_2}(\lambda). \tag{4.17}$$

```
$\mathcal{A}_2$ playing $\mathbf{Exp}^{\mathsf{IV}_2\text{-}b}_{\mathcal{A}_2}(\lambda)$:
    receive $(\mathrm{N}, \mathrm{g}, \mathrm{h})$
    $\mathrm{m}_0 \xleftarrow{\$} \mathbb{Z}_\mathrm{N}$
    $b' \leftarrow \mathcal{A}^{\mathsf{LR}_b}(\mathrm{N}, \mathrm{g})$
    return $b'$

$\mathsf{LR}_b(\psi_{\alpha,\beta})$ :
    call $(\mathrm{u}, \mathrm{v}) \leftarrow \mathsf{IV}_2.\mathsf{LR}_b(-\alpha, \beta - \mathrm{m}_0)$
    $\mathrm{c}_b \leftarrow (\mathrm{u}^{-1}, \mathrm{v}(1+\mathrm{N})^{\mathrm{m}_0})$
    return $\mathrm{c}_b$
```

Figure 4.23: *Description of reduction $\mathcal{A}_2$ used to prove Eqn. 4.17. $\mathcal{A}_2$ runs $\mathcal{A}$ and needs to create an environment* $\mathbf{Exp}^{\mathsf{IND\text{-}KDM\text{-}CPA}\text{-}b}_{\Sigma_{\mathsf{MTY}},\mathcal{A}}$ *that mimics the hop between G_1 and G_2.*

Finally, by Lemmas 4.11 and 4.12, and Thm. 4.1, we obtain

Theorem 4.13. *The scheme Σ_{MTY} is $\mathsf{RKA\text{-}KDM}[\Phi, \Psi]$ secure (for Φ and Ψ as defined above in this section) if the DCR assumption holds in $\mathbb{Z}^*_{\mathrm{N}^2}$.*

4.4.4.5 Applebaum et al. [16]

The majority of this section was written by co-author Dennis Hofheinz.

In this section, we present a secret key version of the PKE scheme of Applebaum et al. [16] and prove it RKA-KDM secure. For compatibility with Applebaum et al.'s application, however, we slightly change the space of secret keys from $\mathbb{Z}_\mathrm{p}^\mu$ to $\{0,1\}^\mu$. Our RKA and KDM transformers

[5] There is a small sampling error here, picking the key from $[\lfloor \mathrm{N}/4 \rfloor]$ rather than $\varphi_\mathsf{E}(\mathrm{N})/4$.

allow encryptions under keys $\mathsf{k} \oplus \Delta$ (for arbitrary $\Delta \in \{0,1\}^{\mu}$) of arbitrary components of the secret key. Security is based on the LWE assumption formulated by Regev [**?**] .

For ease of exposition, we do not detail the choices of the following parameters—these can occur as in Applebaum et al. [16] (with adaptations as in Akavia et al. [8] due to the different choice of secret key). Let q be a polynomial in the security parameter λ, and let $\mu > n$ be integers (that may also depend on λ). By χ, we denote a (discretised Gaussian) error distribution with suitable parameters over $\mathbb{Z}_\mathsf{q}$.

Applebaum et al. [16] show that the LWE assumption over $\mathbb{Z}_\mathsf{q} = \mathbb{Z}_{\mathsf{p}^2}$ and with $\mathbf{s} \leftarrow \mathbb{Z}_\mathsf{p}^n$ is equivalent to Regev's Learning With Errors (LWE) assumption (for $\mathsf{q} = \mathsf{p}$). Furthermore, Akavia et al. [8] show that the LWE assumption with $\mathbf{s} \leftarrow \{0,1\}^n$ is implied by the LWE assumption as above (for different parameters of n, μ). In the following, we will consider $\mathsf{q} = \mathsf{p}^2$ and $\mathbf{s} \in \{0,1\}^n$. Furthermore, for $\mathsf{x} \in \mathbb{R}$, we write $\lceil \mathsf{x} \rfloor_\mathsf{p} := \lceil \mathsf{x} + 1/2 \rceil \bmod \mathsf{p}$ for the nearest integer to x modulo p. The scheme Σ'_{ACPS} (with $\mathsf{m} \in \mathbb{Z}_\mathsf{p}$ and ciphertext space $\mathbb{Z}_\mathsf{q}^{\mu} \times \mathbb{Z}_\mathsf{q}$) is defined in Fig. 4.24.

```
$\Sigma'_{\mathsf{ACPS}}.\mathsf{Pg}(\lambda)$ :
    $\mathsf{pp} \leftarrow \perp$
    return $\mathsf{pp}$

$\Sigma'_{\mathsf{ACPS}}.\mathsf{Kg}(\mathsf{pp})$ :
    $\mathbf{s} \xleftarrow{\$} \{0,1\}^{\mu}$
    $\mathsf{k} \leftarrow \mathbf{s}$
    return $\mathsf{k}$
```

```
$\Sigma'_{\mathsf{ACPS}}.\mathsf{E}(\mathsf{m},\mathsf{k})$ :
    $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_\mathsf{q}^{n \times \mu}$
    $\mathbf{r} \xleftarrow{\$} \chi^{\mu}$
    $\mathbf{x} \xleftarrow{\$} \chi^{\mu}$
    $\mathsf{c} \leftarrow (\mathbf{A} \cdot \mathbf{r}, -(\mathbf{s}^T \cdot \mathbf{A} + \mathbf{x}^T) \cdot \mathbf{r} + \mathsf{p} \cdot \mathsf{m})$
    return $\mathsf{c}$

$\Sigma'_{\mathsf{ACPS}}.\mathsf{D}(\mathsf{c},\mathsf{k})$ :
    $(\mathbf{y}, \mathsf{z}) \leftarrow \mathsf{c}$
    $\mathsf{m} \leftarrow \lceil (\langle \mathbf{s}; \mathbf{y} \rangle + \mathsf{z})/\mathsf{p} \rfloor_\mathsf{p}$
    return $\mathsf{m}$
```

Figure 4.24: *Symmetric encryption scheme* Σ'_{ACPS}

Compared to the PKE scheme of Applebaum et al. [16], we choose $\mathbf{s}$ slightly differently, and also choose different $\mathbf{A}, \mathbf{x}$ upon each encryption. We note that correctness holds only with overwhelming probability over the choice of $\mathbf{r}$ and $\mathbf{x}$. In particular, $|\langle \mathbf{x}; \mathbf{r} \rangle| < \mathsf{p}/2$ with overwhelming probability.

The $\mathsf{RKA}[\Phi]$ **transformer.** For the concrete class of RKA functions

$$\Phi := \{\varphi_\Delta : \{0,1\}^{\mu} \to \{0,1\}^{\mu}, \mathsf{k} \mapsto \mathsf{k} \oplus \Delta : \Delta \in \{0,1\}^{\mu}\},$$

we find an $\mathsf{RKA}[\Phi]$ transformer $\mathcal{F}_{\mathsf{RKA}[\Phi]}$ for Σ'_{ACPS} as follows: Given a ciphertext $\mathrm{c} = (\mathbf{y}, \mathrm{z})$ and a function φ_Δ, it outputs

$$\mathrm{c}' := (\mathbf{y}', \mathrm{z}') \qquad \text{with} \qquad {\mathbf{y}'}_i = (-1)^{\Delta_i}\mathbf{y}_i \quad \text{and} \quad \mathrm{z}' = \mathrm{z} + \sum_{i \in [\mu]} \Delta_i \mathbf{y}_i$$

As with the BHHO scheme, a quick calculation shows that c' is a perfectly distributed ciphertext of m under $\mathrm{k} \oplus \Delta$. Thus:

Lemma 4.14. *$\mathcal{F}_{\mathsf{RKA}[\Phi]}$ is an $\mathsf{RKA}[\Phi]$ transformer in the sense of Def. 4.4.*

KDM security of Σ'_{ACPS}.

Lemma 4.15. *The SKE scheme Σ'_{ACPS} is* IND-KDM-CPA *secure if the LWE assumption holds for the respective parameters.*

Proof:[Sketch] Our scheme is essentially the same as that of Applebaum et al. [16], only with a different distribution of $\mathbf{s}$ (for which, by Akavia et al. [8], the LWE assumption is implied by the "regular" LWE assumption). Hence, we only provide a short overview over the proof of Applebaum et al. [16].

First, we substitute all vectors $\mathbf{s}^{\mathrm{T}}\mathbf{A} + \mathbf{x}^{\mathrm{T}}$ used to handle encryption queries with independently and uniformly random vectors $\mathbf{u}^{\mathrm{T}}$. This step can be justified by applying the LWE assumption.

Next, we observe that now encryption has become lossy, in the sense that ciphertexts are statistically (almost) independent of the underlying message. Indeed, by our choice of $\mu > n$, given $\mathbf{Ar}$, the vector $\mathbf{r}$ still has significant min-entropy. Thus, the value $\langle \mathbf{u}; \mathbf{r} \rangle$ used to pad the encrypted message looks (almost) uniformly and independently distributed. At this point, $\mathcal{A}$'s advantage to distinguish real from fake encryptions is statistically close to zero, and IND-KDM-CPA security follows.

The full scheme Σ_{ACPS}. As in the BHHO setting, we can construct the full scheme Σ_{ACPS} with message space $\mathbb{Z}_{\mathrm{p}}^{\mu}$ from μ instances of Σ'_{ACPS} that use the same public parameters and key in a straightforward manner.

Likewise, by transferring Lemmas 4.14 and 4.15 from Σ'_{ACPS} to Σ_{ACPS} and by Thm. 4.1, we get

Theorem 4.16. *The SKE scheme Σ_{ACPS} is $\mathsf{RKA\text{-}KDM}[\Phi, \Psi]$ secure (for Φ as defined above in this section and Ψ from the full BHHO scheme) if the LWE assumption holds for the respective parameters.*

4.5 Conclusions

This chapter has introduced the notion of RKA-KDM security for symmetric encryption, and provided instantiations based on a number of existing KDM-secure schemes. While the work of Applebaum introduced the joint notion, the work presented in this chapter gives a generic framework for construction of schemes secure under the joint notion, and this approach has the potential to incorporate further security notions. This chapter identifies an issue with the published work on which this chapter is based, and gives a more restrictive and less modular generic framework. The issue of modularity could be recovered by casting the definition of KDM security in the 'transformer' approach.

The instantiations are straightforward and require the isolation of key-homomorphic properties of the underlying KDM-secure schemes, allowing simulation of encryptions under related keys.

In terms of open problems that stem from this work, the modular nature of the framework suggests that it is possible to add further primitives to provide schemes that are secure under three or more notions concurrently. Related-randomness security [168] appears a good candidate as it is heavily linked to RKA security, and the rich field of work in selective-opening security also follows the theme of being a non-standard notion however the multiple definitions and contrived constructions represent a significant barrier to progress in our setting. Extensions to chosen-ciphertext security involve heavy machinery such as NIZK proofs, and candidate schemes are not obvious.

Bibliography

[1] 3GPP TS 35.201 v3.1.1. Specification of the 3GPP confidentiality and integrity algorithms, Document 1: f8 and f9 specification. `http://www.3gpp.org/DynaReport/35201.htm`. Accessed: 3rd March 2015. Cited on page 65.

[2] 3GPP TS 35.202 v3.1.1. Specification of the 3GPP confidentiality and integrity algorithms, Document 2: Kasumi specification. `http://www.3gpp.org/DynaReport/35202.htm`. Accessed: 3rd March 2015. Cited on page 65.

[3] Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 15(2):103–127, 2002. Cited on page 31.

[4] Michel Abdalla, Fabrice Benhamouda, and Alain Passelègue. An algebraic framework for pseudorandom functions and applications to related-key security. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, LNCS, pages 388–409. Springer, August 2015. Cited on page 66.

[5] Michel Abdalla, Fabrice Benhamouda, Alain Passelègue, and Kenneth G. Paterson. Related-key security for pseudorandom functions beyond the linear barrier. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 77–94. Springer, August 2014. Cited on pages 65, 66, and 69.

[6] Tolga Acar, Mira Belenkiy, Mihir Bellare, and David Cash. Cryptographic agility and its relation to circular encryption. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 403–422. Springer, May 2010. Cited on pages 32, 59, and 60.

[7] Pedro Adão, Gergei Bana, Jonathan Herzog, and Andre Scedrov. Soundness of formal encryption in the presence of key-cycles. In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, *ESORICS 2005*, volume 3679 of *LNCS*, pages 374–396. Springer, September 2005. Cited on page 32.

[8] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 474–495. Springer, March 2009. Cited on pages 95 and 96.

[9] Martin R. Albrecht, Pooya Farshim, Kenneth G. Paterson, and Gaven J. Watson. On cipher-dependent related-key attacks in the ideal-cipher model. In Antoine Joux, editor, *FSE 2011*, volume 6733 of *LNCS*, pages 128–145. Springer, February 2011. Cited on page 65.

[10] Jacob Alperin-Sheriff and Chris Peikert. Circular and KDM security for identity-based encryption. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 334–352. Springer, May 2012. Cited on page 32.

[11] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 36–54. Springer, August 2009. Cited on page 4.

[12] Hiroaki Anada and Seiko Arita. Identification schemes from key encapsulation mechanisms. In Abderrahmane Nitaj and David Pointcheval, editors, *AFRICACRYPT 11*, volume 6737 of *LNCS*, pages 59–76. Springer, July 2011. Cited on page 22.

[13] Benny Applebaum. Key-dependent message security: Generic amplification and completeness. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 527–546. Springer, May 2011. Cited on pages 31, 80, and 88.

[14] Benny Applebaum. Garbling XOR gates "for free" in the standard model. Cryptology ePrint Archive, Report 2012/516, 2012. `http://eprint.iacr.org/2012/516`. Cited on page 67.

[15] Benny Applebaum. Garbling XOR gates "for free" in the standard model. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 162–181. Springer, March 2013. Cited on pages 64, 66, 67, 70, 71, and 72.

[16] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, August 2009. Cited on pages 79, 94, 95, and 96.

[17] Benny Applebaum, Danny Harnik, and Yuval Ishai. Semantic security under related-key attacks and applications. In Bernard Chazelle, editor, *ICS 2011*, pages 45–60. Tsinghua University Press, January 2011. Cited on page 66.

[18] Benny Applebaum and Eyal Widder. Related-key secure pseudorandom functions: The case of additive attacks. Cryptology ePrint Archive, Report 2014/478, 2014. `http://eprint.iacr.org/2014/478`. Cited on page 66.

[19] Giuseppe Ateniese, Jan Camenisch, Susan Hohenberger, and Breno de Medeiros. Practical group signatures without random oracles. Cryptology ePrint Archive, Report 2005/385, 2005. `http://eprint.iacr.org/2005/385`. Cited on page 58.

[20] Michael Backes, Markus Dürmuth, and Dominique Unruh. OAEP is secure under key-dependent messages. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 506–523. Springer, December 2008. Cited on pages 33, 36, and 44.

[21] Michael Backes, Ankit Malik, and Dominique Unruh. Computational soundness without protocol restrictions. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *ACM CCS 12*, pages 699–711. ACM Press, October 2012. Cited on page 32.

[22] Michael Backes and Birgit Pfitzmann. Symmetric encryption in a simulatable dolev-yao style cryptographic library. In *17th IEEE Computer Security Foundations Workshop, (CSFW-17 2004), 28-30 June 2004, Pacific Grove, CA, USA*, pages 204–218. IEEE Computer Society, 2004. Cited on page 32.

[23] Michael Backes, Birgit Pfitzmann, and Andre Scedrov. Key-dependent message security under active attacks - brsim/uc-soundness of symbolic encryption with key cycles. In *20th IEEE Computer Security Foundations Symposium, CSF 2007, 6-8 July 2007, Venice, Italy*, pages 112–124. IEEE Computer Society, 2007. Cited on pages 32 and 36.

[24] Feng Bao, Robert H. Deng, and Huafei Zhu. Variations of Diffie-Hellman problem. In Sihan Qing, Dieter Gollmann, and Jianying Zhou, editors, *ICICS 03*, volume 2836 of *LNCS*, pages 301–312. Springer, October 2003. Cited on page 62.

[25] Boaz Barak, Iftach Haitner, Dennis Hofheinz, and Yuval Ishai. Bounded key-dependent message security. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 423–444. Springer, May 2010. Cited on pages 31 and 35.

[26] Mihir Bellare. Practice-oriented provable security. In Ivan Damgård, editor, *Lectures on Data Security, Modern Cryptology in Theory and Practice, Summer School, Aarhus, Denmark, July 1998*, volume 1561 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 1998. Cited on page 9.

[27] Mihir Bellare, Alexandra Boldyreva, and Adriana Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 171–188. Springer, May 2004. Cited on page 11.

[28] Mihir Bellare and David Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 666–684. Springer, August 2010. Cited on pages 65, 69, and 70.

[29] Mihir Bellare and David Cash. Pseudorandom functions and permutations provably secure against related-key attacks. Cryptology ePrint Archive, Report 2010/397, 2010. `http://eprint.iacr.org/2010/397`. Cited on page 69.

[30] Mihir Bellare, David Cash, and Sriram Keelveedhi. Ciphers that securely encipher their own keys. In Yan Chen, George Danezis, and Vitaly Shmatikov, editors, *ACM CCS 11*, pages 423–432. ACM Press, October 2011. Cited on page 31.

[31] Mihir Bellare, David Cash, and Rachel Miller. Cryptography secure against related-key attacks and tampering. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 486–503. Springer, December 2011. Cited on pages 65 and 70.

[32] Mihir Bellare, Anand Desai, Eric Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th FOCS*, pages 394–403. IEEE Computer Society Press, October 1997. Cited on page 13.

[33] Mihir Bellare, Rafael Dowsley, Brent Waters, and Scott Yilek. Standard security does not imply security against selective-opening. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 645–662. Springer, April 2012. Cited on page 4.

[34] Mihir Bellare and Shafi Goldwasser. New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 194–211. Springer, August 1990. Cited on page 9.

[35] Mihir Bellare, Roch Guérin, and Phillip Rogaway. XOR MACs: New methods for message authentication using finite pseudorandom functions. In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 15–28. Springer, August 1995. Cited on page 10.

[36] Mihir Bellare and Viet Tung Hoang. Resisting randomness subversion: Fast deterministic and hedged public-key encryption in the standard model. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 627–656. Springer, April 2015. Cited on pages 62 and 66.

[37] Mihir Bellare, Viet Tung Hoang, and Sriram Keelveedhi. Instantiating random oracles via UCEs. Cryptology ePrint Archive, Report 2013/424, 2013. `http://eprint.iacr.org/2013/424`. Cited on page 62.

[38] Mihir Bellare, Viet Tung Hoang, and Sriram Keelveedhi. Instantiating random oracles via UCEs. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 398–415. Springer, August 2013. Cited on page 62.

[39] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *ACM CCS 12*, pages 784–796. ACM Press, October 2012. Cited on pages 80 and 88.

[40] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. Cryptology ePrint Archive, Report 2012/265, 2012. `http://eprint.iacr.org/2012/265`. Cited on pages 89 and 90.

[41] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, April 2009. Cited on page 4.

[42] Mihir Bellare and Sriram Keelveedhi. Authenticated and misuse-resistant encryption of key-dependent data. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 610–629. Springer, August 2011. Cited on page 32.

[43] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 341–358. Springer, August 1994. Cited on page 10.

[44] Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 491–506. Springer, May 2003. Cited on pages 65 and 68.

[45] Mihir Bellare, Sarah Meiklejohn, and Susan Thomson. Key-versatile signatures and applications: RKA, KDM and joint enc/sig. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 496–513. Springer, May 2014. Cited on pages 33 and 65.

[46] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Tatsuaki Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 531–545. Springer, December 2000. Cited on page 12.

[47] Mihir Bellare, Kenneth G. Paterson, and Susan Thomson. RKA security beyond the linear barrier: IBE, encryption and signatures. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 331–348. Springer, December 2012. Cited on pages 66, 67, and 70.

[48] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, November 1993. Cited on pages 11 and 37.

[49] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 232–249. Springer, August 1994. Cited on page 10.

[50] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 399–416. Springer, May 1996. Cited on page 12.

[51] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, May / June 2006. Cited on pages 9 and 45.

[52] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *20th ACM STOC*, pages 1–10. ACM Press, May 1988. Cited on page 8.

[53] Eli Biham. New types of cryptoanalytic attacks using related keys (extended abstract). In Tor Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 398–409. Springer, May 1994. Cited on page 64.

[54] Eli Biham, Orr Dunkelman, and Nathan Keller. Related-key boomerang and rectangle attacks. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 507–525. Springer, May 2005. Cited on page 65.

[55] Eli Biham, Orr Dunkelman, and Nathan Keller. A related-key rectangle attack on the full KASUMI. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 443–461. Springer, December 2005. Cited on page 65.

[56] Eli Biham, Orr Dunkelman, and Nathan Keller. A unified approach to related-key attacks. In Kaisa Nyberg, editor, *FSE 2008*, volume 5086 of *LNCS*, pages 73–96. Springer, February 2008. Cited on page 65.

[57] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 513–525. Springer, August 1997. Cited on pages 63 and 65.

[58] Eleanor Birrell, Kai-Min Chung, Rafael Pass, and Sidharth Telang. Randomness-dependent message security. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 700–720. Springer, March 2013. Cited on page 66.

[59] Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, and Adi Shamir. Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 299–319. Springer, May 2010. Cited on page 65.

[60] Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 1–18. Springer, December 2009. Cited on page 65.

[61] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. Distinguisher and related-key attack on the full AES-256. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 231–249. Springer, August 2009. Cited on page 65.

[62] Allison Bishop, Susan Hohenberger, and Brent Waters. New circular security counterexamples from decision linear and learning with errors. Cryptology ePrint Archive, Report 2015/715, 2015. http://eprint.iacr.org/2015/715. Cited on page 32.

[63] Nir Bitansky and Ran Canetti. On strong simulation and composable point obfuscation. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 520–537. Springer, August 2010. Cited on page 66.

[64] Nir Bitansky, Ran Canetti, and Shai Halevi. Leakage-tolerant interactive protocols. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 266–284. Springer, March 2012. Cited on page 66.

[65] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 62–75. Springer, August 2003. Cited on pages 29, 30, 33, 37, 44, 47, and 71.

[66] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 1–12. Springer, August 1998. Cited on page 2.

[67] Florian Böhl, Gareth T. Davies, and Dennis Hofheinz. RKA-KDM secure encryption from public-key encryption. Cryptology ePrint Archive, Report 2013/653, 2013. http://eprint.iacr.org/2013/653. Cited on pages 5, 63, 64, and 72.

[68] Florian Böhl, Gareth T. Davies, and Dennis Hofheinz. Encryption schemes secure under related-key and key-dependent message attacks. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 483–500. Springer, March 2014. Cited on pages 5, 63, 64, 72, 75, and 85.

[69] Dan Boneh. The decision diffie-hellman problem. In Joe Buhler, editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer, 1998. Cited on page 26.

[70] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, August 2004. Cited on page 26.

[71] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 37–51. Springer, May 1997. Cited on pages 63 and 65.

[72] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 108–125. Springer, August 2008. Cited on pages 30, 31, 38, 79, 80, and 82.

[73] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic PRFs and their applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 410–428. Springer, August 2013. Cited on page 66.

[74] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 1–20. Springer, August 2010. Cited on pages 31, 79, 85, and 91.

[75] Zvika Brakerski, Shafi Goldwasser, and Yael Tauman Kalai. Black-box circular-secure encryption beyond affine functions. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 201–218. Springer, March 2011. Cited on page 31.

[76] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 505–524. Springer, August 2011. Cited on page 32.

[77] Lawrence Brown, Matthew Kwan, Josef Pieprzyk, and Jennifer Seberry. Improving resistance to differential cryptanalysis and the redesign of LOKI. In Hideki Imai, Ronald L. Rivest, and Tsutomu Matsumoto, editors, *ASIACRYPT'91*, volume 739 of *LNCS*, pages 36–50. Springer, November 1993. Cited on page 64.

[78] Christina Brzuska, Pooya Farshim, and Arno Mittelbach. Indistinguishability obfuscation and UCEs: The case of computationally unpredictable sources. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 188–205. Springer, August 2014. Cited on page 62.

[79] Christina Brzuska, Pooya Farshim, and Arno Mittelbach. Random-oracle uninstantiability from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors,

TCC 2015, Part II, volume 9015 of *LNCS*, pages 428–455. Springer, March 2015. Cited on page 62.

[80] Christina Brzuska and Arno Mittelbach. Using indistinguishability obfuscation via UCEs. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 122–141. Springer, December 2014. Cited on page 62.

[81] Christina Brzuska and Arno Mittelbach. Universal computational extractors and the superfluous padding assumption for indistinguishability obfuscation. Cryptology ePrint Archive, Report 2015/581, 2015. `http://eprint.iacr.org/2015/581`. Cited on page 62.

[82] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 351–368. Springer, April 2009. Cited on pages 31, 33, and 39.

[83] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, May 2001. Cited on pages 29, 30, 36, 37, and 38.

[84] Ran Canetti and Marc Fischlin. Universally composable commitments. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 19–40. Springer, August 2001. Cited on page 8.

[85] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 209–218. ACM, 1998. Cited on page 11.

[86] Ran Canetti, Yael Tauman Kalai, Mayank Varia, and Daniel Wichs. On symmetric encryption and point obfuscation. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 52–71. Springer, February 2010. Cited on page 32.

[87] David Cash, Matthew Green, and Susan Hohenberger. New definitions and separations for circular security. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 540–557. Springer, May 2012. Cited on pages 32, 35, 60, and 61.

[88] Jinyong Chang, Rui Xue, and Anling Zhang. The KDM-CCA security of the kurosawa-desmedt scheme. *IEICE Trans. Fundamentals*, 98(4):1032–1037, 2015. Cited on pages 40 and 62.

[89] Sanjit Chatterjee, Alfred Menezes, and Palash Sarkar. Another look at tightness. In Ali Miri and Serge Vaudenay, editors, *SAC 2011*, volume 7118 of *LNCS*, pages 293–319. Springer, August 2012. Cited on page 12.

[90] Clifford Cocks. A note on non-secret encryption. CESG Report, 1973. Cited on page 15.

[91] Hubert Comon-Lundh and Véronique Cortier. How to prove security of communication protocols? A discussion on the soundness of formal models w.r.t. computational ones. In Thomas Schwentick and Christoph Dürr, editors, *28th International Symposium on Theoretical Aspects of Computer Science, STACS 2011, March 10-12, 2011, Dortmund, Germany*, volume 9 of *LIPIcs*, pages 29–44. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011. Cited on page 32.

[92] Ronald Cramer, Yevgeniy Dodis, Serge Fehr, Carles Padró, and Daniel Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 471–488. Springer, April 2008. Cited on page 66.

[93] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer, August 1998. Cited on pages 17, 20, and 26.

[94] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. Cryptology ePrint Archive, Report 2001/085, 2001. `http://eprint.iacr.org/2001/085`. Cited on page 20.

[95] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, April / May 2002. Cited on page 20.

[96] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003. Cited on pages 20 and 86.

[97] Gareth T. Davies and Martijn Stam. KDM security in the hybrid framework. Cryptology ePrint Archive, Report 2013/567, 2013. `http://eprint.iacr.org/2013/567`. Cited on pages 5 and 28.

[98] Gareth T. Davies and Martijn Stam. KDM security in the hybrid framework. In Josh Benaloh, editor, *CT-RSA 2014*, volume 8366 of *LNCS*, pages 461–480. Springer, February 2014. Cited on pages 5 and 28.

[99] Alexander W. Dent. A designer's guide to KEMs. In Kenneth G. Paterson, editor, *9th IMA International Conference on Cryptography and Coding*, volume 2898 of *LNCS*, pages 133–151. Springer, December 2003. Cited on pages 21, 45, and 47.

[100] Alexander W. Dent. Hybrid cryptography. Cryptology ePrint Archive, Report 2004/210, 2004. `http://eprint.iacr.org/2004/210`. Cited on page 18.

[101] Alexander W. Dent. A note on game-hopping proofs. Cryptology ePrint Archive, Report 2006/260, 2006. `http://eprint.iacr.org/2006/260`. Cited on page 10.

[102] Alexander W. Dent. On the equivalence of two models for key-dependent-message encryption. Cryptology ePrint Archive, Report 2009/572, 2009. `http://eprint.iacr.org/2009/572`. Cited on page 36.

[103] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. Cited on pages 2, 15, and 27.

[104] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 232–250. Springer, August 2006. Cited on page 66.

[105] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000. Cited on page 17.

[106] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 423–440. Springer, May 2014. Cited on page 66.

[107] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. In *40th FOCS*, pages 523–534. IEEE Computer Society Press, October 1999. Cited on page 4.

[108] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th FOCS*, pages 293–302. IEEE Computer Society Press, October 2008. Cited on pages 4 and 66.

[109] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In Andrew Chi-Chih Yao, editor, *ICS 2010*, pages 434–452. Tsinghua University Press, January 2010. Cited on page 66.

[110] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985. Cited on page 26.

[111] James H. Ellis. The possibility of secure non-secret digital encryption. CESG Report, 1970. Cited on page 15.

[112] James H. Ellis. The story of non-secret encryption. CESG Report, 1997. Cited on page 15.

[113] Marc Fischlin, Anja Lehmann, Thomas Ristenpart, Thomas Shrimpton, Martijn Stam, and Stefano Tessaro. Random oracles with(out) programmability. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 303–320. Springer, December 2010. Cited on page 11.

[114] David Galindo, Javier Herranz, and Jorge L. Villar. Identity-based encryption with master key-dependent message security and leakage-resilience. In Sara Foresti, Moti Yung, and Fabio Martinelli, editors, *ESORICS 2012*, volume 7459 of *LNCS*, pages 627–642. Springer, September 2012. Cited on page 32.

[115] Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 258–277. Springer, February 2004. Cited on page 65.

[116] Rosario Gennaro and Victor Shoup. A note on an encryption scheme of kurosawa and desmedt. Cryptology ePrint Archive, Report 2004/194, 2004. `http://eprint.iacr.org/2004/194`. Cited on page 55.

[117] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009. Cited on page 32.

[118] David Goldenberg and Moses Liskov. On related-secret pseudorandomness. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 255–272. Springer, February 2010. Cited on page 66.

[119] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th FOCS*, pages 102–115. IEEE Computer Society Press, October 2003. Cited on page 11.

[120] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. Cited on pages 9, 13, and 29.

[121] Michael Gorski and Stefan Lucks. New related-key boomerang attacks on AES. In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *INDOCRYPT 2008*, volume 5365 of *LNCS*, pages 266–278. Springer, December 2008. Cited on page 65.

[122] Vipul Goyal, Adam O'Neill, and Vanishree Rao. Correlated-input secure hash functions. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 182–200. Springer, March 2011. Cited on pages 66, 69, and 70.

[123] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, April 2008. Cited on page 39.

[124] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 202–219. Springer, March 2009. Cited on page 32.

[125] Shai Halevi and Hugo Krawczyk. Security under key-dependent inputs. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS 07*, pages 466–475. ACM Press, October 2007. Cited on pages 31 and 71.

[126] Brett Hemenway and Rafail Ostrovsky. Building lossy trapdoor functions from lossy encryption. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 241–260. Springer, December 2013. Cited on page 66.

[127] Javier Herranz, Dennis Hofheinz, and Eike Kiltz. The kurosawa-desmedt key encapsulation is not chosen-ciphertext secure. Cryptology ePrint Archive, Report 2006/207, 2006. `http://eprint.iacr.org/2006/207`. Cited on page 20.

[128] Felix Heuer, Tibor Jager, Eike Kiltz, and Sven Schäge. On the selective opening security of practical public-key encryption schemes. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 27–51. Springer, March / April 2015. Cited on page 22.

[129] Dennis Hofheinz. Circular chosen-ciphertext security with compact ciphertexts. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 520–536. Springer, May 2013. Cited on pages 31 and 35.

[130] Dennis Hofheinz and Eike Kiltz. Secure hybrid encryption from weakened key encapsulation. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 553–571. Springer, August 2007. Cited on pages 20 and 26.

[131] Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 21–38. Springer, August 2008. Cited on page 12.

[132] Dennis Hofheinz and Dominique Unruh. Towards key-dependent message security in the standard model. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 108–126. Springer, April 2008. Cited on page 30.

[133] Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private circuits II: Keeping secrets in tamperable circuits. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 308–327. Springer, May / June 2006. Cited on page 65.

[134] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481. Springer, August 2003. Cited on pages 65 and 66.

[135] Tetsu Iwata and Tadayoshi Kohno. New security proofs for the 3GPP confidentiality and integrity algorithms. In Bimal K. Roy and Willi Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 427–445. Springer, February 2004. Cited on page 64.

[136] Antoine Joux and Kim Nguyen. Separating decision Diffie-Hellman from computational Diffie-Hellman in cryptographic groups. *Journal of Cryptology*, 16(4):239–247, September 2003. Cited on page 26.

[137] Bhavana Kanukurthi and Leonid Reyzin. An improved robust fuzzy extractor. In Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, editors, *SCN 08*, volume 5229 of *LNCS*, pages 156–171. Springer, September 2008. Cited on page 66.

[138] John Kelsey, Bruce Schneier, and David Wagner. Key-schedule cryptoanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 237–251. Springer, August 1996. Cited on page 64.

[139] John Kelsey, Bruce Schneier, and David Wagner. Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In Yongfei Han, Tatsuaki Okamoto, and Sihan Qing, editors, *ICICS 97*, volume 1334 of *LNCS*, pages 233–246. Springer, November 1997. Cited on page 64.

[140] Joe Kilian and Phillip Rogaway. How to protect DES against exhaustive key search. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 252–267. Springer, August 1996. Cited on page 9.

[141] Joe Kilian and Phillip Rogaway. How to protect DES against exhaustive key search (an analysis of DESX). *Journal of Cryptology*, 14(1):17–35, 2001. Cited on page 9.

[142] Jongsung Kim, Seokhie Hong, and Bart Preneel. Related-key rectangle attacks on reduced AES-192 and AES-256. In Alex Biryukov, editor, *FSE 2007*, volume 4593 of *LNCS*, pages 225–241. Springer, March 2007. Cited on page 65.

[143] Lars R. Knudsen. Cryptanalysis of LOKI91. In Jennifer Seberry and Yuliang Zheng, editors, *AUSCRYPT'92*, volume 718 of *LNCS*, pages 196–208. Springer, December 1993. Cited on page 64.

[144] Neal Koblitz and Alfred J. Menezes. Another look at "provable security". *Journal of Cryptology*, 20(1):3–37, January 2007. Cited on page 9.

[145] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer, August 1999. Cited on page 3.

[146] Venkata Koppula, Kim Ramchen, and Brent Waters. Separations in circular security for arbitrary length key cycles. Cryptology ePrint Archive, Report 2013/683, 2013. `http://eprint.iacr.org/2013/683`. Cited on page 35.

[147] Venkata Koppula, Kim Ramchen, and Brent Waters. Separations in circular security for arbitrary length key cycles. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 378–400. Springer, March 2015. Cited on page 32.

[148] Hugo Krawczyk. LFSR-based hashing and authentication. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 129–139. Springer, August 1994. Cited on page 20.

[149] Steve Kremer, Graham Steel, and Bogdan Warinschi. Security for key management interfaces. In *Proceedings of the 24th IEEE Computer Security Foundations Symposium, CSF 2011, Cernay-la-Ville, France, 27-29 June, 2011*, pages 266–280. IEEE Computer Society, 2011. Cited on page 36.

[150] Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442. Springer, August 2004. Cited on page 20.

[151] Ralf Küsters and Max Tuengerthal. Composition theorems without pre-established session identifiers. In Yan Chen, George Danezis, and Vitaly Shmatikov, editors, *ACM CCS 11*, pages 41–50. ACM Press, October 2011. Cited on page 32.

[152] Peeter Laud and Ricardo Corin. Sound computational interpretation of formal encryption with composed keys. In Jong In Lim and Dong Hoon Lee, editors, *ICISC 03*, volume 2971 of *LNCS*, pages 55–66. Springer, November 2004. Cited on page 32.

[153] Gaëtan Leurent and Phong Q. Nguyen. How risky is the random-oracle model? In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 445–464. Springer, August 2009. Cited on page 11.

[154] Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Improved constructions of PRFs secure against related-key attacks. In Ioana Boureanu, Philippe Owesarski, and Serge Vaudenay, editors, *ACNS 14*, volume 8479 of *LNCS*, pages 44–61. Springer, June 2014. Cited on page 66.

[155] Allison B. Lewko and Brent Waters. Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM CCS 09*, pages 112–120. ACM Press, November 2009. Cited on page 68.

[156] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 31–46. Springer, August 2002. Cited on page 31.

[157] Xianhui Lu, Bao Li, and Dingding Jia. KDM-CCA security from RKA secure authenticated encryption. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 559–583. Springer, April 2015. Cited on page 31.

[158] Stefan Lucks. Ciphers secure against related-key attacks. In Bimal K. Roy and Willi Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 359–370. Springer, February 2004. Cited on page 65.

[159] Tal Malkin, Isamu Teranishi, and Moti Yung. Efficient circuit-size independent public key encryption with KDM security. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 507–526. Springer, May 2011. Cited on pages 31, 79, 90, 91, and 92.

[160] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer, February 2004. Cited on page 11.

[161] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 278–296. Springer, February 2004. Cited on pages 4 and 66.

[162] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *Journal of the ACM*, 51(2):231–262, 2004. Cited on page 68.

[163] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990. Cited on pages 17 and 39.

[164] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 111–126. Springer, August 2002. Cited on page 11.

[165] Tatsuaki Okamoto and David Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In David Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 159–175. Springer, April 2001. Cited on page 18.

[166] Tatsuaki Okamoto, Shigenori Uchiyama, and Eiichiro Fujisaki. EPOC: Efficient probabilistic public-key encryption (submission to p1363a). In *IEEE P1363a*, 1998. Cited on page 18.

[167] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, May 1999. Cited on page 25.

[168] Kenneth G. Paterson, Jacob C. N. Schuldt, and Dale L. Sibborn. Related randomness attacks for public key encryption. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 465–482. Springer, March 2014. Cited on pages 66 and 97.

[169] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 129–140. Springer, August 1992. Cited on page 8.

[170] Krzysztof Pietrzak. Subspace LWE. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 548–563. Springer, March 2012. Cited on page 66.

[171] Baodong Qin, Shengli Liu, and Zhengan Huang. Key-dependent message chosen-ciphertext security of the cramer-shoup cryptosystem. In Colin Boyd and Leonie Simpson, editors, *Information Security and Privacy - 18th Australasian Conference, ACISP 2013, Brisbane, Australia, July 1-3, 2013. Proceedings*, volume 7959 of *Lecture Notes in Computer Science*, pages 136–151. Springer, 2013. Cited on page 40.

[172] Michael O. Rabin. Digital signatures and public key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Massachusetts Institute of Technology, January 1979. Cited on page 2.

[173] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, August 1992. Cited on page 17.

[174] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signature and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, 1978. Cited on page 2.

[175] Phillip Rogaway. Authenticated-encryption with associated-data. In Vijayalakshmi Atluri, editor, *ACM CCS 02*, pages 98–107. ACM Press, November 2002. Cited on page 12.

[176] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 16–31. Springer, December 2004. Cited on page 31.

[177] Ron Rothblum. On the circular security of bit-encryption. Cryptology ePrint Archive, Report 2012/102, 2012. `http://eprint.iacr.org/2012/102`. Cited on page 32.

[178] Hovav Shacham. A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074, 2007. `http://eprint.iacr.org/2007/074`. Cited on page 26.

[179] Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979. Cited on page 8.

[180] Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28(4):656–715, 1949. Cited on pages 1 and 8.

[181] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, May 1997. Cited on page 62.

[182] Victor Shoup. Using hash functions as a hedge against chosen ciphertext attack. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 275–288. Springer, May 2000. Cited on page 20.

[183] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004. `http://eprint.iacr.org/2004/332`. Cited on pages 9 and 45.

[184] Victor Shoup and Rosario Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 1–16. Springer, May / June 1998. Cited on page 17.

[185] Dominique Unruh. Programmable encryption and key-dependent messages. Cryptology ePrint Archive, Report 2012/423, 2012. `http://eprint.iacr.org/2012/423`. Cited on pages 32 and 36.

[186] Hoeteck Wee. Public key encryption against related key attacks. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 262–279. Springer, May 2012. Cited on page 66.

[187] Hoeteck Wee. KDM-security via homomorphic smooth projective hashing. Cryptology ePrint Archive, Report 2015/721, 2015. `http://eprint.iacr.org/2015/721`. Cited on page 31.

[188] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd FOCS*, pages 80–91. IEEE Computer Society Press, November 1982. Cited on page 9.

[189] Scott Yilek. Resettable public-key encryption: How to encrypt on a virtual machine. In Josef Pieprzyk, editor, *CT-RSA 2010*, volume 5985 of *LNCS*, pages 41–56. Springer, March 2010. Cited on page 66.

[190] Wentao Zhang, Lei Zhang, Wenling Wu, and Dengguo Feng. Related-key differential-linear attacks on reduced AES-192. In K. Srinathan, C. Pandu Rangan, and Moti Yung, editors, *INDOCRYPT 2007*, volume 4859 of *LNCS*, pages 73–85. Springer, December 2007. Cited on page 65.