Diss. ETH No. 23415

Non-Malleable Codes and Public-Key Encryption

A thesis submitted to attain the degree of

Doctor of Sciences of ETH Zurich

(Dr. sc. ETH Zurich)

presented by

Sandro Coretti
MSc in Computer Science, ETH Zurich

born on 23 December 1986
citizen of Bregaglia GR, Switzerland

accepted on the recommendation of

Prof. Dr. Ueli Maurer, examiner
Prof. Dr. Yevgeniy Dodis, co-examiner

2016

Acknowledgements

First of all, I would like to express my gratitude to Ueli Maurer, my advisor. His abstract perspective on the field of cryptography and mathematics in general is unique, and I have learned many valuable lessons from him during our research discussions.

Sincere thanks also go to Yevgeniy Dodis for his support and valuable advice and for co-refereeing this thesis.

I owe a debt of gratitude to Martin Hirt, Esther Hänggi, and Stefan Wolf with whom I had the pleasure of working before starting my doctoral studies. Collaborating with them was incredibly inspiring and piqued my interest in research.

Research is a great deal more interesting if it is an experience shared with other people. I thank all of my collaborators on projects both in and outside the scope of this thesis—Björn Tackmann, Daniele Venturi, Vassilis Zikas, Juan Garay, Ran Cohen—and the students I have had the pleasure of supervising during my time as a doctoral student—Pascal Pfister, Liu Zhang Chen-Da, and Patrick Towa.

Life as an ETH employee would have been considerably less comfortable without the flexibility and competence of Beate Bernhard and Denise Spicher, our group and department secretaries, who provided the single most convenient interface—that of a human being—to ETH's administrative side.

I am very grateful to my good friends and co-workers Kfir Barhum, Grégory Demay, and Pavel Raykov. We shared many extraordinary experiences and kept a strong bond during both good and difficult times. I would also like to thank my short-time office mates Gian-Pietro Farina and Arpita Parta, who are among the kindest and warmest people I have ever met. Special thanks also go to Daniel Tschudi, who had to endure me for far longer a time and was a great partner for discussing research but also issues most likely inconsequential to humanity. I am also grateful

to all other former and current members of the Information Security and Cryptography Group at ETH I have had the pleasure of working with for the good times we have had here: Divesh Aggarwal, Joël Alwen, Christian Badertscher, Maria Dubovitskaya, Robert Enderlein, Daniel Jost, Simon Knellwolf, Christoph Lucas, Christian Matt, Gregor Seiler, and Stefano Tessaro.

Being a PhD student can be surprisingly (not in hindsight, of course) harsh occasionally, and it is during such times—and all other times as well—that I could rely on my dear friends. This holds particularly true for Kfir Barhum, for my long-time roommates Luka Cuderman and Matias Thalmann, and for Claudia Jann, Loris Lago, Mia Meng, Fabrizio Pedrun, and Annatina Poltera, the members of the legendary Mahai-Mashi, with all of whom I have shared many unforgettable experiences.

Last but not least, I would like to express my deepest gratitude to all of my family, in particular to my parents Nando and Elisabeth, my sister Eveline, my grandparents Dori and Walter, my godmother Jolanda, my uncle Walter and his wife Wioleta, and my late godfather Fredu, his wife Doris, and their great kids, Gina and Joy. Thank you so much for you love and support. I love you all, and I wish you all the best and peaceful and long lives.

Abstract

A constructive perspective on public-key encryption. *Public-key encryption (PKE)* is an extremely important and fundamental cryptographic primitive. The security of PKE has received much attention in the cryptographic literature, and many security notions for PKE have been proposed.

When a PKE scheme is used in a larger protocol, the security of this protocol is proved by showing a reduction of breaking a certain security property of the PKE scheme to breaking the security of the protocol. Consequently, each protocol requires in principle its own tailor-made security reduction. Moreover, which PKE security notion should be used in a given context is a priori not evident: PKE security notions are usually defined in terms of a certain game that an efficient adversary cannot win with non-negligible advantage; the employed games model the use of the scheme implicitly through oracle access to its algorithms, and the sufficiency for specific applications is neither explicitly stated nor proven.

The first part of this thesis proposes a new approach to investigating the application of PKE, following the constructive cryptography (CC) paradigm of Maurer and Renner [MR11]: The basic use of PKE is to enable confidential communication from a sender A to a receiver B, assuming A is in possession of B's public key. One can distinguish two relevant cases: The (non-confidential) communication channel from A to B can be authenticated (e.g., because messages are signed) or non-authenticated. The application of PKE is shown to provide the construction of a secure channel from A to B from two assumed authenticated channels, one in each direction, or, alternatively, if the channel from A to B is completely insecure, the construction of a confidential channel without authenticity.

The composition theorem of CC implies that the assumed channels can either be physically realized or can themselves be constructed cryptographically, and also that the constructed channels can directly be used

in any applications that require such a channel. In other words, several construction steps can be composed, which guarantees the soundness of this approach and eliminates the need for separate reduction proofs.

In addition to the above, several popular game-based security notions (and variants thereof) are revisited and given constructive semantics by demonstrating which type of construction is achieved by a PKE scheme satisfying which notion.

Domain extension for public-key encryption. One approach towards basing public-key encryption (PKE) schemes on weak and credible assumptions is to build "stronger" or more general schemes generically from "weaker" or more restricted ones. One particular line of work in this context was initiated by Myers and shelat [MS09] and continued by Hohenberger, Lewko, and Waters [HLW12], who investigated *domain extension* for CCA-secure PKE—which is the strongest standard PKE security notion, requiring security against attackers with access to a decryption oracle—that is, they provide constructions of multi-bit CCA-secure PKE from single-bit CCA-secure PKE.

It is well-known that encrypting each bit of a plaintext string independently is not CCA-secure—the resulting scheme is *malleable*. The second part of this thesis investigates whether this malleability can be dealt with using the conceptually simple approach—suggesting itself when one takes a constructive view on the issue of PKE domain extension—of applying a suitable non-malleable code (a notion introduced by Dziembowski *et al.* [DPW10]) to the plaintext and subsequently encrypting the resulting codeword bit-by-bit. The attacker's ability to ask multiple decryption queries requires that the underlying code be *continuously* non-malleable (as defined by Faust *et al.* [FMNV14]). Since, as is also shown in this thesis, this flavor of non-malleability can only be achieved if the code is allowed to "self-destruct," the resulting scheme inherits this property and therefore only achieves a weaker variant of CCA security.

The second main contribution of this thesis is formalizing this notion of so-called *indistinguishability under (chosen-ciphertext) self-destruct attacks* (IND-SDA) as CCA security with the restriction that the decryption oracle stops working once the attacker submits an invalid ciphertext. First, it is shown that the above approach based on non-malleable codes yields a solution to the problem of domain extension for IND-SDA-secure PKE, provided that the underlying code is continuously non-malleable against a *reduced* form of bit-wise tampering. Then, it is proved that the code by [DPW10] is actually already continuously non-malleable against (even

full) bit-wise tampering; this constitutes an *information-theoretically* secure continuously non-malleable code, a technical contribution that may be of independent interest. Compared to the previous approaches to PKE domain extension, the resulting scheme is more efficient and intuitive, at the cost of not achieving full CCA security. This result is also one of the first applications of non-malleable codes in a context other than memory tampering.

Self-destruct attacks. The third contribution of this thesis is a thorough investigation of security against self-destruct attacks. A new security notion for PKE dubbed *non-malleability under (chosen-ciphertext) self-destruct attacks* (NM-SDA), which is a natural generalization of IND-SDA security and the so-called (standard) notion of *non-malleability under chosen-plaintext attacks* (NM-CPA), is introduced. The notions of IND-SDA and NM-CPA are shown to be incomparable, which implies that NM-SDA is a *strictly* stronger notion than either of them. A black-box construction by Choi *et al.* [CDMW08] of NM-CPA PKE from basic IND-CPA PKE is shown to also achieve NM-SDA security. As such, NM-SDA is a strongest PKE security notion currently known to be implied by IND-CPA security.

Finally, the thesis treats the domain-extension problem for NM-SDA PKE. It turns out that the approach based on non-malleable codes outlined above cannot work with standard non-malleable codes. Therefore, a novel notion of non-malleable codes with *secret state* is introduced and a code strong enough for the domain extension is constructed.

Zusammenfassung

Public-Key Encryption: Eine konstruktive Perspektive. Public-Key Encryption (PKE) ist eine extrem wichtige und fundamentale kryptographische Primitive. Der Sicherheit von PKE ist bereits viel Aufmerksamkeit gewidmet worden und es gibt dementsprechend eine grosse Anzahl an Sicherheitsdefinitionen für PKE.

Wird PKE in einem Protokoll verwendet, so muss die Sicherheit des Protokolls durch eine Reduktion vom Brechen der PKE auf das Brechen des Protokolls bewiesen werden. Im Prinzip benötigt man deswegen für jedes neue Protokoll eine massgeschneiderte Sicherheitsreduktion. Zudem ist a priori meist unklar, welcher Typ Sicherheit für einen gegebenen Kontext angemessen ist: PKE-Sicherheitsdefinitionen basieren für gewöhnlich auf sogenannten Games, die effiziente Angreifer nur mit vernachlässigender Wahrscheinlichkeit gewinnen können sollen. Solche Games modellieren die Anwendung von PKE bloss implizit durch sogenannte Orakel und im Allgemeinen wird weder die Anwendbarkeit in bestimmten Szenarien bewiesen, noch wird definiert, was Anwendbarkeit überhaupt bedeutet.

Der erste Teil dieser Arbeit schlägt einen neuen Ansatz, basierend auf dem Konzept der Constructive Cryptography (CC) von Maurer und Renner [MR11], vor: Die Standardanwendung von PKE ist es, vertrauliche Kommunikation zwischen einem Sender A und einem Empfänger B zu ermöglichen, falls B den öffentlichen Schlüssel von A besitzt. Man kann zwei relevante Fälle unterscheiden: Der (nicht-vertrauliche) Kanal von A zu B kann authentifiziert oder nicht-authentifiziert sein. Es wird beweisen, dass die Anwendung von PKE die Konstruktion eines sicheren Kanals von A zu B von zwei angenommenen authentifizierten Kanälen – einem pro Richtung – erreicht oder, falls der Kanal von A zu B vollständig unsicher ist, die Konstruktion eines vertraulichen, jedoch nicht-authentifizierten Kanals.

Das Kompositionstheorem von CC impliziert, dass die angenommenen Kanäle entweder physisch implementiert oder ihrerseits selbst mittels

kryptographischen Protokollen realisiert werden können, und auch, dass die konstruierten Kanäle direkt in jeder Anwendung eingesetzt werden können, die solche Kanäle erfordert. Anders gesagt: Konstruktionsschritte im obigen Sinn können aneinandergefügt – komponiert – werden. Dies macht den auf CC basierenden Ansatz solide und eliminiert die Notwendigkeit von separaten Reduktionsbeweisen.

Zusätzlich zu oben beschriebenem analysiert diese Arbeit mehrere häufig verwendete game-basierte Sicherheitsdefinitionen (sowie Varianten davon) und zeigt, welche Sicherheitsdefinition welche Konstruktion erreicht. Dadurch erhalten die game-basierten Definitionen eine konstruktive Semantik.

Domain Extension für PKE. Eine Art, die Sicherheit von PKE auf schwache und glaubwürdige Annahmen aufzubauen, besteht darin „stärkere“, allgemeinere Verfahren aus „schwächeren“, spezifischeren zu konstruieren. Eine Reihe von Arbeiten, beginnend mit Myers und shelat [MS09] und fortgeführt von Hohenberger, Lewko und Waters [HLW12], untersucht in diesem Sinne das Problem der *Domain Extension* für CCA-Sicherheit, die die stärkste PKE-Sicherheitsstufe darstellt und Sicherheit gegen Angreifer, die Zugriff auf ein sogenanntes Decryption-Orakel haben, erfordert. Domain Extension ist das Problem, aus ein-bit CCA-sicherer PKE, generisch multi-bit CCA-sichere PKE zu erreichen.

Es ist bekannt, dass, wenn jedes Bit einer Nachricht einzeln verschlüsselt wird, das daraus resultierende Verfahren nicht CCA-sicher ist – es ist sogenannt *malleable (verformbar)*. Der zweite Teil dieser Arbeit untersucht daher folgenden, einfachen, von dem CC-Paradigma inspirierten Ansatz: Um eine multi-bit Nachricht zu verschlüsseln, wird diese zunächst mit einem Non-Malleable Code kodiert (ein Konzept, das von Dziembowski *et al.* [DPW10] eingeführt wurde) und anschliessend wird jedes Bit des Codewortes einzeln verschlüsselt. Da der Angreifer Zugriff auf ein Decryption-Orakel hat, muss der zugrundeliegende Code sogenannt *continuously* non-malleable sein (wie von Faust *et al.* [FMNV14] definiert). Die vorliegende Arbeit zeigt, dass diese Art der Non-Malleability jedoch nur erreicht werden kann, wenn der Code einen sogenannten *„self-destruct“*-Modus hat. Dieses Verhalten findet sich im daraus resultierenden PKE-Verfahren wieder, weshalb dieses lediglich eine schwächere CCA-Variante erreicht.

Der zweite Hauptbeitrag dieser Arbeit besteht in der Formalisierung dieses Sicherheitsbegriffs, der sogenannten *indistinguishability under (chosen-ciphertext) self-destruct attacks* (IND-SDA) als CCA-Sicherheit mit der Einschränkung, dass das Decryption-Orakel nur solange funktioniert, bis

der Angreifer die erste ungültige Anfrage macht. Es wird gezeigt, dass der obige auf Non-Malleable Codes basierende Ansatz für die Domain Extension von IND-SDA-sicherer PKE geeignet ist, falls der verwendete Code gegen eine bestimmte Form von *reduziertem* bit-weisem Tampering non-malleable ist. Danach wird bewiesen, dass der Code von [DPW10] bereits continuously non-malleable ist, sogar gegen *vollwertiges* bit-weises Tampering. Dies stellt einen *informations-theoretisch* sicheren continuously non-malleable Code dar, was ein technischer Beitrag von unabhängigem Interesse sein mag. Im Vergleich zu den vorhergehenden Lösungen im Bereich der Domain Extension für PKE ist das hier vorgestellte Verfahren effizienter und intuitiver, erreicht jedoch nicht die volle IND-CCA-Sicherheitsstufe. Das beschriebene Resultat ist ebenfalls eine der ersten Anwendungen von Non-Malleable Codes ausserhalb des Bereichs des sogenannten Memory Tampering.

Self-Destruct-Attacken. Der dritte Beitrag dieser Arbeit ist eine Analyse der Sicherheit unter Self-Destruct-Attacken. Eine zusätzliche Sicherheitsdefinition, genannt *non-malleability under (chosen-ciphertext) self-destruct attacks* (NM-SDA), wird präsentiert. Es handelt sich dabei um eine natürliche Verallgemeinerung von IND-SDA und der sogenannten (in der Literatur bekannten) *non-malleability under chosen-plaintext attacks* (NM-CPA). Es wird gezeigt, dass IND-SDA- und NM-SDA-Sicherheit nicht vergleichbar sind, was bedeutet, dass NM-SDA *strikt* stärker als die beiden ist. Eine black-box Konstruktion von Choi *et al.* [CDMW08], ursprünglich von NM-CPA PKE basierend auf PKE der grundlegendsten Sicherhietsstufe IND-CPA, ist sogar NM-SDA-sicher, wie in dieser Arbeit bewiesen wird. Dadurch ist NM-SDA eine stärkste derjenigen Sicherheitsstufen, über die bekannt ist, dass sie von IND-CPA aus erreicht werden können.

Schliesslich betrachtet diese Arbeit das Domain Extension-Problem auch für NM-SDA-Sicherheit. Dabei stellt sich heraus, dass dieser Ansatz basierend auf gewöhnlichen Non-Malleable Codes nicht funktionieren kann. Deshalb wird ein neues Konzept von Non-Malleable Codes mit *geheimem Zustand* eingeführt und ein Code vorgestellt, der genügend stark für NM-SDA-Domain Extension ist.

Contents

Chapter 1

Introduction

1.1 Public-Key Encryption

Public-Key Encryption (PKE) is a fundamental cryptographic primitive that allows to achieve confidential communication in a context where only non-confidential communication is available [RSA78]. Specifically, a party A, in possession of the *public-key* of some other party B, may encrypt a message in such a way that only B, using the corresponding *secret key*, can decrypt the message. PKE schemes are used in most modern secure communication protocols, where they most often serve the purpose of securely transmitting a session key, i.e., for key agreement [DH76].

Security notions. Following the seminal work by Diffie and Hellman [DH76] and Rivest, Shamir, and Adleman [RSA78], the first formal definition of PKE security was proposed by Goldwasser and Micali [GM84]. Nowadays, there exist a plethora of security definitions for PKE. The two most common such notions are the notions of *indistinguishability under chosen-plaintext attacks* (IND-CPA) and *indistinguishability under chosen-ciphertext attacks* (IND-CCA). Intuitively, IND-CPA security requires that a *computationally bounded* attacker be unable to *distinguish* encryptions of different messages, and IND-CCA extends this guarantee to the case where the attacker is given access to a *decryption oracle*. There exist (practical) scenarios in which IND-CPA security is insufficient and IND-CCA security is needed. However, while IND-CCA is widely regarded as the "right" PKE security notion and is used in most protocols for secure communication, recent work in this area showed that in *interactive* settings in which the recipient B is online, secure communication can be achieved by means of

IND-CPA security only [MTC13, DF14].

Constructions. There exist quite a number of constructions of PKE schemes whose security is based on specific number-theoretic assumptions. For example, ElGamal [ElG84] proposes an IND-CPA-secure PKE scheme based on the so-called *decisional Diffie-Hellman (DDH) assumption*; Cramer and Shoup [CS98, CS01] propose an IND-CCA-secure PKE scheme under the DDH assumption, and Hofheinz and Kiltz [HK09] put forth a scheme with the same security assuming the *hardness of factoring*.

Considerable effort has also been spent on the (im-)possibiliy of generic constructions. For example, in a seminal paper Impagliazzo and Rudich [IR89] show that there exists no *black-box construction* of public-key encryption from *one-way functions*; Cramer *et al.* [CHH+07] build *bounded-query* chosen-ciphertext secure schemes from chosen-plaintext secure ones, Choi *et al.* [CDMW08] *non-malleable* schemes from chosen-plaintext secure ones, and Lin and Tessaro [LT13] show how the security of *weakly* chosen-ciphertext secure schemes can be amplified. A line of work started by Myers, Sergi, and shelat [MSS12] and continued by Dachman-Soled [Dac14] shows how to obtain chosen-ciphertext secure schemes from plaintext-aware ones. Myers and shelat [MS09] and Hohenberger, Lewko, and Waters [HLW12] generically build a *multi-bit* chosen-ciphertext secure scheme from a *single-bit* chosen-ciphertext secure one.

The "holy grail"—generically building an IND-CCA secure scheme from an IND-CPA-secure one—has so far remained out of reach, and the relation between these two notions (despite partial negative results [GMM07]) remains largely unresolved.

1.2 Non-Malleable Codes

Non-malleable codes (NMCs), introduced in a seminal paper by Dziembowski, Pietrzak, and Wichs [DPW10], allow to encode a message in such a way that an attacker cannot transform the encoding into one that decodes to a related message, where "related" means that the outcome of decoding the tampered encoding can be predicted solely from how the adversary tampers with it (but independently of the encoding itself).

Since the introduction of the concept, NMCs resilient against various forms of tampering have been developed. For example, in the original paper [DPW10], the authors provide an NMC resilient against *bit-wise* tampering, where the attacker tampers with each bit of an encoding independently; Aggarwal *et al.* [ADKO15b] develop NMCs for the so-called

split-state model in which an attacker tampers with *two* independent parts of memory.

On an axis orthogonal to the type of tampering, there have also been results on various different attack models. For instance, Faust *et al.* [FMNV14] and later Jafargholi and Wichs [JW15] investigated the notion of *continuous* tampering, in which an attacker gets to tamper with an encoding *repeatedly*.

1.3 Constructive Cryptography

A paradigm common to most constructive disciplines (e.g., software design) is the decomposition of a large, complex system into small, simple component systems, which in turn may again be decomposed into even smaller and simpler systems. This paradigm is only useful if there is a well-defined notion of system composition and if said composition preserves the relevant properties of the components.

The paradigm of *constructive cryptography (CC)* by Maurer [Mau11], based on the framework of abstract cryptography by Maurer and Renner [MR11], was proposed with the objective of putting the design of cryptographic protocols on a constructive foundation.

In CC, resources, such as various types of communication channels, shared keys, memories, etc., are modeled explicitly, and the goal of a protocol π is to *construct*—in a well-defined sense—a resource S from an assumed resource R, which is denoted by

$$R \overset{\pi}{\Longrightarrow} S.$$

The *assumed* resource R (explicitly) captures the "infrastructure" available to the parties participating in the execution of π, while the *constructed* resource S (explicitly) specifies the functionality π is expected to implement using R.

The key feature of CC is that such construction steps compose: If, in addition to the above, protocol ψ constructs resource T from S, then the composed protocol, denoted by $\psi \circ \pi$, constructs T from R, i.e.,

$$R \overset{\pi}{\Longrightarrow} S \quad \wedge \quad S \overset{\pi}{\Longrightarrow} T \qquad \Longrightarrow \qquad R \overset{\psi \circ \pi}{\Longrightarrow} T.$$

This kind of composition precisely enables the *step-wise refinement* approach outlined above, in which a protocol is built in a modular fashion from isolated construction steps, each construction step is analyzed in isolation, and the composition property guarantees the security of the final protocol.

1.4 Contributions of This Thesis

The contributions of this thesis comprise the works [CMT13], [CMTV15], and [CDTV16].

1.4.1 A Constructive Perspective on PKE

The cryptographic security of PKE is traditionally defined in terms of certain *games* in which no efficient adversary is supposed to achieve a non-negligible advantage. There exists quite a wide spectrum of security notions and variants thereof. These notions are motivated by certain attacks (e.g., a chosen-ciphertext attack) that should be prevented, and, in some cases, new notions are also proposed because they are stronger than previous notions or can be shown to be incomparable.

Game-based security notions only implicitly encode

- the assumptions about the setting in which a PKE scheme is deployed, via oracles available to the attacker, and
- the guaranteed security properties, via the winning condition in the game.

Consequently, game-based notions do not *compose*. That is, when faced with the question of which PKE security notion is suitable or necessary for a certain higher-level protocol (using PKE) to be secure, one first needs to identify an appropriate security notion and then provide a reduction proof to show that a PKE satisfying that notion yields a secure protocol.

An alternative approach, based on the CC paradigm (cf. Section 1.3), to capturing the security of a PKE scheme is to consider a protocol π based on PKE and a *construction*

$$R \stackrel{\pi}{\Longrightarrow} S,$$

making explicit—via the definition of the assumed resource R—in which contexts π can be used securely and—via the definition of the constructed resource S—the security guarantees of π.

In this spirit, in the first part of this thesis (Chapter 3), the use of PKE is treated as such a construction step. First, it is shown how one can construct, using PKE, confidential channels from authenticated and insecure channels. Second, several known game-based security notions (and variants thereof) are revisited and given a constructive semantics, providing an explicit understanding of the application contexts for which a given notion is suitable.

Constructing confidential channels using PKE. From the perspective of CC, the purpose of a public-key encryption scheme is to construct a confidential channel from non-confidential channels. Here, a channel is a *resource* that involves a sender, a receiver, and—to model channels with different levels of security—an attacker. A channel generally allows the sender to transmit a message to the receiver; the security properties of a particular channel are captured by the capabilities available to the attacker, which might, e.g., include reading or modifying the messages in transmission.

The parties access the channel through interfaces that the channel provides and that are specific for each party. For example, the sender's interface allows to input messages, and the receiver's interface allows to receive them. The interfaces are labeled by A, B, and E, where A and B are the sender's and the receiver's interfaces, respectively, and E is the adversary's interface. This thesis considers the following four basic types of channels (from A to B; channels in the opposite direction are defined analogously):

- An *insecure channel*, denoted INSEC_{AB}, allows the adversary to read, deliver, and to delete all messages input at A, as well as to inject its own messages.
- An *authenticated channel*, denoted AUTH_{AB}, still allows to read all messages, but the adversary is limited to forwarding or deleting messages input at interface A.
- A *confidential channel*, denoted CONF_{AB}, only leaks the length of the messages sent by A but does not necessarily prevent injections.
- A *secure channel*, denoted SEC_{AB}, also only leaks the message length and only allows the adversary to forward or delete messages input at A.

To use public-key encryption, the receiver initially generates a key pair and transmits the public key to the sender. The sender needs to obtain the correct public key, which corresponds to assuming that the channel from B to A is authenticated ($\mathsf{1\text{-}AUTH}_{BA}$).[1] To transmit a message confidentially, the sender then encrypts the message under the received public key and sends the ciphertext to the receiver over a channel that could be authenticated or completely insecure.

[1] The "1" denotes that $\mathsf{1\text{-}AUTH}_{BA}$ is a single-use channel, i.e., only one message can be transmitted.

The exact type of channel that is constructed depends on the type of assumed channel used to transmit the ciphertext to the receiver: If the assumed channel is authenticated (AUTH_{AB}) and the PKE scheme is IND-CPA-secure, the constructed channel is a secure channel (SEC_{AB}). If, however, the assumed channel is insecure (INSEC_{AB}) and the PKE scheme is IND-CCA-secure, the constructed channel is a confidential channel (CONF_{AB}). Using the above notation, for protocols π and π' based on IND-CPA and IND-CCA encryption schemes, respectively, these constructions can be written as

$$[\text{1-}\mathsf{AUTH}_{BA}, \mathsf{AUTH}_{AB}] \quad \overset{\pi}{\Longrightarrow} \quad \mathsf{SEC}_{AB}$$

and

$$[\text{1-}\mathsf{AUTH}_{BA}, \mathsf{INSEC}_{AB}] \quad \overset{\pi'}{\Longrightarrow} \quad \mathsf{CONF}_{AB},$$

where, informally, the bracket notation means that both resources in the brackets are available in parallel.

The notion of constructing the confidential (or secure) channel from the two assumed non-confidential ones is made precise in a simulation-based sense [MR11, Mau11], where the simulator can be interpreted as translating all attacks on the protocol into attacks on the constructed (ideal) channel. As the constructed channel is secure by definition, there are no attacks on the protocol.

The composability of the construction notion then means that the constructed channel can again be used as an assumed resource (possibly along with additional assumed or constructed resources) in other protocols. For instance, if a higher-level protocol uses the confidential channel to transmit a message together with a shared secret value in order to achieve an additionally authenticated (and hence fully secure) transmission of the message, then the proof of this protocol is based on the "idealized" confidential channel and does not (need to) include a reduction to the security of the PKE scheme. In the same spirit, the authenticated channel from B to A could be a physically authenticated channel, but it could also be constructed by using, for instance, a digital signature scheme to authenticate the transmission of the public key (which is done by certificates in practice).

Constructive semantics of game-based security notions. Security properties for PKE are often formalized via a game between a hypothetical challenger and an attacker. This thesis assigns constructive semantics to several existing game-based definitions by first characterizing the appropriate assumed and constructed resources and then showing that the

"standard use" of a PKE scheme over those channels (as illustrated above) achieves the construction if (and sometimes only if) it has the considered property.[2]

In particular, it is shown that IND-CPA-security is not only sufficient but also necessary for constructing a secure channel from two authenticated channels. For the construction of a confidential channel from an authenticated and an insecure channel, it turns out that IND-CCA-security, while sufficient, is unnecessarily strong. The transformation only requires the weaker notion of IND-RCCA-security, which was introduced by Canetti *et al.* [CKN03] to avoid the artificial strictness of IND-CCA.

Bellare *et al.* [BHK09] considered several non-equivalent definitional variants of IND-CCA. It is shown that only the stricter notions they consider are sufficient for the channel construction, leaving the exact semantics of the weaker notions unclear.

Non-adaptive chosen-ciphertext security (IND-CCA1) is also considered: the notion corresponds to a transformation between somewhat artificial channels but might still be useful for specific applications.

Finally, an approach to generalizing security in the three-party setting to a setting with multiple senders and receivers is discussed.

1.4.2 On the Gap between IND-CPA and IND-CCA Security

In some contexts, the most basic PKE security notion of IND-CPA is not sufficient, e.g., when the channel from the sender to the receiver is not authenticated, as illustrated in Section 1.4.1. Another example where mere IND-CPA-secure PKE is not adequate is the simple setting of an electronic auction, where the auctioneer U publishes a public key pk and invites several participants $P_1, P_2, \ldots$ to encrypt their bids b_i under pk. As was observed in the seminal paper of Dolev *et al.* [DDN00], although IND-CPA security ensures that P_1 cannot decrypt a bid of P_2 under the ciphertext e_2, it leaves open the possibility that P_1 can create a special ciphertext e_1 that decrypts to a *related* bid b_1 (e.g., $b_1 = b_2 + 1$). Hence, to overcome such "malleability" problems, stronger forms of security are required.

The strongest such security notion is IND-CCA. However, the fact that it is not known whether IND-CCA-secure PKE schemes can be generically built from IND-CPA-secure ones (cf. Section 1.1) motivates the study of various "middle-ground" security notions between IND-CPA and IND-CCA;

[2] Note that the negative results do *not* rule out the existence of other protocols that are derived from the scheme in some possibly more complicated way; those could still achieve the respective construction.

notions that are sufficient for applications, and, yet, might be constructed from simpler basic primitives (e.g., IND-CPA encryption).

One influential such notion is *non-malleability under chosen-plaintext attacks* (NM-CPA), originally introduced by Dolev *et al.* [DDN00] with the goal of precisely addressing the auction example above, by demanding that an adversary not be able to maul ciphertexts to other ciphertexts encrypting related plaintexts. As was later shown by Bellare and Sahai [BS99] and by Pass *et al.* [PSV07], NM-CPA is equivalent to security against adversaries with access to a *non-adaptive* decryption oracle, meaning that the adversary can only ask one "parallel" decryption query. Although NM-CPA appears much closer to IND-CCA than IND-CPA security, a seminal result by Pass *et al.* [PSV06] showed that one can generically build NM-CPA encryption from any IND-CPA-secure scheme, and Choi *et al.* [CDMW08] later proved that this transformation can also be achieved via a black-box construction. Thus, NM-CPA schemes can be potentially based on weaker assumptions than IND-CCA schemes, and yet suffice for important applications.

Another middle-ground security notion for PKE—termed *indistinguishability under (chosen-ciphertext) self-destruct attacks* (IND-SDA)—is introduced by this thesis. With IND-SDA the adversary gets access to an *adaptive* decryption oracle, which, however, stops decrypting after the first *invalid* ciphertext is submitted. Applying this notion to the auction example above, it means that the auctioneer can reuse the secret key for subsequent auctions, as long as all the encrypted bids are valid. Unfortunately, if an invalid ciphertext is submitted, even the results of the *current* auction should be discarded, as IND-SDA security is not powerful enough to argue that the decryptions of the remaining ciphertexts are unrelated w.r.t. prior plaintexts.

The second part of this thesis (Chapter 4) first formally defines the new notion of IND-SDA security. Then, it shows that IND-SDA and NM-CPA are incomparable, i.e., there exist (albeit contrived) PKE schemes that satisfy the former notion but not the latter and vice-versa.

Motivated by the above, a new security notion dubbed *non-malleability under (chosen-ciphertext) self-destruct attacks* (NM-SDA) is introduced. This notion naturally combines NM-CPA and IND-SDA, by allowing the adversary to ask many adaptive "parallel" decryption queries (i.e., a query consists of many ciphertexts) up to the point when the first invalid ciphertext is submitted. In such a case, the whole parallel decryption query containing the invalid ciphertext is still answered in full, but no further decryption queries are allowed. Since NM-SDA security implies both of the

incomparable NM-CPA and IND-SDA notions, it is strictly stronger than them. Hence, NM-SDA security appears to be a strongest natural PKE security notion that is (probably) still weaker than IND-CCA—together with q-bounded CCA-secure PKE [CHH+07], to which it seems incomparable. In particular, it seems to apply better to the auction example above: First, unlike with basic NM-CPA, the auctioneer can reuse the same public key pk, provided no invalid ciphertexts were submitted. Second, unlike IND-SDA, the current auction can be safely completed, even if some ciphertexts are invalid. Compared to IND-CCA, however, the auctioneer will still have to change the public key for *subsequent* auctions if some of the ciphertexts are invalid. Still, one can envision situations in which parties are penalized for submitting such malformed ciphertexts, in which case NM-SDA security might be practically sufficient, leading to an implementation under (potentially) weaker computational assumptions as compared to using a full-blown IND-CCA PKE.

Additionally, the auction example is formalized in the CC framework, and NM-SDA-secure PKE is shown to be sufficient to realize it. Similar constructive semantics are given to the notions of IND-SDA and NM-CPA.

Finally, a generalization of the Choi *et al.* [CDMW08] construction from IND-CPA encryption is presented. The new construction has an improved plaintext-length to ciphertext-length rate (by a factor linear in the security parameter) and is shown to achieve NM-SDA security.[3]

1.4.3 Domain Extension for PKE

For several security notions in public-key cryptography, is is known that single-bit public-key encryption implies multi-bit public-key encryption. For IND-CPA-secure PKE, this question is simple [GM84], since the parallel repetition of a single-bit scheme (i.e., encrypting every bit of a message separately) yields an IND-CPA-secure multi-bit scheme. For the other notions considered in this thesis, i.e., for NM-CPA, IND-SDA, and NM-SDA, as well as for IND-CCA, the parallel repetition (even using independent public keys) is not a scheme that achieves the same security level as the underlying single-bit scheme.

To illustrate this, consider the naïve, parallel-repetition method in more detail: each bit $m[i]$ of a plaintext $m = m[1] \cdots m[k]$ is encrypted under an independent public key pk_i of the single-bit scheme. The resulting scheme is, however, *malleable*: given a ciphertext $e = (e_1, \ldots, e_k)$,

[3] Note that the original scheme by Choi *et al.* [CDMW08] without the rate improvement also achieves NM-SDA security.

where e_i is an encryption of $m[i]$, an attacker can generate a new ciphertext $e' \neq e$ that decrypts to a related message, for instance by copying the first ciphertext component e_1 and replacing the other components by fresh encryptions of, say, 0.

The above malleability issue suggests the following natural *encode-then-encrypt-bit-by-bit (EtEb)* approach: first encode the message using a non-malleable code to protect its integrity, obtaining an n-bit codeword $c = c[1] \cdots c[n]$; then encrypt each bit $c[i]$ of the codeword using public key pk_i as in the naïve scheme from above.

Given that each bit $c[i]$ of the encoding is encrypted under a separate public key, the non-malleable code used in the transformation must be resilient against *bit-wise* tampering. This is seen particularly easily when considering the fact that a single-bit PKE scheme with IND-SDA security allows to construct a single-bit confidential channel $\mathsf{CONF}_{AB}^{\text{1-bit}}$, similarly to IND-CCA security (cf. Section 1.4.1). Since n such PKE schemes in parallel construct n such channels in parallel ($[\mathsf{CONF}_{AB}^{\text{1-bit}}]^n$), the NMC must achieve the transformation

$$[\mathsf{CONF}_{AB}^{\text{1-bit}}]^n \Longmapsto \mathsf{CONF}_{AB}^{k\text{-bit}}.$$

Since the single-bit channels are independent, if they are used to transmit an encoding from A to B, the attacker can tamper with each bit of the encoding separately. (Similar arguments can be made for NM-CPA and NM-SDA security.)

It turns out that plain non-malleable codes as introduced by [DPW10] are not sufficient to obtain PKE domain extension for IND-SDA security: Since such NMCs are only secure against a single tampering, the security of the resulting scheme would only hold with respect to a single decryption. The third part of this thesis (Chapter 5) shows that *continuously* non-malleable codes (Faust *et al.* [FMNV14]) allow to extend the NMC guarantees to multiple decryptions. However, such codes "self-destruct" once an attack has been detected, and, therefore, so must any PKE scheme built on top of them. This is a restriction that is proved to be unavoidable for the EtEb approach to work.[4]

Even continuous non-malleable codes are, however, not strong enough to obtain PKE domain extension for NM-CPA and NM-SDA via the EtEb approach: since these notions allow the attacker to make parallel decryption queries, the underlying code must be non-malleable w.r.t. parallel

[4] In fact, the self-destruct notions were originally discovered in the context of analyzing the EtEb approach for IND-CCA encryption.

tampering attacks. Since one can show that normal NMCs cannot be secure against parallel tampering, this thesis introduces the new notion of *secret-state* NMCs, in which the decoder has access to a secret value it generates initially. When such NMCs are combined with PKE, this secret value simply becomes part of the secret key.

1.4.4 Non-Malleability against Bit-Wise Tampering

The final part of this thesis (Chapter 6) provides an NMC resilient against *continuous* and one against *continuous parallel* tampering. For the former case, a construction by Dziembowski *et al.* [DPW10] secure against non-continuous tampering is shown to withstand continuous tampering attacks as well. For the latter case, a secret-state non-malleable code based on so-called *linear error-correcting secret sharing* and the idea of a secret "trigger" set (inspired by the [CDMW08] construction) is presented.

1.5 Related Work

Real-World/Ideal-World Security. The idea of defining protocol security with respect to an ideal execution was first proposed by Goldreich *et al.* [GMW87], where a simulator was used to formalize that whatever the adversary can achieve in an attack on the protocol he can also achieve in the ideal execution. First formal treatments of this approach were by Goldwasser and Levin [GL90], Micali and Rogaway [MR91], and Beaver [Bea91] in the context of multi-party computation. The concept of a simulator can be traced back to the seminal work by Goldwasser *et al.* [GMR85], who introduced it in the context of zero-knowledge proofs.

General security frameworks that allow the formalization of arbitrary functionalities to be realized by cryptographic protocols have been introduced by Canetti [Can00] as Universal Composability (UC) as well as by Pfitzmann and Waidner [PW01] and Backes *et al.* [BPW07] as Reactive Simulatability (RSIM). Treatments of PKE exist in both frameworks. As explained in more detail in Section 3.4, the treatment in UC is with respect to an "ideal PKE" functionality. Realizing this functionality is equivalent to IND-CCA-security [CKN03].

Canetti and Krawczyk [CK02] formulate UC functionalities that model different types of communication channels and can be interpreted as network resources; they show that their secure channels functionality can be realized by key exchange and symmetric encryption. They do not treat public-key encryption (beyond what is implied by viewing the above scheme as KEM-DEM).

The formalization of the functionalities in [PW01] is closer to the approach used here, but less modular and hence more complicated since they immediately analyze the schemes in a multi-party scenario; the treatment is restricted to and directly proves the case where the authenticated transmission of the ciphertexts is achieved by digital signatures instead of using a generic composition statement. More generally, both frameworks [Can00] and [PW01] are designed from a bottom-up perspective (starting from a selected machine model), whereas this work follows the top-down approach of [MR11], which leads to simpler, more abstract definitions and statements.

Maurer *et al.* [MRT12] described *symmetric* encryption as the construction of confidential channels from non-confidential channels and shared keys, and compared the security definitions they obtained to previous game-based definitions. The goal of this work is to provide a comparable treatment for the case of public-key encryption. In the same spirit, specific anonymity-related properties of public-key encryption and their relation to the construction of receiver-anonymous channels have been discussed by Kohlweiss *et al.* [KMO+13].

Domain extension for PKE. The problem of domain extension for fully IND-CCA-secure PKE was considered by Myers and shelat [MS09] and Hohenberger *et al.* [HLW12]. A detailed comparison between the EtEb approach and these works is given in Section 5.4.1.

Non-malleable codes. Optimal-rate (non-continuous) NMCs against bit-wise tampering are provided by [CG14b]. NMCs also exist against block-wise tampering [CKM11], against bit-wise tampering plus permutations [AGM+15a, AGM+15b], against split-state tampering—both information-theoretic [DKO13, ADL14, CZ14, ADKO15b, ADKO15a] and computational [LL12, DLSZ15]—and in a setting where the computational complexity of the tampering functions is limited [CG14a, FMVW14, JW15].

The typical application of non-malleable codes is to protect cryptographic schemes against memory tampering (see, e.g., [GLM+04, DPW10, DFMV13, DFMV15]). A further application of non-malleable codes has been shown by Agrawal *et al.* [AGM+15a]. They show that one can obtain a non-malleable multi-bit commitment scheme from a non-malleable single-bit commitment scheme by encoding the value with a (specific) non-malleable code and then committing to the codeword bits. Despite the similarity of the approaches, the techniques applied in their paper differ heavily from those used in this theses.

Chapter 2

Preliminaries

2.1 Constructive Cryptography

The constructive cryptography paradigm (cf. Section 1.3) and its construction notion are explained in Section 2.5, after introducing the necessary formalism in Sections 2.2 and 2.3.

2.2 Systems

This thesis uses so-called *systems* to capture resources, protocols, and security games. At the highest level of abstraction (following the hierarchy in [MR11]), systems are objects with interfaces by which they connect to (interfaces of) other systems. This concept of *abstract systems* captures the topological structures that result when multiple systems are connected in this manner.

The abstract systems concept, however, does not model the behavior of systems, i.e., *how* the systems interact via their interfaces. Consequently, statements about cryptographic protocols are statements at the next (lower) abstraction level. In this work, all systems are described in terms of (probabilistic) discrete systems, which are explained in Section 2.3.

Resources and Converters. Resources and converters are the main objects of interest in CC. All resources in this work are systems with three interfaces, labeled by A, B, and E. Converters are two-interface

systems, which are directed in that they have an *inside* and an *outside* interface, denoted by `in` and `out`, respectively.

As a notational convention, upper-case, bold-face letters (e.g., $\mathbf{R}$, $\mathbf{S}$) and upper-case sans-serif fonts (e.g., $\mathsf{INSEC}^{\mathcal{M}}_{AB}$) generally denote resources and lower-case Greek letters (e.g., α, β) or lower-case sans-serif fonts (e.g., enc, dec) denote converters. The set of all resources is denoted by Φ and the set of all converters by Σ.

Converters model protocol engines that are used by the parties, and using a protocol is modeled by connecting the party's interface of the resource to the inside interface of the converter (which hides those two interfaces) and using the outside interface of the converter instead.

For $I \in \{A, B, E\}$, a resource $\mathbf{R} \in \Phi$, and a converter $\alpha \in \Sigma$, the expression $\alpha^I\mathbf{R}$ denotes the composite system obtained by connecting the inside interface of α to interface I of $\mathbf{R}$; the outside interface of α becomes the I-interface of the composite system. The system $\alpha^I\mathbf{R}$ is again a resource.

For two resources $\mathbf{R}$ and $\mathbf{S}$, $[\mathbf{R}, \mathbf{S}]$ denotes the parallel composition of $\mathbf{R}$ and $\mathbf{S}$. For each $I \in \{A, B, E\}$, the I-interfaces of $\mathbf{R}$ and $\mathbf{S}$ are merged and become the *sub-interfaces* $I.1$ and $I.2$ of the I-interface of $[\mathbf{R}, \mathbf{S}]$.

Two converters α and β can be composed serially by connecting the inside interface of β to the outside interface of α, written $\beta \circ \alpha$, with the effect that $(\beta \circ \alpha)^I\mathbf{R} = \beta^I\alpha^I\mathbf{R}$. Moreover, converters can also be taken in parallel, denoted by $[\alpha, \beta]$, with the effect that $[\alpha, \beta]^I[\mathbf{R}, \mathbf{S}] = [\alpha^I\mathbf{R}, \beta^I\mathbf{S}]$; the inner and outer interfaces of $[\alpha, \beta]$ are denoted by `in`.1, `in`.2 and `out`.1, `out`.2, respectively. This work assumes the existence of an identity converter $\mathsf{id} \in \Sigma$ with $\mathsf{id}^I\mathbf{R} = \mathbf{R}$ for all resources $\mathbf{R} \in \Phi$ and interfaces $I \in \{A, B, E\}$.

Distinguishers. A *distinguisher* $\mathbf{D}$ connects to all interfaces of a resource $\mathbf{U}$ and outputs a single bit at the end of its interaction with $\mathbf{U}$. The expression $\mathbf{DU}$ defines a binary random variable corresponding to the output of $\mathbf{D}$ when interacting with $\mathbf{U}$, and the *distinguishing advantage of a distinguisher* $\mathbf{D}$ *on two systems* $\mathbf{U}$ *and* $\mathbf{V}$ is defined as

$$\Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V}) := |\mathsf{P}[\mathbf{DU} = 1] - \mathsf{P}[\mathbf{DV} = 1]|.$$

The distinguishing advantage measures how much the output distribution of $\mathbf{D}$ differs when it is connected to $\mathbf{U}$ as opposed to $\mathbf{V}$. Note that the distinguishing advantage satisfies the triangle inequality, i.e.,

$$\Delta^{\mathbf{D}}(\mathbf{S}, \mathbf{T}) \leq \Delta^{\mathbf{D}}(\mathbf{S}, \mathbf{R}) + \Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{T})$$

for all distinguishers $\mathbf{D}$, and systems $\mathbf{S}$, $\mathbf{R}$, and $\mathbf{T}$.

Reductions. When relating two distinguishing problems, it is convenient to use a special type of system $\mathbf{C}$ that translates one setting into the other. Formally, $\mathbf{C}$ is a converter that has an *inside* and an *outside* interface. When it is connected to a system $\mathbf{S}$, which is denoted by $\mathbf{CS}$, the inside interface of $\mathbf{C}$ connects to the (merged) interface(s) of $\mathbf{S}$ and the outside interface of $\mathbf{C}$ is the interface of the composed system. $\mathbf{C}$ is called a *reduction system* (or simply *reduction*).

To reduce distinguishing two systems $\mathbf{S}, \mathbf{T}$ to distinguishing two systems $\mathbf{U}, \mathbf{V}$, one exhibits a reduction $\mathbf{C}$ such that $\mathbf{CS} \equiv \mathbf{U}$ and $\mathbf{CT} \equiv \mathbf{V}$. Then, for all distinguishers $\mathbf{D}$,

$$\Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V}) = \Delta^{\mathbf{D}}(\mathbf{CS}, \mathbf{CT}) = \Delta^{\mathbf{DC}}(\mathbf{S}, \mathbf{T}),$$

where the last equality follows from the fact that $\mathbf{C}$ can also be thought of as being part of the distinguisher.[1]

2.3 Discrete Systems

The behavior of systems can be formalized by random systems as in [Mau02, Mau13]: A random system $\mathbf{S}$ is a sequence $(\mathsf{p}^{\mathbf{S}}_{Y^i|X^i})_{i\geq 1}$ of conditional probability distributions, where $\mathsf{p}^{\mathbf{S}}_{Y^i|X^i}(y^i, x^i)$ is the probability of observing the outputs $y^i = (y_1, \ldots, y_i)$ given the inputs $x^i = (x_1, \ldots, x_i)$. Systems with multiple interfaces are modeled similarly; the interface to which an input or output is associated is explicitly specified as part of the input or output. For the restricted (but for this work sufficient) class of systems that for each input provide (at most) a single output, an execution of a collection of systems is defined as the consecutive evaluation of the respective random systems (similarly to the model in [Can00]).

If for two systems $\mathbf{R}$ and $\mathbf{S}$,

$$\mathsf{p}^{\mathbf{R}}_{Y^i|X^i} = \mathsf{p}^{\mathbf{S}}_{Y^i|X^i}$$

for all i and for all parameters where both are defined, they are called *equivalent*, denoted by $\mathbf{R} \equiv \mathbf{S}$. In that case, $\Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{S}) = 0$ for all distinguishers $\mathbf{D}$.

A system $\mathbf{S}$ can be extended by a so-called *monotone binary output* (or *MBO*) $\mathcal{B}$, which is an additional one-bit output $B_1, B_2, \ldots$ with the property that $B_i = 1$ implies $B_{i+1} = 1$ for all i. [2] The enhanced system is

[1] This follows from the so-called *composition-order independence*. See [MR11] for more details.

[2] In other words,once the MBO is 1, it cannot return to 0.

denoted by $\hat{\mathbf{S}}$, and its behavior is described by the sequence $(\mathsf{p}^{\hat{\mathbf{S}}}_{Y^i,B_i|X^i})_{i\geq 1}$ (since one is only interested in the behavior as long as the MBO is zero).

If for two systems $\hat{\mathbf{R}}$ and $\hat{\mathbf{S}}$ with MBOs,

$$\mathsf{p}^{\hat{\mathbf{R}}}_{Y^i,B_i=0|X^i} = \mathsf{p}^{\hat{\mathbf{S}}}_{Y^i,B_i=0|X^i}$$

for all i, they are called *game equivalent*, which is denoted by $\hat{\mathbf{R}} \stackrel{g}{\equiv} \hat{\mathbf{S}}$. In such a case, $\Delta^{\mathbf{D}}(\mathbf{R},\mathbf{S}) \leq \Gamma^{\mathbf{D}}(\hat{\mathbf{R}}) = \Gamma^{\mathbf{D}}(\hat{\mathbf{S}})$, where $\Gamma^{\mathbf{D}}(\hat{\mathbf{R}})$ denotes the probability that $\mathbf{D}$ provokes the MBO.[3] For more details and a proof of this fact, consult [Mau13].

2.4 Asymptotics

All statements in this paper are phrased in a non-asymptotic fashion, but asymptotic statements can be obtained by treating systems $\mathbf{S}$ as asymptotic families $\{\mathbf{S}_\lambda\}_{\lambda\in\mathbb{N}}$ and letting the distinguishing advantage be a real-valued function of λ. Then, for a given notion of efficiency, one can consider security w.r.t. classes of efficient distinguishers and a suitable negligibility notion. All reductions in this work are efficient with respect to the standard polynomial-time notions.

2.5 The Notion of Construction

Following the CC paradigm, the task of a cryptographic protocol π is to transform a particular *assumed* resource $\mathbf{R}$ into a (stronger) *constructed* resource $\mathbf{S}$. This is formalized via the well-known *real-world/ideal-world paradigm*, which compares two different settings: the actual execution of the protocol π using the assumed resource $\mathbf{R}$ and an ideal setting, in which each adversarial party accesses $\mathbf{S}$ via a *simulator*. The protocol is secure if there exist suitable simulators such that the real and the ideal settings are indistinguishable to a distinguisher with access to all interfaces. Hence, the simulator captures the fact that everything a party attacking π in the real world can achieve it could also achieve in the ideal setting with $\mathbf{S}$.

In the three-party setting considered in this thesis, two of the parties, Alice and Bob, are considered honest, i.e., they follow their protocol, while the third party, Eve, is adversarial and may behave arbitrarily. Consequently, in this setting

[3] Intuitively, this means that in order to distinguish the two systems, $\mathbf{D}$ has to provoke the MBO.

- resources $\mathbf{S}$ have an *honest* mode, denoted by $\mathbf{S}$-h, that captures the behavior of the resource when Eve does not interfere,
- protocols $\pi = (\pi_1, \pi_2)$ consist of two converters, one for Alice and one for Bob, and
- to prove a particular protocol secure, one needs to
  - show that $\pi_1^A \pi_2^B \mathbf{R}$-$\mathsf{h}$ is indistinguishable from $\mathbf{S}$-h and
  - to exhibit a simulator σ for E such that $\pi_1^A \pi_2^B \mathbf{R}$ is indistinguishable from $\sigma^E \mathbf{S}$.

The above leads to the following definition, which originally appeared in [Mau11]:

Definition 2.1. *Let Φ and Σ be as defined in Section 2.2, and let ε_1 and ε_2 be two functions mapping each distinguisher $\mathbf{D}$ to a real number in $[0, 1]$. A protocol $\pi = (\pi_1, \pi_2) \in \Sigma^2$* constructs resource $\mathbf{S} \in \Phi$ from resource $\mathbf{R} \in \Phi$ with distance $(\varepsilon_1, \varepsilon_2)$ and with respect to the simulator $\sigma \in \Sigma$, *denoted*[4]

$$\mathbf{R} \overset{\pi,\sigma,(\varepsilon_1,\varepsilon_2)}{\Longrightarrow} \mathbf{S},$$

if for all distinguishers $\mathbf{D}$,

$$\begin{cases} \Delta^{\mathbf{D}}({\pi_1}^A {\pi_2}^B \mathbf{R}\text{-}\mathsf{h}, \mathbf{S}\text{-}\mathsf{h}) \leq \varepsilon_1(\mathbf{D}) & \textit{(availability)} \\ \Delta^{\mathbf{D}}({\pi_1}^A {\pi_2}^B \mathbf{R}, \sigma^E \mathbf{S}) \leq \varepsilon_2(\mathbf{D}) & \textit{(security).} \end{cases}$$

The availability condition captures that a protocol must correctly implement the functionality of the constructed resource in the absence of the attacker. The security condition models the requirement that everything the attacker can achieve in the setting with the assumed resource and the protocol, he can also accomplish in the setting with the constructed resource (using the simulator to translate the behavior).

2.6 The Composition Theorem

The construction notion defined in the previous section considers the security of a protocol in isolation. The notion, however, composes: if a (lower-level) protocol constructs the resource that is assumed by another

[4] In less formal contexts, some of the superscripts on $\Longrightarrow$ are occasionally dropped.

(higher-level) protocol, then the composition of those two protocols constructs the same resource as the higher-level protocol, but from the resources assumed by the lower-level protocol, under the assumptions that occur in (at least) one of the individual security statements.

The composition theorem below was first explicitly stated in [MT10], but the statement there was restricted to asymptotic settings. Later, in [KMO+13], the theorem was stated in a way that also allows to capture concrete security statements. The proof, however, still follows the same steps as the one in [MT10].[5]

Theorem 2.1. *Let* $\mathbf{R}, \mathbf{S}, \mathbf{T}, \mathbf{U} \in \Phi$ *be resources. Let* $\pi = (\pi_1, \pi_2)$ *and* $\psi = (\psi_1, \psi_2)$ *be protocols,* σ_π *and* σ_ψ *be simulators, and* $(\varepsilon_\pi^1, \varepsilon_\pi^2)$, $(\varepsilon_\psi^1, \varepsilon_\psi^2)$ *be such that*

$$\mathbf{R} \overset{(\pi, \sigma_\pi, (\varepsilon_\pi^1, \varepsilon_\pi^2))}{\Longrightarrow} \mathbf{S} \quad \text{and} \quad \mathbf{S} \overset{(\psi, \sigma_\psi, (\varepsilon_\psi^1, \varepsilon_\psi^2))}{\Longrightarrow} \mathbf{T}.$$

Then,

$$\mathbf{R} \overset{(\alpha, \sigma_\alpha, (\varepsilon_\alpha^1, \varepsilon_\alpha^2))}{\Longrightarrow} \mathbf{T}$$

with $\alpha := (\psi_1 \circ \pi_1, \psi_2 \circ \pi_2)$, $\sigma_\alpha := \sigma_\pi \circ \sigma_\psi$, *and* $\varepsilon_\alpha^i(\mathbf{D}) := \varepsilon_\pi^i(\mathbf{D}\sigma_\psi^E) + \varepsilon_\psi^i(\mathbf{D}\pi_1^A\pi_2^B)$, *where* $\mathbf{D}\sigma_\psi^E$ *and* $\mathbf{D}\pi_1^A\pi_2^B$ *mean that* $\mathbf{D}$ *applies the converters at the respective interfaces. Moreover,*

$$[\mathbf{R}, \mathbf{U}] \overset{([\pi, (\mathsf{id}, \mathsf{id})], [\sigma_\pi, \mathsf{id}], (\bar{\varepsilon}_\pi^1, \bar{\varepsilon}_\pi^2))}{\Longrightarrow} [\mathbf{S}, \mathbf{U}],$$

with $\bar{\varepsilon}_\pi^i(\mathbf{D}) := \varepsilon_\pi^i(\mathbf{D}[\cdot, \mathbf{U}])$, *where* $\mathbf{D}[\cdot, \mathbf{U}]$ *means that the distinguisher emulates* $\mathbf{U}$ *in parallel. (The analogous statement holds with respect to* $[\mathbf{U}, \mathbf{R}]$ *and* $[\mathbf{U}, \mathbf{S}]$*.)*

2.7 Channels

A channel is a resource that involves a sender A, a receiver B, and—to model channels with different levels of security—an attacker E.[6] The channel types relevant for this thesis are defined below. All channels are parametrized by a message space $\mathcal{M} \subseteq \{0,1\}^*$, which is often dropped

[5] Recall from Section 2.2 that id is the converter that behaves transparently (i.e., allows access to the underlying interface of the resource). Furthermore, it is assumed that the operation $[\cdot, \ldots, \cdot]$ is left-associative; in this way multiple resources can be expressed using a single variable $\mathbf{U}$.

[6] Channels in the opposite direction are defined similarly.

when it is not of importance. All channel resources **S**, whenever no attacker is present (i.e., with **S**-h) relay messages from A to B faithfully. That is, when a message $m \in \mathcal{M}$ in input at A, it is output at B.

The following sections describe the behavior of the channels when an attacker is present at interface E.

2.7.1 Insecure Channel

The insecure channel $\mathsf{INSEC}^{\mathcal{M}}_{AB}$ transmits messages $m \in \mathcal{M}$ and corresponds to, for instance, communication via the Internet. Communication can be controlled via the E-interface, i.e., the attacker learns all messages input at the A-interface and chooses the messages to be output at the B-interface. The channel is described in more detail in Figure 2.1.

Channel $\mathsf{INSEC}^{\mathcal{M}}_{AB}$

on $m \in \mathcal{M}$ *at* A
  output (msg, m) at E

on (inj, m') *with* $m' \in \mathcal{M}$ *at* E
  output m' at B

Figure 2.1: *Insecure communication channel from A to B.*

2.7.2 Authenticated Channel

The authenticated channel $\mathsf{AUTH}^{\mathcal{M}}_{AB}$ authentically transmits messages $m \in \mathcal{M}$. Communication can be controlled via the E-interface with the restriction that only messages input at A may be output at B. That is, the attacker learns all messages input at the A-interface and chooses the messages to be output at the B-interface from a buffer $\mathcal{B}$ containing all messages input at A. The channel is described in more detail in Figure 2.2.

Channel $\mathsf{AUTH}^{\mathcal{M}}_{AB}$

init
  $i \leftarrow 0$
  $\mathcal{B} \leftarrow \emptyset$

on $m \in \mathcal{M}$ *at* A
  $i \leftarrow i + 1$
  $\mathcal{B} \leftarrow \mathcal{B} \cup \{m\}$
  output (msg, i, m) at E

on (dlv, i') *at* E
  if $\exists m : (i', m) \in \mathcal{B}$
    output m at B

Figure 2.2: *Authenticated communication channel from A to B.*

2.7.3 Confidential Channel

The confidential channel $\mathsf{CONF}_{AB}^{\mathcal{M}}$ works as follows: When a message is input at A, it is stored in a buffer $\mathcal{B}$, and its length is output at E. The attacker can (repeatedly) either choose a message from the buffer to be delivered at B or inject a message m' *independent* of the messages in $\mathcal{B}$.[7]

The channel is described in more detail in Figure 2.3.

Channel $\mathsf{CONF}_{AB}^{\mathcal{M}}$

init
- $i \leftarrow 0$
- $\mathcal{B} \leftarrow \emptyset$

on $m \in \mathcal{M}$ *at* A
- $i \leftarrow i + 1$
- $\mathcal{B} \leftarrow \mathcal{B} \cup \{m\}$
- **output** $(\mathsf{msg}, i, |m|)$ at E

on (dlv, i') *at* E
- **if** $\exists m : (i', m) \in \mathcal{B}$
  - **output** m at B

on (inj, m') *at* E
- **output** m' at B

Figure 2.3: *Confidential communication channel from A to B.*

2.7.4 Secure Channel

The secure channel $\mathsf{SEC}_{AB}^{\mathcal{M}}$ works as follows: When a message is input at A, it is stored in a buffer $\mathcal{B}$, and its length is output at E. The attacker can (repeatedly) choose a message from the buffer to be delivered to B.

The channel is described in more detail in Figure 2.4.

Channel $\mathsf{SEC}_{AB}^{\mathcal{M}}$

init
- $i \leftarrow 0$
- $\mathcal{B} \leftarrow \emptyset$

on $m \in \mathcal{M}$ *at* A
- $i \leftarrow i + 1$
- $\mathcal{B} \leftarrow \mathcal{B} \cup \{m\}$
- **output** $(\mathsf{msg}, i, |m|)$ at E

on (dlv, i') *at* E
- **if** $\exists m : (i', m) \in \mathcal{B}$
  - **output** m at B

Figure 2.4: *Secure communication channel from A to B.*

2.7.5 Single-Use Channels

The channels $\mathsf{INSEC}_{AB}^{\mathcal{M}}, \mathsf{AUTH}_{AB}^{\mathcal{M}}, \ldots$ defined above also appear as single-use variants, denoted $\mathsf{1\text{-}INSEC}_{AB}^{\mathcal{M}}, \mathsf{1\text{-}AUTH}_{AB}^{\mathcal{M}}, \ldots$, which allow at most one

[7] That is, the confidential channel is *non-malleable*.

Distinguishing Game $\mathbf{G}_b^{\mathsf{pke}}$

init
  $(\mathsf{pk}, \mathsf{sk}) \leftarrow K$
  output pk

on (chall, m_0)
  $m_1 \leftarrow \mathcal{M}$ s.t. $|m_0| = |m_1|$
  $e \leftarrow E_{\mathsf{pk}}(m_b)$
  output e

on (dec, e')
  if $\mathsf{valid}(e')$
    $m' \leftarrow D_{\mathsf{sk}}(e')$
  else
    $m' \leftarrow \mathsf{test}$
  output m'

Figure 2.5: *Distinguishing game used to define various security notions for PKE schemes* $\Pi = (K, E, D)$*. The predicate* $\mathsf{valid}(\cdot)$ *determines whether a ciphertext is permissible to ask at a given point.*

message to be input at A and output at B.

2.8 Public-Key Encryption

A public-key encryption (PKE) scheme with message space $\mathcal{M} \subseteq \{0,1\}^*$ and ciphertext space $\mathcal{E}$ is defined as three algorithms $\Pi = (K, E, D)$, where the key-generation algorithm K outputs a key pair $(\mathsf{pk}, \mathsf{sk})$, the (probabilistic) encryption algorithm E takes a message $m \in \mathcal{M}$ and a public key pk and outputs a ciphertext $e \leftarrow E_{\mathsf{pk}}(m)$, and the decryption algorithm takes a ciphertext $e \in \mathcal{E}$ and a secret key sk and outputs a plaintext $m \leftarrow D_{\mathsf{sk}}(e)$. The output of the decryption algorithm can be the special symbol $\perp$, indicating an invalid ciphertext.

A PKE scheme is correct if $m = D_{\mathsf{sk}}(E_{\mathsf{pk}}(m))$ (with probability 1 over the randomness in the encryption algorithm) for all messages m and all key pairs $(\mathsf{pk}, \mathsf{sk})$ generated by K.

The following standard security notions are considered in this thesis: IND-CPA (CPA for short), IND-CCA1 (CCA1), IND-RCCA (RCCA), and IND-CCA (CCA). All notions are defined via the distinguishing game $\mathbf{G}_b^{\mathsf{pke}}$ depicted in Figure 2.5 (for different predicates $\mathsf{valid}(\cdot)$).[8]

Chosen-plaintext security. *Indistinguishability under chosen-plaintext attacks* (IND-CPA) considers an attacker trying to decide whether a *challenge ciphertext* is an encryption of a plaintext of his choice or an encryption of a random message. The corresponding game $\mathbf{G}_b^{\mathsf{CPA}}$ is obtained from $\mathbf{G}_b^{\mathsf{pke}}$ by letting $\mathsf{valid}(e') = 0$ for all e'.

[8] Game $\mathbf{G}_b^{\mathsf{pke}}$ defines the so-called *real-or-random* versions of these games, which are equivalent to the more popular *left-or-right* formulations (as shown in [BDJR97] for symmetric encryption).

Chosen-ciphertext security. *Indistinguishability under chosen-ciphertext attacks* considers an attacker trying to decide whether a challenge ciphertext is an encryption of a plaintext m_0 of his choice or an encryption of a random message m_1 *while given access to a decryption oracle*:

- *Lunchtime attacks* (IND-CCA1): The decryption oracle is available only before the challenge ciphertext is output. The corresponding game $\mathbf{G}_b^{\mathsf{CCA1}}$ is obtained from $\mathbf{G}_b^{\mathsf{pke}}$ by letting $\mathsf{valid}(e') = 1$ for all e' before the challenge is output and $\mathsf{valid}(e') = 0$ for all e' after the challenge is output.

- *No-replay attacks* (IND-RCCA): The decryption oracle is available at all times with the restriction that no decryption query c' may decrypt to m_0 or m_1. The corresponding game $\mathbf{G}_b^{\mathsf{RCCA}}$ is obtained from $\mathbf{G}_b^{\mathsf{pke}}$ by letting $\mathsf{valid}(e') = 1$ for all e' before the challenge is output and changing to $\mathsf{valid}(e') = 0$ for e' with $D_{\mathsf{sk}}(e') \in \{m_0, m_1\}$ after the challenge was output. For more details about RCCA-security, see Section 3.2.2 or consult [CKN03], where the notion was introduced.

- *Full attacks* (IND-CCA): The decryption oracle is available at all times with the restriction that the challenge ciphertext may not be queried. The corresponding game $\mathbf{G}_b^{\mathsf{CCA}}$ is obtained from $\mathbf{G}_b^{\mathsf{pke}}$ via $\mathsf{valid}(e') = 1$ for all e' before the challenge is output and changing to $\mathsf{valid}(e) = 0$ after the challenge e is output.

Definition 2.2. *Let* $\mathsf{SN} \in \{\mathsf{CPA}, \mathsf{CCA1}, \mathsf{RCCA}, \mathsf{CCA}\}$, $t \in \mathbb{N}$ *and* $\varepsilon \geq 0$. *A PKE scheme* $\Pi = (K, E, D)$ *is* (t, ε)-SN-*secure if*

$$\Delta^{\mathbf{D}}(\mathbf{G}_0^{\mathsf{SN}}, \mathbf{G}_1^{\mathsf{SN}}) \leq \varepsilon$$

for all distinguishers $\mathbf{D}$ *running in time at most* t.

2.9 Coding Schemes, LECSS, and AMD Codes

Definition 2.3 (Coding scheme)**.** *A* (k, n)-coding scheme (Enc, Dec) over a field $\mathbb{F}$ *consists of a randomized encoding function* $\mathrm{Enc} : \mathbb{F}^k \to \mathbb{F}^n$ *and a deterministic decoding function* $\mathrm{Dec} : \mathbb{F}^n \to \mathbb{F}^k \cup \{\perp\}$ *such that* $\mathrm{Dec}(\mathrm{Enc}(m)) = m$ *(with probability* 1 *over the randomness of the encoding function) for each* $m \in \mathbb{F}^k$. *The special symbol* $\perp$ *indicates an* invalid codeword.

The following notions of linear error-detecting/correcting secret sharing, introduced by Dziembowski *et al.* [DPW10], are used in several places in this thesis.

Definition 2.4 (Linear error-detecting sharing scheme)**.** *Let $n \in \mathbb{N}$ be a security parameter and $\mathbb{F}$ a finite field. A (k, n, δ, τ)* linear error-detecting secret sharing (LEDSS) over $\mathbb{F}$ *is a coding scheme (E, D) over $\mathbb{F}$, with the following properties:*

- Linearity: *For any vectors w output by E and any $c \in \mathbb{F}^n$,*

$$\mathsf{D}(w + c) = \begin{cases} \perp & \text{if } \mathsf{D}(c) = \perp, \text{ and} \\ \mathsf{D}(w) + \mathsf{D}(c) & \text{otherwise.} \end{cases}$$

- Minimum distance: *For any $c \in \mathbb{F}^n$ with $0 < w_{\mathsf{H}}(c) < \delta n$, $\mathsf{D}(c) = \perp$.*
- Secrecy: *The symbols of a codeword are individually uniform over $\mathbb{F}$ and τn-wise independent (over the randomness of E).*

Definition 2.5 (Linear error-correcting sharing scheme)**.** *Let $n \in \mathbb{N}$ be a security parameter and $\mathbb{F}$ a finite field. A (k, n, δ, τ)* linear error-correcting secret sharing (LECSS) over $\mathbb{F}$ *is a triple of algorithms $(\mathsf{E}, \mathsf{D}, \mathsf{R})$, where (E, D) is a (k, n, δ, τ)-LEDSS over $\mathbb{F}$, with the additional property:*

- Error correction: *It is possible to efficiently correct up to $\delta n/2$ errors, i.e., for any $m \in \mathbb{F}^k$ and any w output by $\mathsf{E}(m)$, if $d_{\mathsf{H}}(c, w) \leq t$ for some $c \in \mathbb{F}^n$ and $t < \delta n/2$, then $\mathsf{R}(c, t) = w$.*

This paper considers various instantiations of LECSSs, which are described where they are used.

The following concept of algebraic manipulation detection was introduced by Cramer *et al.* [CDF+08], who also provide an instantiation.

Definition 2.6 (AMD code)**.** *A (k, n)-coding scheme (A, V) is a* ρ-secure algebraic manipulation detection (AMD) code *if for all $m \in \{0,1\}^n$ and non-zero $\Delta \in \{0,1\}^n$, $\mathsf{P}[\mathsf{V}(\mathsf{A}(m) + \Delta) \neq \perp] \leq \rho$, where the probability is over the randomness of the encoding algorithm A.*

2.10 One-Time Signatures

A *digital signature scheme* (DSS) is a triple of algorithms $\Sigma = (K, S, V)$, where the key-generation algorithm K outputs a key pair $(\mathsf{sk}, \mathsf{vk})$, the (probabilistic) signing algorithm S takes a message m and a signing key sk and outputs a signature $s \leftarrow S_{\mathsf{sk}}(m)$, and the verification algorithm takes a verification key vk, a message m, and a signature s and outputs a single bit $V_{\mathsf{vk}}(m, s)$. A *(strong) one-time signature (OTS) scheme* is

a digital signature scheme that is secure as long as an adversary only observes a single signature. More precisely, OTS security is defined using the following game $\mathbf{G}^{\mathsf{ots}}$ played by an adversary $\mathbf{A}$: Initially, the game generates a key pair $(\mathsf{sk}, \mathsf{vk})$ and hands the verification key vk to $\mathbf{A}$. Then, $\mathbf{A}$ can specify a single message m for which he obtains a signature $s \leftarrow S_{\mathsf{vk}}(m)$. Then, the adversary outputs a pair (m', s'). The adversary wins the game if $(m', s') \neq (m, s)$ and $V_{\mathsf{vk}}(m', s') = 1$. The *advantage* of $\mathbf{A}$ is the probability (over all involved randomness) that $\mathbf{A}$ wins the game, and is denoted by $\Gamma^{\mathbf{A}}(\mathbf{G}^{\mathsf{ots}})$.

Definition 2.7. *A DSS scheme Σ is a (t, ε)*-strong one-time signature scheme *if for all adversaries $\mathbf{A}$ with running time at most t, $\Gamma^{\mathbf{A}}(\mathbf{G}^{\mathsf{ots}}) \leq \varepsilon$.*

2.11 Chernoff Bound

The following (standard) Chernoff bound is used.

Theorem 2.2. *Let $X_1, \ldots, X_n$ be i.i.d. with $X_i \sim \mathrm{Be}(p_i)$. Then, for $X := \sum_i X_i$ and $\mu := \sum_i p_i$,*

$$\mathsf{P}[X \leq (1-\varepsilon)\mu] \quad \leq \quad e^{-\mu\varepsilon^2/2}$$

for any $\varepsilon \in (0, 1]$.

2.12 Plotkin Bound

The following theorem allows to bound the number of codewords of a code over a binary alphabet with relative minimum distance $\delta > 1/2$.

Theorem 2.3. *For a code over a binary alphabet with block length n and distance $d \geq \frac{n}{2} + 1$, the maximum number of codewords is*

$$A(n, d) \leq \frac{d}{d - \frac{n}{2}} \leq 1 + \frac{1}{2\varepsilon}$$

where $\varepsilon = \frac{d}{n} - \frac{1}{2}$.

A proof can be found in [MS78, p. 41].

Chapter 3

A Constructive Perspective on Public-Key Encryption

From the perspective of constructive cryptography (CC), the purpose of a public-key encryption (PKE) scheme is to construct a confidential channel from non-confidential channels between a sender and a receiver. Section 3.1 analyzes two such channel constructions based on public-key encryption: one where the communication from the sender to the receiver is authenticated and one where it is not. For either scenario, the appropriate security level required of the PKE for the construction to work is identified.

In Section 3.2 constructive semantics are assigned to several existing game-based definitions by first characterizing the appropriate assumed and constructed resources and then showing that the "standard use" of a PKE scheme over those channels (cf. Section 3.1.1) achieves the construction if (and sometimes only if) it has the considered property.

Finally, Section 3.3 explains how the three-party scenario considered here actually captures settings with multiple senders and receivers, and Section 3.4 points out some differences between the channel-based approach taken here and approaches that idealize the properties of cryptographic schemes.

3.1 Constructing Confidential Channels with PKE

The main purpose of public-key encryption (PKE) is to achieve confidential communication. As a constructive statement, this means that a PKE

scheme Π is viewed as a protocol, a pair of converters $\mathsf{pke} = (\mathsf{enc}, \mathsf{dec})$, whose goal is to construct a confidential channel from non-confidential channels. Differentiating between the two cases where the communication from the sender to the receiver is authenticated and unauthenticated, respectively, this corresponds to the two constructions[1]

$$[\text{1-}\mathsf{AUTH}^{\mathcal{K}}_{BA}, \mathsf{AUTH}^{\mathcal{E}}_{AB}] \quad \overset{\mathsf{pke}}{\Longrightarrow} \quad \mathsf{SEC}^{\mathcal{M}}_{AB} \tag{3.1}$$

and

$$[\text{1-}\mathsf{AUTH}^{\mathcal{K}}_{BA}, \mathsf{INSEC}^{\mathcal{E}}_{AB}] \quad \overset{\mathsf{pke}}{\Longrightarrow} \quad \mathsf{CONF}^{\mathcal{M}}_{AB}, \tag{3.2}$$

where $\mathcal{K}$, $\mathcal{E}$, and $\mathcal{M}$ are the key space, ciphertext space, and plaintext space of Π. For readability, these superscripts are dropped in the rest of this chapter.

In both cases, the *single-use* channel $\text{1-}\mathsf{AUTH}_{BA}$ captures the ability of the sender to obtain the receiver's public key in an authenticated fashion. In construction (3.1), the communication from the sender A to the receiver B is authenticated, which is modeled by the channel AUTH_{AB}. The goal is to achieve a secure channel SEC_{AB}, which only leaks the length of the messages sent at interface A. In construction (3.2), the communication from A to B is completely insecure, which is captured by the insecure channel INSEC_{AB}. In this case, the goal is to achieve a confidential channel CONF_{AB}, which still hides messages input at the A-interface but also allows to inject arbitrary messages (unrelated to those sent by A) at E.

In the following, Section 3.1.1 first explains how a PKE scheme Π can be transformed into a converter pair $\mathsf{pke} = (\mathsf{enc}, \mathsf{dec})$. Then, in Section 3.1.2, it is shown that pke achieves construction (3.1) if the underlying PKE scheme is CPA-secure, and Section 3.1.3 shows that construction (3.2) is realized if the underlying PKE scheme is CCA-secure. Section 3.1.5 discusses how CCA security can be exploited to construct channels with replay protection. Finally, Section 3.1.6 briefly discusses the applicability of the channels constructed in this section.

3.1.1 PKE Schemes as Protocols

Let $\Pi = (K, E, D)$ be a PKE scheme. Based on Π, a pair of protocol converters $\mathsf{pke} = (\mathsf{enc}, \mathsf{dec})$ for constructions (3.1) and (3.2) can be defined as shown below. Both converters have two sub-interfaces `in`.1 and `in`.2 on the inside, as they are connected to a resource that is a parallel composition of two other resources (cf. Section 2.2).

[1] Consult Section 2.7 for the definitions of the channels.

Converter enc works as follows: It initially expects a public key pk at in.1. When a message m is input at the outside interface out, enc outputs $e \leftarrow E_{\mathsf{pk}}(m)$ at in.2. Converter dec initially generates a key pair $(\mathsf{pk}, \mathsf{sk})$ using key-generation algorithm K and outputs pk at in.1. When dec receives e' at in.2, it computes $m' \leftarrow D_{\mathsf{sk}}(e')$ and, if $m' \neq \perp$, outputs m' at the outside interface out.

3.1.2 Secure Channel from Authenticated Channel

Towards proving that protocol pke indeed achieves construction (3.1), note first that the correctness of Π implies that the *availability* condition of Definition 2.1 is satisfied. To prove *security*, one needs to exhibit a simulator σ such that the assumed resource $[\mathsf{1\text{-}AUTH}_{BA}, \mathsf{AUTH}_{AB}]$ with the protocol converters is indistinguishable from the constructed resource SEC_{AB} with the simulator (cf. Figure 3.1).

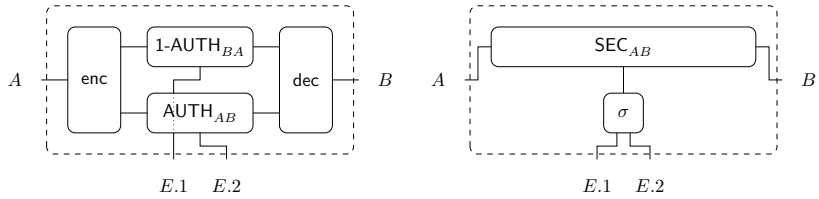

Figure 3.1: *Left: The assumed resource (two authenticated channels) with protocol converters* enc *and* dec *attached to interfaces A and B, denoted* $\mathsf{enc}^A\mathsf{dec}^B[\mathsf{1\text{-}AUTH}_{BA}, \mathsf{AUTH}_{AB}]$. *Right: The constructed resource (a secure channel) with simulator σ attached to the E-interface, denoted* $\sigma^E\mathsf{SEC}_{AB}$. *In particular, σ must simulate the E-interfaces of the two authenticated channels. The protocol is secure if the two systems are indistinguishable.*

Let $\langle\mathsf{AUTH}_{AB}\rangle_n$ and $\langle\mathsf{SEC}_{AB}\rangle_n$ denote the authenticated resp. secure channels processing only the first n messages at interface A. Theorem 3.1 implies that $(\mathsf{enc}, \mathsf{dec})$ realizes (3.1) if the underlying PKE scheme is CPA-secure.

Theorem 3.1. *There exists a simulator σ and for any $n \in \mathbb{N}$ there exists a (efficient) reduction* $\mathbf{C}$ *such that for every* $\mathbf{D}$,

$$\Delta^{\mathbf{D}}(\mathsf{enc}^A\mathsf{dec}^B[\mathsf{1\text{-}AUTH}_{BA}, \langle\mathsf{AUTH}_{AB}\rangle_n], \sigma^E\langle\mathsf{SEC}_{AB}\rangle_n) \leq n \cdot \Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{CPA}}, \mathbf{G}_1^{\mathsf{CPA}}).$$

Proof. First, consider the following simulator σ for interface E of SEC_{AB}, which has two sub-interfaces denoted by $\mathtt{out}.1$ and $\mathtt{out}.2$ on the outside (since the real-world system has two sub-interfaces at E): Initially, σ generates a key pair $(\mathsf{pk}, \mathsf{sk})$ and outputs $(\mathsf{msg}, 1, \mathsf{pk})$ at $\mathtt{out}.1$.[2] When it receives (msg, i, l) at the inside interface $\mathtt{in}$, σ generates an encryption $e \leftarrow E_{\mathsf{pk}}(m_1)$ of a randomly chosen message m_1 of length l and outputs (msg, i, e) at $\mathtt{out}.2$. When (dlv, i') is input at $\mathtt{out}.2$, σ simply outputs (dlv, i') at $\mathtt{in}$. Consider the two systems

$$\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle \mathsf{AUTH}_{AB} \rangle_1] \qquad \text{and} \qquad \sigma^E \langle \mathsf{SEC}_{AB} \rangle_1.$$

Distinguishing $\mathbf{G}_0^{\mathsf{CPA}}$ from $\mathbf{G}_1^{\mathsf{CPA}}$ can be reduced to distinguishing these two systems via the following reduction system $\mathbf{C}'$, which connects to a game on the inside and provides interfaces A, B, and E on the outside (cf. Section 2.2 for details on reduction systems): Initially, $\mathbf{C}'$ takes a value pk from the game (on the inside) and outputs $(\mathsf{msg}, 1, \mathsf{pk})$ at the (outside) $E.1$-interface. When a message m is input at the A-interface of $\mathbf{C}'$, it is passed as (chall, m) to the game. The resulting challenge e is output as $(\mathsf{msg}, 1, e)$ at the $E.2$-interface. When $(\mathsf{dlv}, 1)$ is input at the $E.2$-interface, $\mathbf{C}'$ outputs m at interface B. By inspection, one verifies that

$$\mathbf{C}'\mathbf{G}_0^{\mathsf{CPA}} \equiv \mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle \mathsf{AUTH}_{AB} \rangle_1]$$

and

$$\mathbf{C}'\mathbf{G}_1^{\mathsf{CPA}} \equiv \sigma^E \langle \mathsf{SEC}_{AB} \rangle_1,$$

and thus

$$\begin{aligned}
&\Delta^{\mathbf{D}}(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle \mathsf{AUTH}_{AB} \rangle_n], \sigma^E \langle \mathsf{SEC}_{AB} \rangle_n) \\
&\quad \leq n \cdot \Delta^{\mathbf{DC}''}(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle \mathsf{AUTH}_{AB} \rangle_1], \sigma^E \langle \mathsf{SEC}_{AB} \rangle_1) \\
&\quad = n \cdot \Delta^{\mathbf{DC}''}(\mathbf{C}'\mathbf{G}_0^{\mathsf{CPA}}, \mathbf{C}'\mathbf{G}_1^{\mathsf{CPA}}) \\
&\quad = n \cdot \Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{CPA}}, \mathbf{G}_1^{\mathsf{CPA}}),
\end{aligned}$$

where $\mathbf{C} := \mathbf{C}''\mathbf{C}'$ and the first inequality follows from Lemma 3.3, which is a standard hybrid argument, for a reduction system $\mathbf{C}''$. □

3.1.3 Confidential Channel from Insecure Channel

To prove that protocol pke achieves construction (3.2), one needs to exhibit a simulator σ such that the assumed resource $[\text{1-}\mathsf{AUTH}_{BA}, \mathsf{INSEC}_{AB}]$ with

[2] For simplicity, it is assumed that the public key is always delivered, i.e., that $(\mathsf{dlv},)$ is input at interface E of $\text{1-}\mathsf{AUTH}_{BA}$.

the protocol converters is indistinguishable from the constructed resource CONF_{AB} with the simulator.

Let $\langle\mathsf{INSEC}_{AB}\rangle_n$ and $\langle\mathsf{CONF}_{AB}\rangle_n$ denote the insecure resp. confidential channels processing only the first n messages at interface A. Theorem 3.2 implies that $(\mathsf{enc}, \mathsf{dec})$ realizes (3.2) if the underlying PKE scheme is CCA-secure.

Theorem 3.2. *There exists a simulator σ and for any $n \in \mathbb{N}$ there exists a (efficient) reduction $\mathbf{C}$ such that for every $\mathbf{D}$,*

$$\Delta^{\mathbf{D}}(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle\mathsf{INSEC}_{AB}\rangle_n], \sigma^E\langle\mathsf{CONF}_{AB}\rangle_n) \leq n \cdot \Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{CCA}}, \mathbf{G}_1^{\mathsf{CCA}}).$$

Note that the confidential channel CONF_{AB} is the best channel one can construct from the two assumed channels. As the E-interface has the same capabilities as the A-interface at both the authenticated (from B to A) and the insecure channels, it will necessarily also be possible to inject messages to the receiver via the E-interface by simply applying the sender's protocol converter.

Proof. First, consider the following simulator σ for interface E of CONF_{AB}, which again has two outside sub-interfaces `out`.1 and `out`.2: Initially, it generates a key pair $(\mathsf{pk}, \mathsf{sk})$ and outputs $(\mathsf{msg}, 1, \mathsf{pk})$ at `out`.1. When it receives (msg, i, l) at the inside interface `in`, it generates an encryption $e \leftarrow E_{\mathsf{pk}}(m_1)$ of a randomly chosen message m_1 of length l, outputs (i, e) at `out`.2, and records (c, i). When (inj, e') is input at `out`.2, σ proceeds as follows: If (e', i') has been recorded for some i', it outputs (dlv, i') at `in`. Otherwise, it computes $e' \leftarrow D_{\mathsf{sk}}(e')$ and, if $m' \neq \bot$, outputs (inj, m') at `in`.

Consider now the problem of distinguishing the two systems

$$\mathbf{U} := \mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle\mathsf{INSEC}_{AB}\rangle_1] \quad \text{and} \quad \mathbf{V} := \sigma^E\langle\mathsf{CONF}_{AB}\rangle_1,$$

which are depicted in Figure 3.2.

A distinguisher $\mathbf{D}$ connected to the real-world system $\mathbf{U}$ initially sees a public key at interface E.1. If $\mathbf{D}$ inputs a message m at interface A, an encryption of m (created by enc) is output at interface E.2. When $\mathbf{D}$ inputs a ciphertext e' at E, it sees the decryption of e' (by dec) at B. The ideal-world system $\mathbf{V}$ behaves differently: Initially, $\mathbf{D}$ also sees a public key at E.1. But when it inputs a message m at A, an encryption e of a randomly chosen message is output at interface E.2 (by simulator σ).

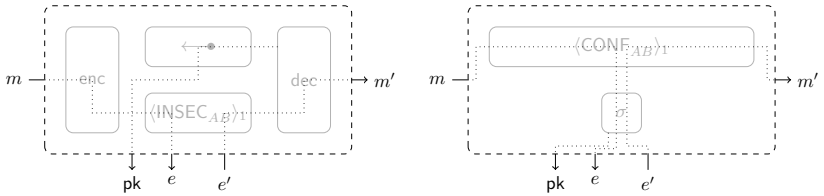

Figure 3.2: *The systems* $\mathbf{U}$ *and* $\mathbf{V}$ *with the "message flow" from the perspective of a distinguisher: Initially, a public-key* pk *is output at interface* E*. Inputting a message* m *at interface* A *causes a ciphertext* e *to be output at the* E*-interface. Note that* e *is the challenge in the* CCA*-game. Inputting a ciphertext* e' *at interface* E *results in a message* m' *being output at* B*. This corresponds to the decryption oracle in the* CCA*-game.*

When e is input at interface $E.2$, m is output at B (as σ issues a $(\mathsf{dlv}, \cdot)$-instruction to the channel). When $e' \neq e$ is input at $E.2$, the decryption of e' (injected by σ) is output at B.

The translation between the channel setting and the game setting is achieved by the following reduction system $\mathbf{C}'$: Initially, $\mathbf{C}'$ takes a value pk from the game (on the inside) and outputs it as $(\mathsf{msg}, 1, \mathsf{pk})$ at the (outside) $E.1$-interface. When a message m is input at interface A of $\mathbf{C}'$, (chall, m) is output to the game. The resulting challenge e is output as $(\mathsf{msg}, 1, e)$ at interface $E.2$. When (inj, e) is input at interface $E.2$, $\mathbf{C}'$ outputs m at interface B. When (inj, e') with $e' \neq e$ is input at interface $E.2$, $\mathbf{C}'$ passes (dec, e') to the game's decryption oracle and outputs the answer m' at interface B, provided $m' \neq \perp$.

By inspection, one verifies that

$$\mathbf{C}'\mathbf{G}_0^{\mathsf{CCA}} \equiv \mathsf{enc}^A\mathsf{dec}^B[\mathsf{1\text{-}AUTH}_{BA}, \langle \mathsf{INSEC}_{AB} \rangle_1]$$

and

$$\mathbf{C}'\mathbf{G}_1^{\mathsf{CCA}} \equiv \sigma^E \langle \mathsf{CONF}_{AB} \rangle_1,$$

and thus

$$\begin{aligned}
&\Delta^{\mathbf{D}}(\mathsf{enc}^A\mathsf{dec}^B[\mathsf{1\text{-}AUTH}_{BA}, \langle \mathsf{INSEC}_{AB} \rangle_n], \sigma^E \langle \mathsf{CONF}_{AB} \rangle_n) \\
&\qquad \leq n \cdot \Delta^{\mathbf{DC}''}(\mathsf{enc}^A\mathsf{dec}^B[\mathsf{1\text{-}AUTH}_{BA}, \langle \mathsf{INSEC}_{AB} \rangle_1], \sigma^E \langle \mathsf{CONF}_{AB} \rangle_1) \\
&\qquad = n \cdot \Delta^{\mathbf{DC}''}(\mathbf{C}'\mathbf{G}_0^{\mathsf{CCA}}, \mathbf{C}'\mathbf{G}_1^{\mathsf{CCA}}) \\
&\qquad = n \cdot \Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{CCA}}, \mathbf{G}_1^{\mathsf{CCA}}),
\end{aligned}$$

where $\mathbf{C} := \mathbf{C}''\mathbf{C}'$ and the first inequality follows from Lemma 3.4, which is a standard hybrid argument, for a reduction system $\mathbf{C}''$. □

3.1.4 Multi-Message Security

Let $(\mathsf{enc}, \mathsf{dec})$ be a protocol constructed from a PKE scheme as shown in Section 3.1.1.

Lemma 3.3. *For every $n \in \mathbb{N}$ there exists a (efficient) reduction $\mathbf{C}''$ such that*

$$\Delta^{\mathbf{D}}(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle \mathsf{AUTH}_{AB}\rangle_n], \sigma^E\langle \mathsf{SEC}_{AB}\rangle_n) \\ \leq n \cdot \Delta^{\mathbf{DC}''}(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle \mathsf{AUTH}_{AB}\rangle_1], \sigma^E\langle \mathsf{SEC}_{AB}\rangle_1),$$

where σ is the simulator from Theorem 3.1.

Proof. Omitted (as similar to the proof of Lemma 3.4). □

Lemma 3.4. *For every $n \in \mathbb{N}$ there exists a (efficient) reduction $\mathbf{C}''$ such that*

$$\Delta^{\mathbf{D}}(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle \mathsf{INSEC}_{AB}\rangle_n], \sigma^E\langle \mathsf{CONF}_{AB}\rangle_n) \\ \leq n \cdot \Delta^{\mathbf{DC}''}(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle \mathsf{INSEC}_{AB}\rangle_1], \sigma^E\langle \mathsf{CONF}_{AB}\rangle_1),$$

where σ is the simulator from Theorem 3.2.

Proof. Let $\mathbf{D}$ be an arbitrary distinguisher. For $i = 1, \ldots, n$, consider the following reduction system $\mathbf{C}''_i$ (which processes at most n inputs at the outside A interface): Initially, $\mathbf{C}''_i$ forwards a public key pk from the inside $E.1$-interface to the outside $E.1$-interface. When the j^{th} message m_0 is input at the outside A-interface, if $j < i$, $\mathbf{C}''_i$ randomly chooses a random message m_1 of length $|m_0|$ and computes $e \leftarrow E_{\mathsf{pk}}(m_1)$, if $j = i$, it outputs m_0 at the inside A-interface and obtains e at the inside $E.2$-interface, and if $j > i$ it computes $e \leftarrow E_{\mathsf{pk}}(m_0)$. In all cases, it outputs (msg, j, e) at the outside $E.2$-interface and records (e, m). When (inj, e') is input at the outside $E.2$-interface, if (e', m') has been recorded for some m', m' is output at the outside B-interface, and otherwise (inj, e') is output at the inside $E.2$-interface and the subsequently received message m' at the inside B-interface is output at the outside B-interface. Note that

$$\mathbf{C}''_1\left(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle \mathsf{INSEC}_{AB}\rangle_1]\right) \\ \equiv \mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle \mathsf{INSEC}_{AB}\rangle_n]$$

and

$$\mathbf{C}''_n\left(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA},\langle\mathsf{INSEC}_{AB}\rangle_1]\right)\equiv\sigma^E\langle\mathsf{CONF}_{AB}\rangle_n.$$

Moreover, for $i=1,\ldots,n-1$, we have

$$\mathbf{C}''_{i-1}\left(\sigma^E\langle\mathsf{CONF}_{AB}\rangle_1\right)\equiv\mathbf{C}''_i\left(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA},\langle\mathsf{INSEC}_{AB}\rangle_1]\right).$$

Thus for the reduction $\mathbf{C}''$ that chooses i uniformly at random from $\{1,\ldots,n\}$ and then implements $\mathbf{C}''_i$,

$$\begin{aligned}&\Delta^{\mathbf{DC}''}(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA},\langle\mathsf{INSEC}_{AB}\rangle_1],\sigma^E\langle\mathsf{CONF}_{AB}\rangle_1)\\&\quad=\frac{1}{n}\Delta^{\mathbf{D}}(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA},\langle\mathsf{INSEC}_{AB}\rangle_n],\sigma^E\langle\mathsf{CONF}_{AB}\rangle_n).\end{aligned}$$

□

3.1.5 Replay-Protected Channels from CCA-Security

As discussed later in Section 3.2.2, CCA-security is overly strict in that only the weaker RCCA-security is necessary to achieve construction (3.2). In fact, using a CCA-secure PKE scheme one can construct a replay-protected confidential channel, which works as CONF_{AB} with the exception that for any index i', the query (dlv,i') is processed at most once. The corresponding protocol converters $(\mathsf{enc}',\mathsf{dec}')$ are built as $(\mathsf{enc},\mathsf{dec})$ in Section 3.1.1, except that dec' processes every ciphertext received at `in`.2 only once. Similarly, the corresponding simulator σ' also processes every ciphertext received at `out`.2 only once.[3]

3.1.6 Applicability of the Constructed Channels

The plain use of PKE yields constructions (3.1) and (3.2), i.e., one obtains the resources SEC_{AB} and CONF_{AB}. Both channels allow the adversary to reorder or replay the messages sent by A. In practice, where PKE is often used to encapsulate symmetric keys, it is important, however, that keys used in various protocols by different users be independent. Thus, it is more useful to obtain independent single-use channels

$$[\text{1-}\mathsf{SEC}_{AB},\ldots,\text{1-}\mathsf{SEC}_{AB}]\quad\text{and}\quad[\text{1-}\mathsf{CONF}_{AB},\ldots,\text{1-}\mathsf{CONF}_{AB}]$$

[3]Note that, in fact, an SD-RCCA-secure PKE scheme suffices (cf. [CKN03] for more details). In this case, dec' and σ' process only one ciphertext per equivalence class.

instead of SEC_{AB} and CONF_{AB}, respectively.

In the authenticated setting, given independent authenticated channels, protocol $(\mathsf{enc}, \mathsf{dec})$ (with only formal modifications) achieves the construction

$$[\text{1-}\mathsf{AUTH}_{AB}, \ldots, \text{1-}\mathsf{AUTH}_{AB}] \overset{(\mathsf{enc},\mathsf{dec})}{\longmapsto} [\text{1-}\mathsf{SEC}_{AB}, \ldots, \text{1-}\mathsf{SEC}_{AB}].$$

In the unauthenticated setting, however, the analogous construction

$$[\text{1-}\mathsf{INSEC}_{AB}, \ldots, \text{1-}\mathsf{INSEC}_{AB}] \overset{(\mathsf{enc},\mathsf{dec})}{\longmapsto} [\text{1-}\mathsf{CONF}_{AB}, \ldots, \text{1-}\mathsf{CONF}_{AB}]$$

is not achieved by $(\mathsf{enc}, \mathsf{dec})$ since, due to the absence of authenticity, the adversary can freely take a ciphertext it observes on any of the insecure channels $\text{1-}\mathsf{INSEC}_{AB}$ and insert it into another one. Thus, the ideal resource cannot consist of independent channels. This issue can be taken care of by (explicitly) introducing session identifiers (SIDs). A systematic treatment of SIDs and handling multiple sessions and senders can be found in [MTC13].

3.2 Constructive Semantics of Game-Based Notions

In this section, several game-based security notions are analyzed from a constructive viewpoint. The analysis of CPA-security from Section 3.1.2 is completed in Section 3.2.1 by showing that it is also necessary to achieve construction (3.1). Moreover, as shown in Section 3.2.2, the notion of CCA is unnecessarily strict for construction (3.2), which in fact only requires the weaker notion of RCCA introduced in [CKN03].

Bellare *et al.* [BHK09] compare several variants of defining CCA-security. As pointed out in Section 3.2.4 below, only the stricter notions they consider are sufficient for construction (3.2). Finally, Section 3.2.5 gives constructive semantics to the notion of IND-CCA1.[4]

3.2.1 Necessity of CPA Security

As proved in Section 3.1.2, indistinguishability under chosen-plaintext attacks, IND-CPA-security, suffices to construct a secure channel from two authenticated channels. It turns out that it is also necessary. That is, if protocol $\mathsf{pke} = (\mathsf{enc}, \mathsf{dec})$, based on a PKE scheme Π as shown in Section 3.1.1, achieves the construction, then Π must be CPA-secure.

[4] A similar treatment is provided for *non-malleability* in Section 4.3.

In the following, let

$$\mathbf{U} := \mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \mathsf{AUTH}_{AB}] \qquad \text{and} \qquad \mathbf{V} := \sigma^E\mathsf{SEC}_{AB},$$

where σ is an *arbitrary* simulator.

Theorem 3.5. *There exist (efficient) reductions* $\mathbf{C}_0$ *and* $\mathbf{C}_1$ *such that for all adversaries* $\mathbf{A}$*,*

$$\Delta^{\mathbf{A}}(\mathbf{G}_0^{\mathsf{CPA}}, \mathbf{G}_1^{\mathsf{CPA}}) \leq \Delta^{\mathbf{AC}_0}(\mathbf{U}, \mathbf{V}) + \Delta^{\mathbf{AC}_1}(\mathbf{U}, \mathbf{V}).$$

Proof. Consider the following reduction systems $\mathbf{C}_0$ and $\mathbf{C}_1$, both connecting to an $\{A, B, E\}$-resource on the inside and providing a single interface on the outside (for the adversary): Initially, both obtain $(\mathsf{msg}, 1, \mathsf{pk})$ at the inside $E.1$-interface and output pk at the outside interface. When (chall, m_0) is received on the outside, $\mathbf{C}_0$ outputs m_0 at the inside A-interface and $\mathbf{C}_1$ a randomly chosen message m_1 of length $|m_0|$. Subsequently, $(\mathsf{msg}, 1, e)$ is received at the inside $E.2$-interface, and e is output (as the challenge) on the outside by both systems. It holds that

$$\mathbf{C}_0\mathbf{U} \equiv \mathbf{G}_0^{\mathsf{CPA}} \qquad \text{and} \qquad \mathbf{C}_1\mathbf{U} \equiv \mathbf{G}_1^{\mathsf{CPA}} \qquad \text{and} \qquad \mathbf{C}_0\mathbf{V} \equiv \mathbf{C}_1\mathbf{V},$$

where the last equivalence follows from the fact that, in $\mathbf{V}$, the input from SEC_{AB} to σ is the same in both systems (the length of the message input at the A-interface of SEC_{AB}), and therefore they behave identically. Hence,

$$\begin{aligned}\Delta^{\mathbf{A}}(\mathbf{G}_0^{\mathsf{CPA}}, \mathbf{G}_1^{\mathsf{CPA}}) &= \Delta^{\mathbf{A}}(\mathbf{C}_0\mathbf{U}, \mathbf{C}_1\mathbf{U}) \\ &\leq \Delta^{\mathbf{A}}(\mathbf{C}_0\mathbf{U}, \mathbf{C}_0\mathbf{V}) + \Delta^{\mathbf{A}}(\mathbf{C}_0\mathbf{V}, \mathbf{C}_1\mathbf{V}) + \Delta^{\mathbf{A}}(\mathbf{C}_1\mathbf{V}, \mathbf{C}_1\mathbf{U}) \\ &= \Delta^{\mathbf{AC}_0}(\mathbf{U}, \mathbf{V}) + \Delta^{\mathbf{AC}_1}(\mathbf{U}, \mathbf{V}).\end{aligned}$$

□

3.2.2 Necessity of Replayable CCA Security

Indistinguishability under chosen-ciphertext attacks, CCA-security, suffices to construct a confidential channel from an authenticated and an insecure one (cf. Section 3.1.3). It is, however, unnecessarily strict, as can be seen from the following example, adapted from [CKN03]: Let Π be a PKE scheme and assume it is CCA-secure. Consider a modified scheme Π' that works exactly as Π, except that a 0-bit is appended to every encryption, which is ignored during decryption. It is easily seen that Π' is

not CCA-secure, since the adversary can obtain a decryption of the challenge ciphertext by flipping its last bit and submitting the result to the decryption oracle. PKE scheme Π' can, however, still be used to achieve construction (3.2) using a simulator that also issues the $(\mathsf{dlv}, \cdot)$-instruction to CONF_{AB} when flipping the last bit of a ciphertext received at the outside interface results in a recorded ciphertext (but otherwise works like σ from Theorem 3.2).

Canetti *et al.* [CKN03] introduced the notion of *replayable chosen ciphertext* security, RCCA, which is more permissive in that it allows the adversary to transform a ciphertext into one that decrypts to the same message. As shown below if protocol $(\mathsf{enc}, \mathsf{dec})$, based on a PKE scheme Π as defined in Section 3.1.1, achieves construction (3.2), then Π must be RCCA-secure. Note that RCCA is also sufficient for the construction if the message space of Π is sufficiently large (cf. Section 3.2.3).

In the following, let

$$\mathbf{U} := \mathsf{enc}^A\mathsf{dec}^B[\mathsf{1\text{-}AUTH}_{BA}, \mathsf{INSEC}_{AB}] \qquad \text{and} \qquad \mathbf{V} := \sigma^E\mathsf{CONF}_{AB},$$

where σ is an *arbitrary* simulator.

Theorem 3.6. *There exist (efficient) reductions* $\mathbf{C}_0$ *and* $\mathbf{C}_1$ *such that for all adversaries* $\mathbf{A}$*,*

$$\Delta^{\mathbf{A}}(\mathbf{G}_0^{\mathsf{RCCA}}, \mathbf{G}_1^{\mathsf{RCCA}}) \leq \Delta^{\mathbf{AC}_0}(\mathbf{U}, \mathbf{V}) + \Delta^{\mathbf{AC}_1}(\mathbf{U}, \mathbf{V}).$$

Proof. Consider the following reductions $\mathbf{C}_0$ and $\mathbf{C}_1$. Again, both connect to an $\{A, B, E\}$-resource on the inside and provide a single interface on the outside: Initially, both obtain $(\mathsf{msg}, 1, \mathsf{pk})$ at the inside $E.1$-interface and output pk at the outside interface. When (chall, m_0) is received on the outside, *both* systems choose a random message m_1. $\mathbf{C}_0$ outputs m_0 at the inside A-interface and $\mathbf{C}_1$ outputs m_1. Subsequently, $(\mathsf{msg}, 1, e)$ is received at the inside E-interface, and c is output on the outside by both systems. When a decryption query (dec, e') is received on the outside, both systems output (inj, e') at the inside $E.2$-interface. A subsequently received message m' at B is output on the outside by both systems (as answer to the decryption query) unless $m' \in \{m_0, m_1\}$, in which case test is returned. It holds that

$$\mathbf{C}_0\mathbf{U} \equiv \mathbf{G}_0^{\mathsf{RCCA}} \qquad \text{and} \qquad \mathbf{C}_1\mathbf{U} \equiv \mathbf{G}_1^{\mathsf{RCCA}} \qquad \text{and} \qquad \mathbf{C}_0\mathbf{V} \equiv \mathbf{C}_1\mathbf{V},$$

where the last equivalence follows from the fact that, in $\mathbf{V}$, the input from SEC_{AB} to σ is the same in both systems (the length of the message input

at the A-interface of SEC_{AB}) and that decryption queries causing m_0 or m_1 to be output at the B-interface are answered by test. Hence,

$$\begin{aligned}\Delta^{\mathbf{A}}(\mathbf{G}_0^{\mathsf{RCCA}}, \mathbf{G}_1^{\mathsf{RCCA}}) &= \Delta^{\mathbf{A}}(\mathbf{C}_0\mathbf{U}, \mathbf{C}_1\mathbf{U}) \\ &\leq \Delta^{\mathbf{A}}(\mathbf{C}_0\mathbf{U}, \mathbf{C}_0\mathbf{V}) + \Delta^{\mathbf{A}}(\mathbf{C}_0\mathbf{V}, \mathbf{C}_1\mathbf{V}) \\ &\quad + \Delta^{\mathbf{A}}(\mathbf{C}_1\mathbf{V}, \mathbf{C}_1\mathbf{U}) \\ &= \Delta^{\mathbf{AC}_0}(\mathbf{U}, \mathbf{V}) + \Delta^{\mathbf{AC}_1}(\mathbf{U}, \mathbf{V}).\end{aligned}$$

□

3.2.3 Sufficiency of Replayable CCA Security

To settle the question of equivalence between transformation (3.2) and RCCA-security, it remains to see whether RCCA-security suffices to achieve (3.2). It turns out that this is the case if the message space $\mathcal{M}$ of the underlying PKE is large. For simplicity, it is assumed that all messages in $\mathcal{M}$ have equal length.

Theorem 3.7. *There exist a simulator σ and for any $n \in \mathbb{N}$ there exists a (efficient) reduction $\mathbf{C}$ such that for every $\mathbf{D}$,*

$$\begin{aligned}&\Delta^{\mathbf{D}}(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle\mathsf{INSEC}_{AB}\rangle_n], \sigma^E\langle\mathsf{CONF}_{AB}\rangle_n) \\ &\qquad\leq n \cdot \Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{RCCA}}, \mathbf{G}_1^{\mathsf{RCCA}}) + \frac{n^2}{|\mathcal{M}|} \quad (3.3)\end{aligned}$$

Proof (sketch). One first shows that

$$\begin{aligned}&\Delta^{\mathbf{D}}(\mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle\mathsf{INSEC}_{AB}\rangle_1], \sigma^E\langle\mathsf{CONF}_{AB}\rangle_1) \\ &\qquad\leq \Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{RCCA}}, \mathbf{G}_1^{\mathsf{RCCA}}) + \frac{n}{|\mathcal{M}|}. \quad (3.4)\end{aligned}$$

The proof can be generalized using a standard hybrid argument.

Consider the following simulator σ (with two outside sub-interfaces $\mathtt{out}.1$ and $\mathtt{out}.2$): Initially, σ generates a key pair $(\mathsf{pk}, \mathsf{sk})$ and outputs $(\mathsf{msg}, 1, \mathsf{pk})$ at $\mathtt{out}.1$. When it receives (msg, i, l) at the inside interface $\mathtt{in}$, it generates an encryption $e \leftarrow E_{\mathsf{pk}}(m_1)$ of a randomly chosen message m_1 (of length l), outputs (msg, i, e) at $\mathtt{out}.2$, and records (m, i). When (inj, e') is input at $\mathtt{out}.2$, σ proceeds as follows: It computes $m' \leftarrow D_{\mathsf{sk}}(c')$. If (m', i') has been recorded for some i', it outputs (dlv, i') at its inside interface. Otherwise, if $m' \neq \perp$, it outputs (inj, m') at the inside interface. Set

$$\mathbf{U} := \mathsf{enc}^A\mathsf{dec}^B[\text{1-}\mathsf{AUTH}_{BA}, \langle\mathsf{INSEC}_{AB}\rangle_1]$$

and

$$\mathbf{V} := \sigma^E \langle \mathsf{CONF}_{AB} \rangle_1.$$

The translation between the game and the channel setting is done by the reduction $\mathbf{C}$. Initially, $\mathbf{C}$ takes pk from the game and outputs it at the E-interface. When a message m is input at interface A of $\mathbf{C}$, it is forwarded to the game. The resulting challenge e is output as $(\mathsf{msg}, 1, c)$ at interface E. When (inj, e') with $e' \neq e$ is input at interface E, $\mathbf{C}$ passes e' to the game's decryption oracle. If the answer is test, it outputs m at interface B. If the answer is a message $m' \neq \perp$, it is output at B. Clearly,

$$\mathbf{CG}_1^{\mathsf{RCCA}} \equiv \mathbf{V}.$$

Moreover, for any $\mathbf{D}$, $\Delta^{\mathbf{D}}(\mathbf{CG}_0^{\mathsf{RCCA}}, \mathbf{U}) \leq n/|\mathcal{M}|$, since the two systems behave identically until $\mathbf{D}$ inputs (inj, e') for an e' that decrypts to m_1 (chosen by $\mathbf{G}_0^{\mathsf{RCCA}}$) at the E-interface. Therefore,

$$\begin{aligned}\Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V}) &\leq \Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{CG}_0^{\mathsf{RCCA}}) + \Delta^{\mathbf{D}}(\mathbf{CG}_0^{\mathsf{RCCA}}, \mathbf{CG}_1^{\mathsf{RCCA}}) \\ &\leq \frac{n}{|\mathcal{M}|} + \Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{RCCA}}, \mathbf{G}_1^{\mathsf{RCCA}}),\end{aligned}$$

and the claim follows. □

3.2.4 Variants of CCA Security

Bellare *et al.* [BHK09] analyze several ways of enforcing the condition that the adversary must not query the challenge ciphertext e to the decryption oracle. They consider modifications along two axes: First, the condition can be enforced during the entire game (b for *both* phases) or only in the second phase (s for *second* phase), i.e., after e has been given to the adversary. Second, one can either exclude adversaries with a non-zero probability of violating the condition from consideration (e for *exclusion*) or penalize an adversary (by declaring the game lost) whenever he asks the challenge e (p for *penalty*). The combination of these choices yields four *non-equivalent* notions IND-CCA-sp, IND-CCA-se, IND-CCA-bp, IND-CCA-be. The s-notions are equivalent to each other and to the formulation of CCA-security in this work (cf. Section 2.8). The e-notions are strictly weaker and do in fact not even imply CCA1-security [BHK09]. Since CCA1-security is weaker than RCCA-security and RCCA is needed for construction (3.2), they are not sufficient for (3.2).

3.2.5 CCA1 Security

The notion of IND-CCA1-security is defined via a corresponding game $\mathbf{G}^{\mathsf{CCA1}}$, which works as $\mathbf{G}^{\mathsf{CCA}}$ except that no decryption queries are answered once the adversary has been given the challenge ciphertext. The most natural way to translate this into a constructive statement is to consider the construction of a (type of) confidential channel $\mathsf{CONF\text{-}STOP}_{AB}$ where the adversary can inject messages at interface E only as long as no message has been input at A from an insecure channel $\mathsf{INSEC\text{-}STOP}_{AB}$ with the same property.

Let $\langle\mathsf{INSEC\text{-}STOP}_{AB}\rangle_n$ and $\langle\mathsf{CONF\text{-}STOP}_{AB}\rangle_n$ denote the insecure resp. confidential channels processing only the first n messages at interface A. Theorem 3.8, whose proof is omitted since it is very similar to the proof of Theorem 3.2, implies that protocol $\mathsf{pke} = (\mathsf{enc}, \mathsf{dec})$ built from a CCA1-secure PKE scheme Π as in Section 3.1.1 achieves

$$[\mathsf{1\text{-}AUTH}_{BA}, \mathsf{INSEC\text{-}STOP}_{AB}] \quad \stackrel{\mathsf{pke}}{\Longrightarrow} \quad \mathsf{CONF\text{-}STOP}_{AB}. \tag{3.5}$$

Theorem 3.8. *There exists a simulator σ and for any $n \in \mathbb{N}$ there exists a (efficient) reduction $\mathbf{C}$ such that for every $\mathbf{D}$,*

$$\Delta^{\mathbf{D}}(\mathsf{enc}^A\mathsf{dec}^B[\mathsf{1\text{-}AUTH}_{BA}, \langle\mathsf{INSEC\text{-}STOP}_{AB}\rangle_n], \sigma^E\langle\mathsf{CONF\text{-}STOP}_{AB}\rangle_n) \\ \leq n \cdot \Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{CCA1}}, \mathbf{G}_1^{\mathsf{CCA1}}).$$

Although this construction seems somewhat artificial, it can be used in any setting where the assumed channel is an appropriate modeling of an available physical channel (or can itself be constructed from such a channel).

3.3 Capturing Settings with Multiple Senders

Although in this work the security definitions for public-key encryption are phrased in a setting where there is only one legitimate sender (at the A-interface), the treatment captures the setting with multiple senders. What is needed to formalize this more general case explicitly is a lifting of the setting with interfaces A, B, and E into the multi-party setting with many senders. In the case where all senders in the multi-sender setting faithfully follow the protocol, this lifting simply relates all those sender interfaces to the single sender interface in the setting with three interfaces.

In a scenario with multiple senders, it is important to formulate the guarantees that are maintained if one or more of the senders deviate from

the protocol because their machines are controlled by some attacker. This is captured in most security frameworks by considering an external adversary that has the capability of corrupting some of the parties. In the context of PKE and secure communication, the goal is to still provide confidentiality guarantees to non-corrupted senders. (If the receiver is corrupted, then no security can be guaranteed.)

The ability of an attacker to act on behalf of corrupted senders means that it can directly send (potentially bogus) ciphertexts to the receiver, even if the communication to the receiver is authenticated. This capability corresponds exactly to the case of assuming only an unauthenticated channel, where the messages are injected via the E-interface. Hence, our treatment extends to the case of (static) sender corruption by considering the lifting that relates the interfaces of the senders in the multi-party scenario to the A-interface in the three-party setting, and provides the capabilities of the statically corrupted parties also at the E-interface. The lifting mappings described above are generic for constructive cryptography and not specific to public-key encryption, and hence formalizing them is not in the scope of the current paper.

In summary, the security of public-key encryption in the presence of potentially (statically) corrupted senders corresponds exactly to the construction of a confidential channel CONF_{AB} from one insecure channel INSEC_{AB} and one authenticated channel $\mathsf{1\text{-}AUTH}_{BA}$ in the opposite direction, as discussed in Section 3.1.3. This implies that in the presence of (static) corruption, IND-RCCA security is required and sufficient both in the case where the channel from the sender to the receiver is authenticated and also where it is not authenticated.

3.4 Idealized Algorithms vs. Resources

The security guarantees that one requires from a cryptographic scheme can be modeled in fundamentally different ways, even within a single formal security framework. One approach, which underlies the public-key encryption functionality $\mathcal{F}_{\text{PKE}}$ in [CKN03], is to idealize the properties of the algorithms that comprise the scheme. Such a functionality corresponds to a cryptographic scheme, and its interfaces closely resemble the interfaces of the algorithms (although, e.g., the private key is never output by $\mathcal{F}_{\text{PKE}}$). In such a treatment, elements that are essential for using the scheme, such as the ciphertext or the public key, will still appear in the functionality, but they are idealized in that, e.g., the ciphertext is independent of the corresponding plaintext; the idealized scheme is unbreakable by definition.

Another—fundamentally different—approach is to explicitly model *resources* that are available to one or more parties. The communication channels described in Section 2.7 can be considered *network resources*; there are also functionalities in the UC framework, such as $\mathcal{F}_{\text{AUTH}}$ or $\mathcal{F}_{\text{SC}}$ in [CK02], that can be interpreted in this way. More generally, one can also think of randomness, memory, or even computation as resources of this type. Following the constructive paradigm, the guarantees of a cryptographic scheme are *not* a resource, but modeled as the guarantee that the scheme transforms one (assumed) resource into another (constructed) resource.[5] Compared to ideal functionalities of the above type, the description of resources tends to be simpler and easier to understand. For example, in the case of public-key encryption, the confidential channel does not need to specify implementation artifacts such as ciphertexts or public keys.

While both approaches allow to divide the security proof of a composite protocol into several separate steps that can be proven independently, only the second approach enables a fully modular protocol design. Each subprotocol achieves a well-defined construction step transforming a resource R into a resource S, which abstracts from how S is achieved. A higher-level protocol can thus use such a resource S independently of how it is obtained, and the construction of S can be replaced with a different one without affecting the design or proof of the higher-level protocol. Concretely, a protocol using the resource CONF_{AB} does not depend on whether or not the channel is constructed by a public-key encryption scheme, whereas a protocol using the functionality $\mathcal{F}_{\text{PKE}}$ will always be specific to this step.

[5] By contrast, a typical UC security statement is that a cryptographic scheme implements some functionality. While statements about *hybrid* protocols in UC appear similar to constructive statements, they are less expressive since, e.g., the UC framework technically does not allow to make statements about assuming only *bounded* resources, as protocols that use hybrid functionalities can always instantiate arbitrarily many functionalities of a given type.

Chapter 4

Self-Destruct Attacks

The strongest security level for public-key encryption (PKE) is *indistinguishability under chosen-ciphertext attacks* IND-CCA, where the adversary is given unrestricted adaptive access to a decryption oracle (modulo not being able to ask the "challenge ciphertext"); this notion is sufficient for most natural applications of PKE.

Despite numerous efforts, it remains unknown whether IND-CCA-secure PKE schemes can be generically built from IND-CPA-secure PKE. This motivates the study of various "middle-ground" security notions between IND-CPA and IND-CCA, which are sufficient for applications, and, yet, can be constructed from the more basic IND-CPA security notion. The study of domain extension for PKE via non-malleable codes (cf. Chapter 5) lead to the discovery of two such security notions:

- Indistinguishability under *self-destruct attacks* (IND-SDA): the attacker gets access to an (adaptive) decryption oracle that only answers decryption queries up to the first invalid ciphertext submitted.

- *Non-malleability* under *self-destruct attacks* (NM-SDA): The attacker gets to adaptively ask many "parallel" decryption queries (i.e., a query consists of many ciphertexts) up to the point when the first invalid ciphertext is submitted. In such a case, the whole parallel decryption query containing the invalid ciphertext is still answered in full, but no future decryption queries are allowed.

The notion of IND-SDA turns up when combining single-bit PKE with non-malleable codes, and NM-SDA is a natural generalization of both IND-SDA and the standard notion NM-CPA, which can be seen as security against

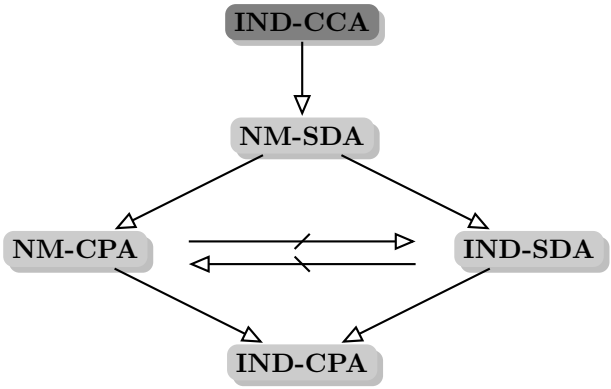

Figure 4.1: *Diagram of the relationships between the new security notions considered in this chapter. $X \to Y$ means that X implies Y; $X \not\to Y$ indicates a separation between X and Y. Notions with the same shades are equivalent under black-box transformations; notions with different shades are not known to be equivalent.*

an attacker that asks but a single parallel decryption query. Section 4.1 contains the definitions of all three notions.

As shown in Section 4.2, IND-SDA and NM-CPA are *incomparable* notions in that there are schemes that satisfy the former but not the latter and vice versa. Hence, NM-SDA, being a generalization of both IND-SDA and NM-CPA, is strictly stronger than either notion (cf. Figure 4.1).

Section 4.3 provides constructive semantics for the new notions (in the spirit of Section 3.2).

Finally, Section 4.4 shows how to generically transform a IND-CPA-secure PKE scheme into a NM-SDA-secure one.

4.1 Definitions of the New Notions

The new self-destruct-attack notions are formalized using the distinguishing game $\mathbf{G}^{\mathsf{pke\text{-}sda}}_{q,p,b}$, depicted in Figure 4.2: The distinguisher (adversary) is initially given a public key and then specifies a message m_0. This message or a random message m_1 of the same length is encrypted and the adversary is given the resulting challenge ciphertext. During the entire game, the distinguisher has access to a decryption oracle that allows him to make at most q decryption queries, each consisting of at most p ciphertexts. Once the distinguisher specifies an invalid ciphertext, the decryption oracle self-

Distinguishing Game $\mathbf{G}^{\mathsf{pke\text{-}sda}}_{q,p,b}$

init
  $\mathsf{ctr} \leftarrow 0$
  $(\mathsf{pk}, \mathsf{sk}) \leftarrow K$
  output pk

on (chall, m_0)
  $m_1 \leftarrow \mathcal{M}$ s.t. $|m_0| = |m_1|$
  $e \leftarrow E_{\mathsf{pk}}(m_b)$
  output e

on $(\mathsf{dec}, e^{(1)}, \ldots, e^{(p)})$
  $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$
  for $j \leftarrow 1$ **to** p
    if $\mathsf{valid}(e^{(j)})$
      $m^{(j)} \leftarrow D_{\mathsf{sk}}(e^{(j)})$
    else
      $m^{(j)} \leftarrow \mathsf{test}$
  output $(m^{(1)}, \ldots, m^{(p)})$
  if $\exists j : m^{(j)} = \perp$ *or* $\mathsf{ctr} \geq q$
    self-destruct

Figure 4.2: *Distinguishing game used to define the self-destruct-attack security notions for PKE schemes* $\Pi = (K, E, D)$*. The numbers* $q, p \in \mathbb{N}$ *specify the maximum number of decryption queries and their size, respectively. The command* **self-destruct** *results in all future decryption queries being ignored.*

destructs, i.e., no additional decryption queries are answered.

Define $\mathsf{valid}(e') = 1$ for all e' before the challenge is output and change to $\mathsf{valid}(e) = 0$ after the challenge e is output.

Depending on the values of q and p, one obtains the three notions NM-CPA, IND-SDA, and NM-SDA:

- *Non-malleability* (NM-CPA): The adversary can make a single decryption query consisting of arbitrarily many ciphertexts, i.e., $q = 1$ and p arbitrary (denoted by $p = *$). For readability, set $\mathbf{G}^{\mathsf{NM\text{-}CPA}}_b := \mathbf{G}^{\mathsf{pke\text{-}sda}}_{1,*,b}$ for $b \in \{0, 1\}$.[1]

- *Indistinguishability under self-destruct attacks* (IND-SDA): The adversary can make arbitrarily many decryption queries, but each may consist of a single ciphertext only, i.e., q arbitrary (denoted by $q = *$) and $p = 1$. For readability, set $\mathbf{G}^{\mathsf{IND\text{-}SDA}}_b := \mathbf{G}^{\mathsf{pke\text{-}sda}}_{*,1,b}$ for $b \in \{0, 1\}$.

- *Non-malleability under self-destruct attacks* (NM-SDA): The adversary can make arbitrarily many decryption queries, each consisting of arbitrarily many ciphertexts, i.e., q and p arbitrary (denoted by $q = p = *$). For readability, set $\mathbf{G}^{\mathsf{NM\text{-}SDA}}_b := \mathbf{G}^{\mathsf{pke\text{-}sda}}_{*,*,b}$ for $b \in \{0, 1\}$.

[1] Note that the way NM-CPA is defined here is slightly stronger than the normal notion. This is due to the adversary's ability to ask a parallel decryption query at any time—as opposed to only after receiving the challenge ciphertext in earlier definitions (cf., e.g., [PSV06]).

Definition 4.1. *Let* $\mathsf{SN} \in \{\mathsf{NM\text{-}CPA}, \mathsf{IND\text{-}SDA}, \mathsf{NM\text{-}SDA}\}$*,* $t \in \mathbb{N}$ *and* $\varepsilon \geq 0$*. A PKE scheme* $\Pi = (K, E, D)$ *is* (t, ε)*-*SN*-secure if*

$$\Delta^{\mathbf{D}}(\mathbf{G}_0^{\mathsf{SN}}, \mathbf{G}_1^{\mathsf{SN}}) \leq \varepsilon$$

for all distinguishers $\mathbf{D}$ *running in time at most* t*.*

Remark. Note that one or both of the parameters $q, p \in \mathbb{N}$ are sometimes made explicit, resulting in the definition of, e.g., a (t, q, p, ε)-NM-SDA-secure scheme. Correspondingly, one considers, e.g., the game $\mathbf{G}_{q,p,b}^{\Pi,\mathsf{NM\text{-}SDA}}$ (making the PKE Π in consideration explicit as well).

4.2 Separating IND-SDA and NM-CPA

The notions of NM-CPA and IND-SDA security are incomparable, as shown in this section. That is, there are schemes that are NM-CPA-secure but not IND-SDA-secure and vice versa.

4.2.1 NM-CPA Does Not Imply IND-SDA

The modified scheme. Let λ be the security parameter and $\Pi = (K, E, D)$ be a NM-CPA-secure PKE scheme with message space $\mathcal{M} = \{0,1\}^\lambda$. Consider the following modification $\Pi' = (K', E', D')$ of Π (cf. Figure 4.3):

- The key generation algorithm K' works as K but additionally samples a uniformly random message $\rho \leftarrow \mathcal{M}$, which becomes part of the secret key.
- The encryption algorithm E' works as E except that it prepends a zero-bit to all ciphertexts.
- The decryption algorithm D' proceeds as follows upon receiving a ciphertext $e' = \beta \| e$: If $\beta = 1$, it outputs ρ. If $\beta = 0$, it decrypts $m \leftarrow D_{\mathsf{sk}}(e)$. If $m = \rho$, the decryption algorithm outputs the secret key, and otherwise m.

Security of the modified scheme. PKE scheme Π' clearly is not IND-SDA-secure: A distinguisher simply queries $1 \| E_{\mathsf{pk}}(m)$ for some message m to obtain message ρ. By subsequently querying $0 \| E_{\mathsf{pk}}(\rho)$, the distinguisher obtains the secret key.

PKE Scheme $\Pi' = (K', E', D')$

Key Generation K'
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow K$
- $\rho \leftarrow \mathcal{M}$
- $\mathsf{pk}' \leftarrow \mathsf{pk}$
- $\mathsf{sk}' \leftarrow (\rho, \mathsf{sk})$
- **return** $(\mathsf{pk}', \mathsf{sk}')$

Encryption $E'_{\mathsf{pk}'}(m)$
- $e \leftarrow E_{\mathsf{pk}}(m)$
- $e' \leftarrow 0 \,\|\, e$
- **return** e'

Decryption $D'_{\mathsf{sk}'}(e')$
- $\beta \,\|\, e \leftarrow e'$
- $m \leftarrow D_{\mathsf{sk}}(e)$
- **if** $\beta = 1$
  - **return** ρ
- **else**
  - **if** $m = \rho$
    - **return** sk
  - **else**
    - **return** m

Figure 4.3: *PKE scheme* Π' *based on an* NM-CPA*-secure PKE scheme* Π*.*

The modified scheme is, however, still NM-CPA-secure as implied by the following lemma:

Lemma 4.1. *For all* $p \in \mathbb{N}$*, there exists a reduction* $\mathbf{C}$ *such that for all distinguishers* $\mathbf{D}$*,*

$$\Delta^{\mathbf{D}}(\mathbf{G}^{\Pi',\mathsf{NM\text{-}CPA}}_{1,p,0}, \mathbf{G}^{\Pi',\mathsf{NM\text{-}CPA}}_{1,p,1}) \leq \Delta^{\mathbf{DC}}(\mathbf{G}^{\Pi,\mathsf{NM\text{-}CPA}}_{1,p,0}, \mathbf{G}^{\Pi,\mathsf{NM\text{-}CPA}}_{1,p,1}) + 2 \cdot p2^{-\lambda}.$$

Proof. Reduction $\mathbf{C}$ works as follows: Initially, it chooses $\rho \leftarrow \mathcal{M}$ uniformly at random and forwards the public key received on the inside to the outside. Upon receiving (chall, m_0) on the outside, $\mathbf{C}$ forwards it to the inside, which results in a ciphertext e^* being received on the inside. The value $0\|e^*$ is then output by $\mathbf{C}$ on the outside.

Moreover, $\mathbf{C}$ answers each component e' of the parallel decryption query received at the outside as follows: It first parses e' as $\beta\|e$. Then, if $\beta = 1$, the answer to the query is ρ. Otherwise, $\mathbf{C}$ uses its own decryption oracle on the inside to decrypt e and answers the query by the answer m.[2]

It is easily seen that for $b \in \{0, 1\}$,

$$\mathbf{G}^{\Pi',\mathsf{NM\text{-}CPA}}_{1,p,b} \quad \text{and} \quad \mathbf{CG}^{\Pi,\mathsf{NM\text{-}CPA}}_{1,p,b}$$

differ only if $\mathbf{D}$ asks a ciphertext that decrypts to ρ. This event occurs with probability $p \cdot |\mathcal{M}| = p \cdot 2^{-\lambda}$. The lemma now follows using a simple triangle inequality. □

[2] Of course, $\mathbf{C}$ actually asks a single parallel query with the ciphertexts e for all components.

PKE Scheme $\Pi'' = (K'', E'', D'')$

Key Generation K''
$(\mathsf{pk}'', \mathsf{sk}'') \leftarrow K$
return $(\mathsf{pk}'', \mathsf{sk}'')$

Encryption $E''_{\mathsf{pk}''}(m)$
$c \leftarrow \mathrm{Enc}(m)$
$e \leftarrow E_{\mathsf{pk}}(c)$
$e'' \leftarrow 0 \parallel 0 \parallel 0^{\nu} \parallel e$
return e''

Decryption $D''_{\mathsf{sk}''}(e'')$
$\beta \parallel d \parallel i \parallel e \leftarrow e''$
$c \leftarrow D_{\mathsf{sk}}(e)$
$m \leftarrow \mathrm{Dec}(c)$
if $\beta_1 = 0$
  if $(\beta_2 = 0) \wedge (i = 0^{\nu})$
    return m
  else
    return $\perp$
else
  if $(c[i] = d)$
    return 0^k
  else
    return $\perp$

Figure 4.4: *PKE scheme* Π'' *based on an* IND-SDA*-secure PKE scheme* Π.

4.2.2 IND-SDA Does Not Imply NM-CPA

The modified scheme. Let λ be the security parameter and $\Pi = (K, E, D)$ be a IND-SDA-secure PKE scheme with message space $\mathcal{M} = \{0,1\}^{\lambda}$. Moreover, let $(\mathrm{Enc}, \mathrm{Dec})$ be a (k, λ)-coding scheme with $\tau\lambda$-secrecy (as defined for LEDSS in Section 2.9) for some constant $\tau > 0$ and some $k > 0$. Consider the following modification $\Pi'' = (K'', E'', D'')$ of Π (cf. Figure 4.4):

- The key generation algorithm K'' is the same as K.
- The encryption algorithm E'' works as follows: To encrypt a message $m \in \{0,1\}^k$, it computes $c \leftarrow \mathrm{Enc}(m)$ and $e \leftarrow E_{\mathsf{pk}}(c)$ and outputs $e'' \leftarrow 0 \| 0 \| 0^{\nu} \| e$, where $\nu := \lceil \log \lambda \rceil$.
- The decryption algorithm D'' proceeds as follows upon receiving a ciphertext $e'' = \beta \| d \| i \| e$: If $\beta = 0$, $d = 0$, and $i = 0^{\nu}$, it decrypts $c \leftarrow D_{\mathsf{sk}}(e)$, computes $m \leftarrow \mathrm{Dec}(c)$, and outputs m. If $\beta = 1$ and $c[i] = d$ (i.e., if d is a correct guess for the i^{th} bit of the encoding), D'' outputs 0^k. In all other cases, it outputs $\perp$.[3]

[3]Note that in general, not all ν-bit strings i are valid indices. If the decryption algorithm encounters an invalid index, it also outputs $\perp$. For readability this issue is ignored in the remainder of this section.

Security of the modified scheme. PKE scheme Π'' is not NM-CPA-secure: A distinguisher can recover each bit $i \in [n]$ of the encoding c^* encrypted in the challenge ciphertext $0\|0\|0^\nu\|e^*$ by via a single parallel query containing ciphertexts

$$e^{(i)} := 1 \parallel 0 \parallel i \parallel e^*.$$

If the answer to the i^{th} query is 0^k, then $c^*[i] = 0$; otherwise $c^*[i] = 1$. Computing $\mathrm{Dec}(c^*)$ yields the plaintext encrypted by the challenge.

The modified scheme is, however, still IND-SDA-secure as implied by the following lemma:

Lemma 4.2. *For all $q \in \mathbb{N}$, there exist reductions $\mathbf{C}_0$ and $\mathbf{C}_1$ such that for all distinguishers $\mathbf{D}$,*

$$\begin{aligned}\Delta^{\mathbf{D}}(\mathbf{G}_0^{\Pi'',\mathsf{IND\text{-}SDA}}, \mathbf{G}_1^{\Pi'',\mathsf{IND\text{-}SDA}}) \\ \leq \Delta^{\mathbf{DC}_0}(\mathbf{G}_0^{\Pi,\mathsf{IND\text{-}SDA}}, \mathbf{G}_1^{\Pi,\mathsf{IND\text{-}SDA}}) + \Delta^{\mathbf{DC}_1}(\mathbf{G}_0^{\Pi,\mathsf{IND\text{-}SDA}}, \mathbf{G}_1^{\Pi,\mathsf{IND\text{-}SDA}}) \\ + 2^{-\tau\lambda}.\end{aligned}$$

Proof. Let $b \in \{0,1\}$ and consider the hybrid system $\mathbf{H}_b$ that works exactly as $\mathbf{G}_b^{\Pi,\mathsf{IND\text{-}SDA}}$ except that

- the ciphertext e^* in the challenge ciphertext $0\|0\|0^\nu\|e^*$ is computed as the encryption of a random λ-bit string (instead of an encoding of m_b), and
- decryption queries of the form $1\|d\|i\|e^*$ are answered based on an internally generated encoding $c_b = \mathrm{Enc}(m_b)$, i.e., the answer is 0^k if $c_b[i] = d$ and $\perp$ otherwise.

Let $\mathbf{C}_b$ be the following reduction: Initially, it obtains a public key pk at the inside interface, which it forwards to the outside interface. When (chall, m_0) is received on the outside, the reduction chooses a random message m_1 and computes $c_b \leftarrow \mathrm{Enc}(m_b)$ and outputs (chall, c_b) on the inside. Subsequently, it obtains a ciphertext e^* and outputs $0\|0\|0^\nu\|e^*$ on the outside. $\mathbf{C}_b$ answers decryption queries $\beta\|d\|i\|e$ as follows (implementing the self-destruct mode if the answer is $\perp$):

- If $\beta = 0$, $d = 0$, and $i = 0^\nu$, $\mathbf{C}_b$ proceeds as follows: If $e = e^*$, the answer to the query is test. Otherwise, it outputs (dec, e) at the inside interface. The value c subsequently received at the inside interface is decoded to $m \leftarrow \mathrm{Dec}(c)$ and output at the outside interface. If $\beta = 0$ but $d \neq 0$ or $i \neq 0^\nu$, $\mathbf{C}_b$ responds with $\perp$.

- If $\beta = 1$ and $e = e^*$, $\mathbf{C}_b$ outputs 0^k if $c_b[i] = d$ and $\perp$ otherwise.
- If $\beta = 1$ and $e \neq e^*$, $\mathbf{C}_b$ outputs (dec, e) at the inside interface and subsequently obtains a value c at the inside interface. $\mathbf{C}_b$ outputs 0^k if $c[i] = d$ and $\perp$ otherwise.

By inspection, one verifies that for $b \in \{0, 1\}$

$$\mathbf{C}_b \mathbf{G}_0^{\Pi,\mathsf{IND\text{-}SDA}} \equiv \mathbf{G}_b^{\Pi'',\mathsf{IND\text{-}SDA}}$$

and that

$$\mathbf{C}_b \mathbf{G}_1^{\Pi,\mathsf{IND\text{-}SDA}} \equiv \mathbf{H}_b.$$

Moreover, note that $\Delta^{\mathbf{D}}(\mathbf{H}_0, \mathbf{H}_1) \leq 2^{-\tau\lambda}$ due to the $\tau\lambda$-secrecy of the coding scheme $(\mathrm{Enc}, \mathrm{Dec})$ and the fact that the hybrids $\mathbf{H}_b$ do not leak any information about c_b other than by answering decryption queries with $\beta = 1$ and $e = e^*$.

The lemma now follows using a simple triangle inequality. □

4.3 Constructive Semantics of the New Notions

IND-SDA security. Consider a PKE scheme Π and the protocol $\mathsf{pke} = (\mathsf{enc}, \mathsf{dec})$ based on it as shown in Section 3.1.1. Consider the following minor modification pke' of pke: dec, after receiving the first invalid ciphertext e' (i.e., $D_{\mathsf{sk}}(e') = \perp$), stops decrypting. It can be shown along the lines of the proof of Theorem 3.2 that pke' also achieves transformation (3.2).

NM-SDA security. The notions of NM-CPA and NM-SDA provide security against attackers with access to decryption oracles that answer *parallel* queries. As shown below, both notions, albeit weaker than full IND-CCA-security, suffice for the scenario of a *blind auction*, where participants submit encrypted bits. It turns out that NM-CPA schemes can be used for a single auction, whereas NM-SDA schemes can be used as long as no participant submits an invalid ciphertext.

To model the auction, consider the following channel AUC_{AB}: It internally keeps an initially empty list $\mathcal{L}$ of messages. When the i^{th} message m is input at interface A, it is recorded as (i, m) and $(i, |m|)$ is output at interface E. When (dlv, i') is input at interface E and if (i', m') has been recorded, m' is appended to $\mathcal{L}$. When (inj, m') is input at interface E, m' is appended to $\mathcal{L}$. When $\mathsf{dlv\text{-}all}$ is input at B, all messages in $\mathcal{L}$ are output at B, and $\mathcal{L}$ is flushed.

Consider the following protocol $\mathsf{pke}'' = (\mathsf{enc}'', \mathsf{dec}'')$, built based on Π as $\mathsf{pke} = (\mathsf{enc}, \mathsf{dec})$ in Section 3.1.1, except that dec'' only outputs the messages it received once $\mathsf{dlv\text{-}all}$ is input at the outside interface; if $\perp$ is among these messages, the converter halts. Theorem 4.3 below implies that pke'' achieves construction

$$[\mathsf{1\text{-}AUTH}_{BA}, \mathsf{INSEC}_{AB}] \quad \overset{\mathsf{pke}''}{\longmapsto} \quad \mathsf{AUC}_{AB}. \tag{4.1}$$

if Π is $\mathsf{NM\text{-}SDA}$-secure.

Theorem 4.3. *There exists a simulator σ and for any $n \in \mathbb{N}$ there exists a (efficient) reduction $\mathbf{C}$ such that for every $\mathbf{D}$,*

$$\Delta^{\mathbf{D}}(\mathsf{enc}''^{A}\mathsf{dec}''^{B}[\mathsf{1\text{-}AUTH}_{BA}, \langle \mathsf{INSEC}_{AB}\rangle_n], \sigma^{E}\langle \mathsf{AUC}_{AB}\rangle_n) \\ \leq n \cdot \Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{NM\text{-}SDA}}, \mathbf{G}_1^{\mathsf{NM\text{-}SDA}}).$$

Proof. Let σ be the simulator from Theorem 3.2. Consider the two systems

$$\mathsf{enc}''^{A}\mathsf{dec}''^{B}[\mathsf{1\text{-}AUTH}_{BA}, \langle \mathsf{INSEC}_{AB}\rangle_1] \qquad \text{and} \qquad \sigma^{E}\langle \mathsf{AUC}_{AB}\rangle_1.$$

Distinguishing $\mathbf{G}_0^{\mathsf{NM\text{-}SDA}}$ from $\mathbf{G}_1^{\mathsf{NM\text{-}SDA}}$ can be reduced to distinguishing these two systems via the following reduction system $\mathbf{C}'$. Initially, $\mathbf{C}'$ takes pk from the game and outputs it at the E-interface. When a message m is input at interface A of $\mathbf{C}'$, it is forwarded as (chall, m) to the game. The challenge e from the game is output as $(\mathsf{msg}, 1, e)$ at interface E. When (inj, e') is input at interface E, $\mathbf{C}'$ records e' in a list $\mathcal{L}$. When $\mathsf{dlv\text{-}all}$ is input at interface B, $\mathbf{C}'$ passes the vector of all recorded ciphertexts in $\mathcal{L}$ as a decryption query to the game. In the subsequently received vector of plaintexts from the game, it replaces all test-messages by m. Then, it outputs all the plaintexts at B and clears the list $\mathcal{L}$. If any of the plaintexts output are $\perp$, it halts. It holds that

$$\mathbf{C}'\mathbf{G}_0^{\mathsf{NM\text{-}SDA}} \equiv \mathsf{enc}''^{A}\mathsf{dec}''^{B}[\mathsf{1\text{-}AUTH}_{BA}, \langle \mathsf{INSEC}_{AB}\rangle_1]$$

and

$$\mathbf{C}'\mathbf{G}_1^{\mathsf{NM\text{-}SDA}} \equiv \sigma^{E}\langle \mathsf{AUC}_{AB}\rangle_1,$$

and thus

$$\begin{aligned}
&\Delta^{\mathbf{D}}(\mathsf{enc}''^{A}\mathsf{dec}''^{B}[\mathsf{1\text{-}AUTH}_{BA}, \langle \mathsf{INSEC}_{AB}\rangle_n], \sigma^{E}\langle \mathsf{AUC}_{AB}\rangle_n) \\
&\quad \leq n \cdot \Delta^{\mathbf{DC}''}(\mathsf{enc}''^{A}\mathsf{dec}''^{B}[\mathsf{1\text{-}AUTH}_{BA}, \langle \mathsf{INSEC}_{AB}\rangle_1], \sigma^{E}\langle \mathsf{AUC}_{AB}\rangle_1) \\
&\quad = n \cdot \Delta^{\mathbf{DC}''}(\mathbf{C}'\mathbf{G}_0^{\mathsf{NM\text{-}SDA}}, \mathbf{C}'\mathbf{G}_1^{\mathsf{NM\text{-}SDA}}) \\
&\quad = n \cdot \Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{NM\text{-}SDA}}, \mathbf{G}_1^{\mathsf{NM\text{-}SDA}}),
\end{aligned}$$

where $\mathbf{C} := \mathbf{C}''\mathbf{C}'$ and the first inequality follows from a standard hybrid argument for a reduction system $\mathbf{C}''$ (similar to that in the proof of Lemma 3.4). □

NM-CPA security. Based on AUC_{AB}, one can also define a single-auction variant $\mathsf{AUC\text{-}SINGLE}_{AB}$ that shuts down after the first dlv-all command. A straight-forward adaptation of the above protocol pke'' yields a protocol that achieves

$$[\text{1-}\mathsf{AUTH}_{BA}, \mathsf{INSEC}_{AB}] \quad \Longmapsto \quad \mathsf{AUC\text{-}SINGLE}_{AB}. \tag{4.2}$$

4.4 NM-SDA Security from IND-CPA Security

NM-SDA security can be achieved in a black-box fashion from IND-CPA security. Specifically, a generalization using LECSS (cf. Section 2.9) of the scheme by Choi *et al.* [CDMW08] (dubbed the CDMW construction in the remainder of this section) is NM-SDA-secure. Using a constant-rate LECSS allows to improve the rate of the CDMW construction from $\Omega(1/\lambda^2)$ to $\Omega(1/\lambda)$, where λ is the security parameter. This abstraction might also give a deeper understanding of the result of [CDMW08]. The main difficulty in the analysis is to extend the original proof to deal with adaptively chosen parallel decryption queries (with self-destruct).

4.4.1 The CDMW Construction

The CDMW construction uses a randomized Reed-Solomon code, which is captured as a special case by the notion of a linear error-correcting secret sharing (LECSS) $(\mathsf{E}, \mathsf{D}, \mathsf{R})$ (cf. Section 2.9).

The LECSS has to satisfy an additional property, which is that given a certain number of symbols chosen uniformly at random and independently and a plaintext m, one can efficiently produce an encoding that matches the given symbols and has the same distribution as $\mathsf{E}(m)$. It is described in more detail in the proof of Lemma 4.9, where it is needed.[4]

Let $\Pi = (K, E, D)$ be a PKE scheme with message space $\mathcal{M} = \{0,1\}^\ell$ (we assume $\ell = \Omega(\lambda)$), and let $\Sigma = (K^{\mathsf{ots}}, S, V)$ be a one-time signature scheme with verification keys of length $\kappa = \mathcal{O}(\lambda)$. Moreover, let $\alpha > 0$ be any constant and (E, D) a (k, n, δ, τ)-LECSS over $\mathrm{GF}(2^\ell)$ with $\delta > 2\alpha$.

The CDMW construction (cf. Figure 4.5), to encrypt a plaintext $m \in \{0,1\}^{k\ell}$, first computes an encoding $(c_1, \ldots, c_n) \leftarrow \mathsf{E}(m)$ and then creates

[4] Of course, the Reed-Solomon-based LECSS from [CDMW08] has this property.

PKE Scheme $\Pi' = (K', E', D')$

Key Generation K'
  for $(b, i, j) \in \{0,1\} \times [\kappa] \times [n]$
  $(\mathsf{pk}^b_{i,j}, \mathsf{sk}^b_{i,j}) \leftarrow K$
  $\mathsf{PK} \leftarrow (\mathsf{pk}^b_{i,j})_{b,i,j}$
  $\mathsf{SK} \leftarrow (\mathsf{sk}^b_{i,j})_{b,i,j}$
  $T \leftarrow \binom{[n]}{\tau n}$
  return $(\mathsf{PK}, (\mathsf{SK}, T))$

Encryption $E'_{\mathsf{PK}}(m)$
  $(c_1, \ldots, c_n) \leftarrow \mathsf{E}(m)$
  $(\mathsf{verk}, \mathsf{sigk}) \leftarrow K^{\mathsf{ots}}$
  $(v[1], \ldots, v[\kappa]) \leftarrow \mathsf{verk}$
  for $(i, j) \in [\kappa] \times [n]$
  $e_{i,j} \leftarrow E_{\mathsf{pk}^{v[i]}_{i,j}}(c_j)$
  $\mathbf{E} \leftarrow (e_{i,j})_{i,j}$
  $\sigma \leftarrow S_{\mathsf{sigk}}(\mathbf{E})$
  return $(\mathbf{E}, \mathsf{verk}, \sigma)$

Decryption $D'_{(\mathsf{SK},T)}(\mathbf{E}, \mathsf{verk}, \sigma)$
  if $V_{\mathsf{verk}}(\mathbf{E}, \sigma) = 0$
  return $\perp$
  for $j \in T$
  decrypt j^{th} column of $\mathbf{E}$
  if *not all entries identical*
  return $\perp$
  decrypt first row of $\mathbf{E}$ to c
  $w \leftarrow \mathsf{R}(c, \alpha n)$
  if $w = \perp$ *or* $\exists j \in T : c_j \neq w_j$
  return $\perp$
  return $\mathsf{D}(w)$

Figure 4.5: *The CDMW PKE scheme* Π' *constructed from a CPA-secure scheme* Π *[CDMW08].*

the $(\kappa \times n)$-matrix $\mathbf{C}$ in which this encoding is repeated in every row. For every entry $\mathbf{C}_{ij}$ of this matrix, there are two possible public keys $\mathsf{pk}^b_{i,j}$; which of them is used to encrypt the entry is determined by the i^{th} bit $v[i]$ of the verification key $\mathsf{verk} = (v[1], \ldots, v[\kappa])$ of a freshly generated key pair for Σ. In the end, the encrypted matrix $\mathbf{E}$ is signed using verk, producing a signature σ. The ciphertext is $(\mathbf{E}, \mathsf{verk}, \sigma)$.

The decryption algorithm first verifies the signature. Then, it decrypts all columns indexed by a set $T \subset [n]$, chosen as part of the secret key, and checks that each column consists of a single value only. Finally, it decrypts the first row and tries to find a codeword with relative distance at most α. If so, it checks whether the codeword matches the first row in the positions indexed by T. If all checks pass, it outputs the plaintext corresponding to the codeword; otherwise it outputs $\perp$.

Theorem 4.4. *Let* $t \in \mathbb{N}$ *and* Π *be a* $(t + t_{\mathsf{cpa}}, \varepsilon_{\mathsf{cpa}})$-IND-CPA*-secure PKE scheme,* $\alpha > 0$, (E, D) *a* (k, n, δ, τ)*-LECSS with* $\delta > 2\alpha$*, and* Σ *a* $(t +$

$t_{\mathsf{ots}}, \varepsilon_{\mathsf{ots}})$*-secure OTS scheme with verification-key length* κ*. Then, for any* $q, p \in \mathbb{N}$*, PKE scheme* Π' *is* (t, q, p, ε)-NM-SDA*-secure with*

$$\varepsilon \;=\; (1-\tau)\kappa n \cdot \varepsilon_{\mathsf{cpa}} + 2 \cdot \varepsilon_{\mathsf{ots}} + 4 \cdot p(1-\tau)^{\alpha n},$$

where t_{cpa} *and* t_{ots} *represent the overhead incurred by corresponding reductions.*

Instantiating the construction. Note that the security proof below does not use the linearity of the LECSS. The CDMW construction can be seen as using a Reed-Solomon-based LECSS with rate $\mathcal{O}(1/\kappa)$. If the construction is instantiated with a constant-rate LECSS, the final rate improves over CDMW by a factor of $\Omega(\kappa) = \Omega(\lambda)$. More concretely, assuming a constant-rate CPA encryption, a ciphertext of length $\mathcal{O}(\lambda^3)$ can encrypt a plaintext of length $\Omega(\lambda^2)$ as compared to $\Omega(\lambda)$ for plain CDMW. As shown in Section 4.4.3, the LECSS can be instantiated with constructions based on Reed-Solomon or algebraic geometric codes (which also satisfy the additional property mentioned above), both with constant rate. Among the constant-rate codes, algebraic geometric codes allow to choose the parameters optimally also for shorter plaintexts.

4.4.2 Security Proof of the CDMW Construction

Overview

In the following, let $\mathbf{G}_b^{\mathsf{CPA}}$ be the IND-CPA game for the underlying PKE scheme Π, $\mathbf{G}_b^{\mathsf{NM\text{-}SDA}}$ the game for the constructed PKE scheme Π', and $\mathbf{G}^{\mathsf{ots}}$ be the security game for the OTS scheme Σ.

The proof follows the original one by [CDMW08]. The main change is that one needs to argue that, unless they contain invalid ciphertexts, adaptively chosen parallel queries do not allow the attacker to obtain useful information, in particular on the secret set T. This is facilitated by using the *self-destruct lemma* (cf. Section 7). The proof proceeds in three steps using two hybrid games $\mathbf{H}_b$ and $\mathbf{H}'_b$:

- The first hybrid $\mathbf{H}_b$ gets rid of signature forgeries for the verification key used to create the challenge ciphertext. The indistinguishability of the hybrid from $\mathbf{G}_b^{\mathsf{NM\text{-}SDA}}$ follows from the security of the OTS scheme and requires only minor modifications compared to the original proof.

- The second hybrid $\mathbf{H}'_b$ uses an alternative decryption algorithm. The indistinguishability of $\mathbf{H}'_b$ and $\mathbf{H}_b$ holds unconditionally; this step requires new techniques compared to the original proof.

- Finally, the distinguishing advantage between $\mathbf{H}'_0$ and $\mathbf{H}'_1$ is bounded by a reduction to the IND-CPA security of the underlying scheme Π; the reduction again resembles the corresponding one in [CDMW08].

Dealing with Forgeries

For $b \in \{0,1\}$, hybrid $\mathbf{H}_b$ behaves as $\mathbf{G}_b^{\mathsf{NM\text{-}SDA}}$ but generates the signature key pair $(\mathsf{sigk}^*, \mathsf{verk}^*)$ used for the challenge ciphertext initially and rejects any decryption query $(\mathbf{E}', \sigma', \mathsf{verk}')$ if $\mathsf{verk}' = \mathsf{verk}^*$.

Lemma 4.5. *For $b \in \{0,1\}$, there exists a reduction $\mathbf{R}'_b$ such that for all distinguishers $\mathbf{D}$,*

$$\Delta^{\mathbf{D}}(\mathbf{G}_b^{\mathsf{NM\text{-}SDA}}, \mathbf{H}_b) \;\leq\; \Gamma^{\mathbf{DR}'_b}(\mathbf{G}^{\mathsf{ots}}).$$

Proof. $\mathbf{R}'_b$ is a standard reduction to the unforgeability of Σ. □

Alternative Decryption Algorithm

For $b \in \{0,1\}$, hybrid $\mathbf{H}'_b$ behaves as $\mathbf{H}_b$ but for the way it answers decryption queries $(\mathbf{E}', \sigma', \mathsf{verk}')$: As before, it first verifies the signature σ' and checks that each column of $\mathbf{E}'$ consists of encryptions of a single value. Then, it determines the first position i at which verk' and verk^* differ, i.e., where $v'[i] \neq v^*[i]$. It decrypts the i^{th} row of $\mathbf{E}$ and checks if there is a codeword w within distance $2\alpha n$.[5] If such w does not exist or else if w does not match the *first* row in a position indexed by T, the check fails. Otherwise, the plaintext corresponding to w is output.

Lemma 4.6. *For $b \in \{0,1\}$ and all distinguishers $\mathbf{D}$,*

$$\Delta^{\mathbf{D}}(\mathbf{H}_b, \mathbf{H}'_b) \;\leq\; 2 \cdot p(1-\tau)^{\alpha n}.$$

The proof of Lemma 4.6 shows that the original and alternative decryption algorithms are indistinguishable not just for a single parallel query (as is sufficient for NM-CPA) but even against adaptively chosen parallel queries (with self-destruct). It is the main technical contribution of this section.

[5] Recall that the actual decryption algorithm always decrypts the first row and tries to find w within distance αn.

At the core of the proof is an analysis of how different types of encoding matrices $\mathbf{C}$ are handled inside the two decryption algorithms. To that end, one can define two games $\mathbf{B}$ and $\mathbf{B}'$ (below) that capture the behaviors of the original and the alternative decryption algorithms, respectively. The proof is completed by bounding $\Delta(\mathbf{B}, \mathbf{B}')$ (for all distinguishers) and showing the existence of a wrapper $\mathbf{W}_b$ such that $\mathbf{W}_b\mathbf{B}$ behaves as $\mathbf{H}_b$ and $\mathbf{W}_b\mathbf{B}'$ as $\mathbf{H}'_b$ (also below). This proves the lemma since $\Delta^{\mathbf{D}}(\mathbf{H}_b, \mathbf{H}'_b) = \Delta^{\mathbf{D}}(\mathbf{W}_b\mathbf{B}, \mathbf{W}_b\mathbf{B}') = \Delta^{\mathbf{D}\mathbf{W}_b}(\mathbf{B}, \mathbf{B}')$.

The games $\mathbf{B}$ and $\mathbf{B}'$ behave as follows: Both initially choose a random size-τ subset of $[n]$. Then, they accept parallel queries with components of the type $(\mathbf{C}, i)$ for $\mathbf{C} \in \mathbb{F}^{\kappa \times n}$ and $i \in [\kappa]$. The answer to each component is computed as follows:

1. Both games check that all columns indexed by T consist of identical entries.

2. Game $\mathbf{B}$ tries to find a codeword w with distance less than αn from the *first* row (regardless of i), whereas $\mathbf{B}'$ tries to find w within $2\alpha n$ of row i. Then, if such a w is found, *both* games check that it matches the *first* row of $\mathbf{C}$ in the positions indexed by T.

3. If all checks succeed, the answer to the (component) query is w; otherwise, it is $\perp$.

Both games then output the answer vector and implement the self-destruct, i.e., if any of the answers is $\perp$, all *future* queries are answered by $\perp$.

Claim 4.7. *For $b \in \{0, 1\}$ and all distinguishers $\mathbf{D}$,*

$$\Delta^{\mathbf{D}}(\mathbf{B}, \mathbf{B}') \;\leq\; 2 \cdot p(1-\tau)^{\alpha n}.$$

Encoding matrices. Towards a proof of Claim 4.7, consider the following partition of the set of encoding matrices $\mathbf{C}$ (based on the classification in [CDMW08]):

1. There exists a codeword w within αn of the first row of $\mathbf{C}$, and all rows have distance at most αn.

2. (a) There exist two rows in $\mathbf{C}$ with distance greater than αn.

   (b) The rest; in this case the first row differs in more than αn positions from any codeword.

Observe that queries $(\mathbf{C}, i)$ with $\mathbf{C}$ of type 1 are treated identically by both $\mathbf{B}$ and $\mathbf{B}'$: A codeword w within αn of the first row of $\mathbf{C}$ is certainly found by $\mathbf{B}$; since all rows have distance at most αn, w is within $2\alpha n$ of row i and thus also found by $\mathbf{B}'$. Furthermore, note that if $\mathbf{C}$ is of type 2b, it is always rejected by $\mathbf{B}$ (but not necessarily by $\mathbf{B}'$).

Consider the hybrids $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{B}}'$ that behave as $\mathbf{B}$ and $\mathbf{B}'$, respectively, but always reject *all* type-2 queries. Since type-1 queries are treated identically, $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{B}}'$ are indistinguishable. Moreover:

Claim 4.8. *For all distinguishers* $\mathbf{D}$*,*

$$\Delta^{\mathbf{D}}(\mathbf{B}, \tilde{\mathbf{B}}) \;\leq\; p(1-\tau)^{\alpha n} \quad \textit{and} \quad \Delta^{\mathbf{D}}(\tilde{\mathbf{B}}', \mathbf{B}') \;\leq\; p(1-\tau)^{\alpha n}.$$

The proof of Claim 4.8 follows a generic paradigm, at whose core is the so-called *self-destruct lemma*, which deals with the indistinguishability of hybrids with the self-destruct property and is explained in detail in Section 7. Roughly, this lemma applies whenever the first hybrid (in this case $\mathbf{B}$ resp. $\mathbf{B}'$) can be turned into the second one (in this case $\tilde{\mathbf{B}}$ resp. $\tilde{\mathbf{B}}'$) by changing ("bending") the answers to a subset (the "bending set") of the possible queries to always be $\perp$, and when additionally non-bent queries have a unique answer (cf. the statement of Lemma 7.1). Intuitively, the lemma states that parallelism and adaptivity do not help distinguish (much) in such cases.

Proof. To use the self-destruct lemma, note that $\mathbf{B}$, $\tilde{\mathbf{B}}$, $\tilde{\mathbf{B}}'$, and $\mathbf{B}'$ all answer queries from $\mathcal{X} := \mathbb{F}^{\kappa \times n} \times [\kappa]$ by values from $\mathcal{Y} := \mathbb{F}^n$. Moreover, note that they use as internal randomness a uniformly chosen element T from the set $\mathcal{R} := \binom{[n]}{\tau n}$ of size-τn subsets of $[n]$.

Consider first $\mathbf{B}$ and $\tilde{\mathbf{B}}$. Let $g : \mathcal{X} \times \mathcal{R} \to \mathcal{Y}$ correspond to how $\mathbf{B}$ answers queries $(\mathbf{C}, i)$ (see above). Let $\mathcal{B}$ be the set $\mathcal{B}$ of all type-2a-queries. Then, $\tilde{\mathbf{B}}$ is its $\mathcal{B}$-bending (cf. Definition 7.2). Observe that queries $x = (\mathbf{C}, i) \notin \mathcal{B}$ are either of type 1 or 2b. For the former, the unique answer y_x is the codeword w within αn of the first row of $\mathbf{C}$; for the latter, y_x is $\perp$. Therefore, using the self-destruct lemma (Lemma 7.1), for all distinguishers $\mathbf{D}$,

$$\Delta^{\mathbf{D}}(\mathbf{B}, \tilde{\mathbf{B}}) \;\leq\; p \cdot \max_{(\mathbf{C},i) \in \mathcal{B}} \mathsf{P}[g((\mathbf{C}, i), T) \neq \perp],$$

where the probability is over the choice of T. Since type-2a queries have two rows with distance greater than αn, the probability over the choice of T that this remains unnoticed is at most $(1-\tau)^{\alpha n}$.

For the second part of the claim, consider $\mathbf{B}'$ and $\tilde{\mathbf{B}}'$. Now, let $g : \mathcal{X} \times \mathcal{R} \to \mathcal{Y}$ correspond to how $\mathbf{B}'$ answers queries $(\mathbf{C}, i)$ (see above again), and let $\mathcal{B}$ be the set $\mathcal{B}$ of all type-2-queries. Then, $\tilde{\mathbf{B}}'$ is the $\mathcal{B}$-bending of $\mathbf{B}'$.

Note that all queries $x = (\mathbf{C}, i) \notin \mathcal{B}'$ are of type 1, and the unique answer y_x is the codeword w within $2\alpha n$ of row i of $\mathbf{C}$. Therefore, using Lemma 7.1 again, for all distinguishers $\mathbf{D}$,

$$\Delta^{\mathbf{D}}(\mathbf{B}', \tilde{\mathbf{B}}') \;\leq\; p \cdot \max_{(\mathbf{C},i)\in\mathcal{B}'} \mathsf{P}[g'((\mathbf{C}, i), T) \neq \perp],$$

where the probability is again over the choice of T. Since type-2a queries have two rows with distance greater than αn and in type-2b queries the first row differs in more than αn positions from any codeword, the probability over the choice of T that this remains unnoticed is at most $(1-\tau)^{\alpha n}$. □

Proof (of Claim 4.7). The proof follows using the triangle inequality:

$$\begin{aligned}\Delta^{\mathbf{D}}(\mathbf{B}, \mathbf{B}') &\leq \Delta^{\mathbf{D}}(\mathbf{B}, \tilde{\mathbf{B}}) + \Delta^{\mathbf{D}}(\tilde{\mathbf{B}}, \tilde{\mathbf{B}}') + \Delta^{\mathbf{D}}(\tilde{\mathbf{B}}', \mathbf{B}') \\ &\leq 2 \cdot p(1-\tau)^{\alpha n}.\end{aligned}$$

□

Wrapper. It remains to show that there exists a wrapper $\mathbf{W}_b$ such that $\mathbf{W}_b\mathbf{B}$ behaves as $\mathbf{H}_b$ and $\mathbf{W}_b\mathbf{B}'$ as $\mathbf{H}'_b$. The construction of $\mathbf{W}_b$ is straight forward: $\mathbf{H}_b$ and $\mathbf{H}'_b$ generate all keys and the challenge in the identical fashion; therefore, $\mathbf{W}_b$ can do it the same way. $\mathbf{W}_b$ answers decryption queries $(\mathbf{E}', \mathsf{verk}', \sigma')$ by first verifying the signature σ' and rejecting queries if σ' is invalid or if verk' is identical to the verification key verk^* chosen for the challenge, decrypting the entire matrix $\mathbf{E}'$ to $\mathbf{C}'$ and submitting $(\mathbf{C}', i)$ to the oracle (either $\mathbf{B}$ or $\mathbf{B}'$), where i is the first position at which verk' and verk^* differ, and decoding the answer w and outputting the result or simply forwarding it if it is $\perp$. Moreover, $\mathbf{W}_b$ implements the self-destruct. By inspection it can be seen that $\mathbf{W}_b\mathbf{B}$ implements the original decryption algorithm and $\mathbf{W}_b\mathbf{B}'$ the alternative one.

Reduction to IND-CPA Security

Lemma 4.9. *There exists a reduction* $\mathbf{R}$ *such that for all distinguishers* $\mathbf{D}$*,*

$$\Delta^{\mathbf{D}}(\mathbf{H}'_0, \mathbf{H}'_1) \quad = \quad (1-\tau)\kappa n \cdot \Delta^{\mathbf{DR}}(\mathbf{G}_0^{\mathsf{CPA}}, \mathbf{G}_1^{\mathsf{CPA}}).$$

Proof (sketch). The proof is a straight-forward generalization of the original proof by [CDMW08]; the only difference is that it needs to process multiple parallel decryption queries and implement the self-destruct feature appropriately. For ease of exposition, we describe the reduction to a many-public-key version of the CPA game for Π.[6]

Reduction $\mathbf{R}$ initially chooses the secret set T and creates the challenge OTS key pair with verification key $\mathsf{verk}^* = (v^*[1], \ldots, v^*[\kappa])$ and all key pairs $(\mathsf{pk}^b_{i,j}, \mathsf{sk}^b_{i,j})$ with $j \in T$ or $b \neq v^*[i]$. The remaining $(1-\tau)\kappa n$ key pairs are generated by the CPA game.

Recall that the LECSS is assumed to satisfy the following property: Given τn symbols $(c_i)_{i \in T}$ chosen uniformly at random and independently and any plaintext $m \in \mathbb{F}^k$, one can efficiently sample symbols $(c_i)_{i \notin T}$ such that $(c_1, \ldots, c_n)$ has the same distribution as $\mathsf{E}(m)$. Using this fact, $\mathbf{R}$ creates the challenge for m_0 as follows: It picks the random symbols $(c_i)_{i \in T}$ and completes them to a full encoding c_{m_0} with the above procedure, using m_0 as the plaintext. Let $\mathbf{C}_{m_0}$ be the corresponding matrix (obtained by copying the encodings κ times). Observe that this matrix would match a matrix $\mathbf{C}_{m_1}$ generated from any other message m_1 in the columns indexed by T. These entries are encrypted by $\mathbf{R}$, using the public key $\mathsf{pk}^b_{i,j}$ for entry (i,j) for which $b \neq v^*[i]$. Denote by $\mathbf{C}'_{m_0}$ the matrix $\mathbf{C}_{m_0}$ with the columns in T removed. The reduction outputs $(\mathsf{chall}, \mathbf{C}'_{m_0})$ to its oracle and obtains the corresponding ciphertexts, which it combines appropriately with the ones it created itself to form the challenge ciphertext.

Finally, note that since the reduction knows all the secret keys $\mathsf{pk}^b_{i,j}$ with $b \neq v^*[i]$, it can implement the alternative decryption algorithm (and the self-destruct). □

Overall Proof

Proof (of Theorem 4.4). Let t_{cpa} be the overhead caused by reduction $\mathbf{R}$ and t_{ots} the larger of the overheads caused by $\mathbf{R}'_0$ and $\mathbf{R}'_1$. Moreover, let $\mathbf{D}$ be a distinguisher with running time at most t. Using the triangle

[6] In the many-public-key version of the CPA game, an attacker can play the CPA game for several independently generated public keys simultaneously; this is equivalent to the normal formulation by a standard hybrid argument [BBM00].

inequality, and Lemmas 4.5, 4.6, and 4.9,

$$\begin{aligned}
\Delta^{\mathbf{D}}(\mathbf{G}_0^{\mathsf{NM\text{-}SDA}}, \mathbf{G}_1^{\mathsf{NM\text{-}SDA}}) \;\leq\; & \Delta^{\mathbf{D}}(\mathbf{G}_0^{\mathsf{NM\text{-}SDA}}, \mathbf{H}_0) \\
& + \Delta^{\mathbf{D}}(\mathbf{H}_0, \mathbf{H}_0') + \Delta^{\mathbf{D}}(\mathbf{H}_0', \mathbf{H}_1') \\
& + \Delta^{\mathbf{D}}(\mathbf{H}_1', \mathbf{H}_1) + \Delta^{\mathbf{D}}(\mathbf{H}_1, \mathbf{G}_1^{\mathsf{NM\text{-}SDA}}) \\
\leq\; & \Gamma^{\cdot \mathbf{R}_0' \mathbf{D}}(\mathbf{G}^{\mathsf{ots}}) + 2 \cdot p(1-\tau)^{\alpha n} \\
& + (1-\tau)\kappa n \cdot \Delta^{\cdot \mathbf{R} \mathbf{D}}(\mathbf{G}_{\Pi}^{\mathsf{CPA}} 0, \mathbf{G}_{\Pi}^{\mathsf{CPA}} 1) \\
& + 2 \cdot p(1-\tau)^{\alpha n} + \Gamma^{\cdot \mathbf{R}_1' \mathbf{D}}(\mathbf{G}^{\mathsf{ots}}) \\
\leq\; & \varepsilon_{\mathsf{ots}} + 2 \cdot p(1-\tau)^{\alpha n} \\
& + (1-\tau)\kappa n \cdot \varepsilon_{\mathsf{cpa}} + 2 \cdot p(1-\tau)^{\alpha n} + \varepsilon_{\mathsf{ots}}.
\end{aligned}$$

□

4.4.3 LECSS for the CDMW Construction

In this section we show how to instantiate the LECSS used for the CDMW construction in Section 4.4.1. Let $\mathbb{F}$ be a finite field of size $L = 2^{\ell}$, where ℓ is the plaintext length of the IND-CPA scheme used in the construction. Then, there are the following variants of a (k, n, δ, τ)-LECSS:

- *CDMW Reed-Solomon codes:* The original CDMW construction can be seen as using a Reed-Solomon-based LECSS with rate $\Theta(1/\lambda)$, which is suboptimal (see next item).
- *Constant-Rate Reed-Solomon codes:* Cheraghchi and Guruswami [CG14b] provide a LECSS based on a construction by Dziembowski *et al.* [DPW10] and on Reed-Solomon (RS) codes with $\ell = \Theta(\log n)$. One can show that it achieves the following parameters (not optimized): $\alpha = 1/8$, $\tau = 1/8$ and rate $k/n \geq 1/4$ (i.e., all constant).
- *Algebraic geometric codes:* Using algebraic geometric (AG) codes, Cramer *et al.* [CHH+07] provide a LECSS with $\ell = \mathcal{O}(1)$ and still constant error correction, secrecy, and rate (but with worse concrete constants than Reed-Solomon codes).

Note that asymptotically, RS and AG codes are equally good: both have constant rate, distance, and secrecy. However, since with AG codes ℓ is constant (i.e., they work over an alphabet of constant size), the minimal plaintext length can be shorter than with RS codes.

Chapter 5

PKE Domain Extension via Non-Malleable Codes

Domain extension for public-key encryption (PKE) is the problem of transforming a single-bit PKE satisfying a particular security notion into a multi-bit scheme for the same notion. This section presents two different perspectives on this issue:

- *The constructive approach:* As shown in Section 3.1, PKE schemes can be used to obtain confidential channels from non-confidential ones; in particular, a single-bit scheme achieves a single-bit such channel. To obtain multi-bit confidential channels, one needs to find a way of transforming several single-bit channels into a *single* multi-bit channel. The composition theorem of CC guarantees the security of the composed construction.

- *The game-based approach:* With this approach, a single-bit scheme is combined in an arbitrary (yet preferably black-box) fashion to obtain a multi-bit scheme and the transformation is accompanied by a (direct) reduction from breaking the single-bit scheme to breaking the multi-bit scheme.

  Translated to the game-based view, the constructive approach suggests the following natural *encode-then-encrypt-bit-by-bit (EtEb)* approach to domain extension for PKE: To encrypt a (multi-bit) message, first encode it with a suitable code. Then, encrypt each bit of the resulting codeword (under an independent public key) using the

single-bit scheme.[1]

This chapter shows that using suitable *non-malleable codes (NMCs)*, the EtEb approach can be used to obtain PKE domain extension for all three notions IND-SDA, NM-CPA, and NM-SDA introduced in Chapter 4; to obtain suitable codes for the cases of NM-CPA and NM-SDA security, the classical NMC notion needs to be extended to so-called NMCs *with secret state*.

Both the constructive and the game-based approaches work for all three cases. However, for reasons of presentation and comparison, the constructive approach is used in Section 5.2 for the case of IND-SDA security and the game-based approach in Section 5.3 for NM-CPA and NM-SDA security. The constructive approach is particularly intuitive and conveys very well why non-malleable codes, which are introduced in Section 5.1, are the proper choice for the EtEb paradigm.

Section 5.4 analyzes the efficiency of the constructions based on the EtEb approach and compares it to the efficiency of the schemes by [MS09, HLW12] and to the straight-forward detour via IND-CPA security and the construction by [CDMW08].

Suitable codes to be used with the constructions presented in this chapter are provided in Chapter 6.

5.1 Definitions of Non-Malleable Codes

This section introduces all variants of non-malleable codes (NMCs) used in this thesis: Standard NMCs are defined in Section 5.1.1 and a multi-encoding generalization thereof in Section 5.1.2. The new variant of *secret-state* NMCs, which can be made resilient against *parallel* tampering attacks, is presented in Section 5.1.3. Finally, Section 5.1.4 introduces bit-wise tampering functions, which is the main type of tamper functions encountered in the context of PKE domain extension.

5.1.1 Security against Simple Tampering

Non-malleable codes, introduced by Dziembowski *et al.* [DPW10], are coding schemes (cf. Section 2.9) that protect the encoded messages against certain classes of adversarially chosen modifications, in the sense that the decoding will result either in the original message or in an unrelated value.

[1] Note that transforming single-bit PKE into multi-bit PKE directly allows for a wider variety of transformations. For example, it seems that the schemes by Myers and shelat [MS09] and Hohenberger *et al.* [HLW12] cannot be cast as EtEb transformations.

Tamperable Memory $\mathsf{MEM}^{\mathcal{F},n}$

on first $x \in \{0,1\}^n$ *at* A
| record x

on $(\mathsf{tamper}, \perp)$ *at* E
| **self-destruct**

on (tamper, f) *at* E
| **if** x *has been recorded*
| | $\tilde{x} \leftarrow f(x)$
| | **output** $\tilde{x}$ at B

Figure 5.1: *Tamperable memory* $\mathsf{MEM}^{\mathcal{F},n}$. *The* **self-destruct***-command causes the resource to output* $\perp$ *at interface* B *and halt.*

Basic non-malleable codes [DPW10] provide the above guarantee in a context where the adversary is allowed to modify a (random) codeword c (of a message of his choice) by specifying a function $f : \{0,1\}^n \to \{0,1\}^n$ from a particular function class $\mathcal{F}$ and observe the output of the decoding algorithm applied to the *tampered codeword* $f(c)$.

Continuous non-malleability, introduced in [FMNV14], extends this guarantee to the case where the adversary is allowed to perform multiple such modifications of a target codeword c. That is, he can repeatedly and adaptively specify functions $f \in \mathcal{F}$ and see the decoding of the tampered codeword $f(c)$.[2] Unless explicitly stated otherwise, all statements in this thesis are w.r.t. *continuous* non-malleability.

Memory resource. Non-malleable codes can be captured constructively in that they transform tamperable memory into untamperable memory. To that end, consider the $\{A, B, E\}$-setting resource $\mathsf{MEM}^{\mathcal{F},n}$ in Figure 5.1, for a class $\mathcal{F}$ of functions $f : \{0,1\}^n \to \{0,1\}^n$. It allows a *single* message x to be encoded at interface A. At the attacker interface E, the resource (repeatedly) accepts instructions (tamper, f) for $f \in \mathcal{F}$, which causes $f(x)$ to be output at interface B. The resource also allows E to issue the command $(\mathsf{tamper}, \perp)$, upon which the resource "self-destructs," i.e., it outputs $\perp$ at B and halts.

Coding schemes as protocols. Any (k, n)-coding scheme $\mathrm{CS} = (\mathrm{Enc}, \mathrm{Dec})$ can be turned into a protocol $\mathsf{nmc} = (\mathsf{encode}, \mathsf{decode})$ that can be attached to $\mathsf{MEM}^{\mathcal{F},n}$ in a straight-forward manner:

- Converter encode, upon receiving m at the outside interface, computes $\mathrm{Enc}(m)$ and outputs it at the inside interface.
- Converter decode, upon receiving $\tilde{c}$ at the inside interface, computes $\tilde{m} \leftarrow \mathrm{Dec}(\tilde{c})$ and outputs it at the outside interface. If $\tilde{m} = \perp$,

[2] The functions f specified by the adversary are always applied to the same c.

decode halts. Upon receiving $\perp$ at the inside interface, decode outputs $\perp$ at the outside interface and halts.

Defining non-malleability. The non-malleability requirement can now be captured by requiring that nmc (based on (Enc, Dec)) achieve

$$\mathsf{MEM}^{\mathcal{F},n} \overset{\mathsf{nmc},\sigma,(0,\varepsilon)}{\Longrightarrow} \mathsf{MEM}^{\mathcal{F}_{\mathsf{triv}},k} \tag{5.1}$$

for some simulator σ and $\varepsilon \geq 0$, where $\mathcal{F}_{\mathsf{triv}}$ is the class of trivial functions, i.e., it consists only of the identity function, denoted by id, and constant functions.
Set

$$\mathbf{R}^{\mathcal{F},\mathrm{CS}} := \mathsf{encode}^A \mathsf{decode}^B \mathsf{MEM}^{\mathcal{F},n}$$

and

$$\mathbf{S}^{\sigma} := \mathbf{S}^{\mathcal{F}_{\mathsf{triv}},\sigma} := \sigma^E \mathsf{MEM}^{\mathcal{F}_{\mathsf{triv}},k}.$$

Then, the standard definition of (continuous) non-malleability is recovered by requiring that

$$\Delta^{\mathbf{D}}(\mathbf{R}^{\mathcal{F},\mathrm{CS}}, \mathbf{S}^{\sigma}) \leq \varepsilon$$

for all distinguishers $\mathbf{D}$.

Definition 5.1. *A coding scheme* $\mathrm{CS} = (\mathrm{Enc}, \mathrm{Dec})$ *is* $(\mathcal{F}, \varepsilon)$-non-malleable *if there exists a (efficient) simulator* σ *such that*

$$\Delta^{\mathbf{D}}(\mathbf{R}^{\mathcal{F},\mathrm{CS}}, \mathbf{S}^{\sigma}) \leq \varepsilon$$

for all distinguishers $\mathbf{D}$.

Strong non-malleability. The original paper on NMCs by Dziembowski *et al.* [DPW10] also introduced a notion of *strong* non-malleability, which can be thought of as prescribing a particular simulation strategy. This notion is not considered in this thesis.

Non-malleable reductions. Instead of requiring that a coding scheme immediately achieve construction (5.1), one can instead consider an intermediate (weaker) tamper-function class $\mathcal{F}'$ and consider the construction

$$\mathsf{MEM}^{\mathcal{F},n} \overset{\mathsf{nmc},\sigma,(0,\varepsilon)}{\Longrightarrow} \mathsf{MEM}^{\mathcal{F}',k}. \tag{5.2}$$

This leads to the notion of a *non-malleable reduction*, a concept introduced by Aggarwal *et al.* [ADKO15b].

Definition 5.2. *A coding scheme* $(\mathrm{Enc}, \mathrm{Dec})$ *is an* $(\mathcal{F}, \mathcal{F}', \varepsilon)$-non-malleable reduction *if there exists a (efficient) simulator* σ *such that*

$$\Delta^{\mathbf{D}}(\mathbf{R}^{\mathcal{F},\mathrm{CS}}, \mathbf{S}^{\mathcal{F}',\sigma}) \leq \varepsilon$$

for all distinguishers $\mathbf{D}$.

Non-malleable reductions are composable: If CS is an $(\mathcal{F}, \mathcal{F}')$-reduction and CS$'$ a $(\mathcal{F}', \mathcal{F}'')$-reduction, then combining CS and CS$'$ (in a straightforward manner) yields an $(\mathcal{F}, \mathcal{F}'')$-reduction. This follows from the CC composition theorem (cf. Section 2.6).

5.1.2 Security when Encoding Many Messages

This section introduces the notion of *adaptive* (continuous) non-malleability, which is an extension of (continuous) non-malleability in that the adversary is allowed to adaptively specify *multiple messages* $m^{(1)}, m^{(2)}, \ldots$ and the functions may depend on all of the corresponding codewords $c^{(1)}, c^{(2)}, \ldots$. That is, the tamper class $\bar{\mathcal{F}}$ is actually a sequence $(\mathcal{F}^{(i)})_{i\geq 1}$ of function families with $\mathcal{F}^{(i)} \subseteq \{f \mid f : (\{0,1\}^n)^i \to \{0,1\}^n\}$, and after encoding i messages, the adversary chooses functions from $\mathcal{F}^{(i)}$.[3]

For the definition of adaptive non-malleability, consider the resource $\mathsf{A\text{-}MEM}^{\mathcal{F},n}$ depicted in Figure 5.2. Similarly to Section 5.1.1, let $\mathsf{nmc} = (\mathsf{encode}, \mathsf{decode})$ be the protocol built based on a (k, n)-coding scheme $\mathrm{CS} = (\mathrm{Enc}, \mathrm{Dec})$ and set

$$\bar{\mathbf{R}}^{\bar{\mathcal{F}},\mathrm{CS}} := \mathsf{encode}^A \mathsf{decode}^B \mathsf{A\text{-}MEM}^{\bar{\mathcal{F}},n}$$

and

$$\bar{\mathbf{S}}^{\sigma} := \bar{\mathbf{S}}^{\bar{\mathcal{F}}_{\mathsf{triv}},\sigma} := \sigma^E \mathsf{A\text{-}MEM}^{\bar{\mathcal{F}}_{\mathsf{triv}},k}.$$

for a simulator σ, where $\bar{\mathcal{F}}_{\mathsf{triv}} = (\bar{\mathcal{F}}^{(i)}_{\mathsf{triv}})_{i\geq 1}$ and $\bar{\mathcal{F}}^{(i)}_{\mathsf{triv}}$ consists of constant functions and functions $\mathrm{id}^{(i)}$ with $\mathrm{id}^{(j)}(x^{(1)}, \ldots, x^{(i)}) = x^{(j)}$.

Definition 5.3. *A coding scheme* $\mathrm{CS} = (\mathrm{Enc}, \mathrm{Dec})$ *is* $(\bar{\mathcal{F}}, \varepsilon, \ell, q)$-adaptive non-malleable *if there exists a (efficient) simulator* σ *such that*

$$\Delta^{\mathbf{D}}(\langle \bar{\mathbf{R}}^{\bar{\mathcal{F}},\mathrm{CS}} \rangle_{\ell,q}, \langle \bar{\mathbf{S}}^{\sigma} \rangle_{\ell,q}) \leq \varepsilon$$

[3] A similar adaptive notion has been already considered for continuous *strong* non-malleability in the split-state model [FMNV15].

Adaptively Tamperable Memory $\mathsf{A\text{-}MEM}^{\mathcal{F},n}$

init
  $i \leftarrow 0$

on $x \in \{0,1\}^n$ *at* A
  $i \leftarrow i+1$
  $x^{(i)} \leftarrow x$

on $(\mathsf{tamper}, \perp)$ *at* E
  self-destruct

on (tamper, f) *with* $f \in \mathcal{F}^{(i)}$ *at* E
  if $i > 0$
    $\tilde{x} \leftarrow f(x^{(1)}, \ldots, x^{(i)})$
    output $\tilde{x}$ at B

Figure 5.2: *Adaptively tamperable memory* $\mathsf{A\text{-}MEM}^{\mathcal{F},n}$*. The command* **self-destruct** *causes the resource to output* $\perp$ *at interface* B *and halt.*

for all distinguishers $\mathbf{D}$*, where* $\langle\cdot\rangle_{\ell,q}$ *denotes that only the first* ℓ *queries at the* A*-interface and only the first* q *queries at the* E*-interface are processed.*[4]

Note that $(\bar{\mathcal{F}}, \varepsilon, 1, *)$-adaptive non-malleable (where $*$ stands for "arbitrary") is the same as $(\mathcal{F}^{(1)}, \varepsilon)$-non-malleable.

5.1.3 A Novel Flavor of Non-Malleability: Parallel Tampering

Codes with secret state. PKE domain extension based on non-malleable codes for NM-CPA and NM-SDA-secure PKE schemes (cf. Section 5.3) requires security against *parallel* tampering attacks, in which an attacker gets to repeatedly (and adaptively) specify parallel queries. The first time a parallel query results in an invalid encoding, the self-destruct mode occurs, but the attacker still obtains the answers to that entire parallel query.

It can be shown that this security level cannot be achieved by conventional non-malleable codes (cf. Section 6.3.3). To circumvent this impossibility, one can consider so-called *secret-state* non-malleable codes.

Definition 5.4 (Coding scheme with secret state)**.** *A* (k, n)-coding scheme with secret state (CSS) *is a triple of algorithms* $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$*, where the (randomized)* state-generation algorithm Gen *outputs a* secret state s *from some set* $\mathcal{S}$*, the (randomized)* encoding algorithm $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$ *takes a* k*-bit plaintext* m *and outputs an* n*-bit encoding* $c \leftarrow \mathrm{Enc}(m)$*, and the (deterministic)* decoding algorithm $\mathrm{Dec} : \{0,1\}^n \times \mathcal{S} \to \{0,1\}^k \cup \{\perp\}$ *takes an encoding as well as some secret state* $s \in \mathcal{S}$ *and outputs a plaintext* $m \leftarrow \mathrm{Dec}(c, s)$ *or the special symbol* $\perp$*, indicating an* invalid encoding.

[4] The reason for making both the number ℓ of encoded messages and the number q of tamper queries explicit is that the security of constructions of adaptive non-malleable codes in this thesis depend on these quantities.

Parallel Tamperable Memory $\mathsf{P\text{-}MEM}_p^{\mathcal{F},n}$

on first x *at* A
  record x

on $(\mathsf{tamper}, \perp)$ *at* E
  self-destruct

on $(\mathsf{tamper}, f^{(1)}, \ldots, f^{(p)})$ *at* E
  if x *has been recorded*
    for $j = 1$ **to** p
      $x^{(j)} \leftarrow f^{(j)}(x)$
    output $(x^{(1)}, \ldots, x^{(p)})$ at B
    if $\exists j : x^{(j)} = \perp$
      self-destruct

Figure 5.3: *Parallel tamperable memory* $\mathsf{P\text{-}MEM}_p^{\mathcal{F},n}$. *The command* **self-destruct** *causes the resource to output* $\perp$ *at interface* B *and halt.*

Memory resource. To define security against parallel tampering, consider the $\{A, B, E\}$-setting resource $\mathsf{P\text{-}MEM}_p^{\mathcal{F},n}$ in Figure 5.3, for a class $\mathcal{F}$ of functions $f : \{0,1\}^n \to \{0,1\}^n$. It allows a single message x to be encoded at interface A. At the attacker interface E, the resource (repeatedly) accepts instructions $(\mathsf{tamper}, f^{(1)}, \ldots, f^{(p)})$ for $f^{(j)} \in \mathcal{F}$, which causes $(f^{(1)}(x), \ldots, f^{(p)}(x))$ to be output at interface B. The resource also allows E to issue the command $(\mathsf{tamper}, \perp)$, upon which the resource "self-destructs," i.e., it outputs $\perp$ at B and halts.

Coding schemes as protocols. For the case of parallel tampering, the protocol $\mathsf{p\text{-}nmc} = (\mathsf{p\text{-}encode}, \mathsf{p\text{-}decode})$ attached to $\mathsf{P\text{-}MEM}_p^{\mathcal{F},n}$ based on a coding scheme with secret state $\mathrm{CS} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ is obtained as follows:

- Converter $\mathsf{p\text{-}encode}$, upon receiving m at the outside interface, computes $\mathrm{Enc}(m)$ and outputs it at the inside interface.
- Converter $\mathsf{p\text{-}decode}$ initially generates the secret state $s \leftarrow \mathrm{Gen}$. Upon receiving $(c^{(1)}, \ldots, c^{(p)})$ at the inside interface, $\mathsf{p\text{-}decode}$ computes $m^{(j)} \leftarrow \mathrm{Dec}(c^{(j)}, s)$ for $j = 1, \ldots, p$ and outputs $(m^{(1)}, \ldots, m^{(p)})$ at the outside interface. If $m^{(j)} = \perp$ for some j, $\mathsf{p\text{-}decode}$ halts. Upon receiving $\perp$ at the inside interface, $\mathsf{p\text{-}decode}$ outputs $\perp$ at the outside interface and halts.

Defining non-malleability. Non-malleability against parallel tampering can now be captured by requiring that nmc (based on $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$) achieve

$$\mathsf{P\text{-}MEM}_p^{\mathcal{F},n} \overset{\mathsf{p\text{-}nmc},\sigma,(0,\varepsilon)}{\Longrightarrow} \mathsf{P\text{-}MEM}_p^{\mathcal{F}_{\mathsf{triv}},k} \tag{5.3}$$

for some simulator σ and $\varepsilon \geq 0$, where $\mathcal{F}_{\mathsf{triv}}$ is the class of trivial functions, i.e., it consists only of the identity function, denoted by id, and constant

functions.
Set

$$\mathbf{R}_p^{\mathcal{F},\mathrm{CS}} := \mathsf{p\text{-}encode}^A \mathsf{p\text{-}decode}^B \mathsf{P\text{-}MEM}_p^{\mathcal{F},n}$$

and

$$\mathbf{S}_p^{\sigma} := \mathbf{S}_p^{\mathcal{F}_{\mathsf{triv}},\sigma} := \sigma^E \mathsf{P\text{-}MEM}_p^{\mathcal{F}_{\mathsf{triv}},k}.$$

Then, the definition of (continuous) non-malleability against parallel tampering is obtained by requiring that

$$\Delta^{\mathbf{D}}(\mathbf{R}_p^{\mathcal{F},\mathrm{CS}}, \mathbf{S}_p^{\sigma}) \leq \varepsilon$$

for all distinguishers $\mathbf{D}$.

Definition 5.5. *A coding scheme with secret state* $\mathrm{CS} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *is* $(\mathcal{F}, \varepsilon)$-parallel non-malleable *if there exists a (efficient) simulator* σ *such that*

$$\Delta^{\mathbf{D}}(\mathbf{R}_p^{\mathcal{F},\mathrm{CS}}, \mathbf{S}_p^{\sigma}) \leq \varepsilon$$

for all distinguishers $\mathbf{D}$.

Remark. One could also define an adaptive variant for parallel tampering, but this is beyond the scope of this thesis.

5.1.4 Bit-Wise Tampering Functions

This thesis considers two main tamper-function classes $\mathcal{F}_{\mathsf{bit}}$ and $\mathcal{F}_{\mathsf{copy}}$ that allow an attacker to tamper with each bit of an encoding *individually*. While $\mathcal{F}_{\mathsf{bit}}$ allows arbitrary tampering with each bit, functions in $\mathcal{F}_{\mathsf{copy}}$ are not allowed to flip bits of the encoding.

Tampering with a single encoding. A function $f : \{0,1\}^n \to \{0,1\}^n$ in $\mathcal{F}_{\mathsf{bit}}$ is characterized by a vector $(f[1], \ldots, f[n])$, where $f[i] : \{0,1\} \to \{0,1\}$ is the function applied to the i^{th} bit. That is, $f[i] \in \{\mathsf{zero}, \mathsf{one}, \mathsf{keep}, \mathsf{flip}\}$ with the meaning that the i^{th} bit is either set to 0 (zero), set to 1 (one), unchanged (keep), or flipped (flip). Similarly, a function $f[i] \in \mathcal{F}_{\mathsf{copy}}$ is restricted to being $f[i] \in \{\mathsf{zero}, \mathsf{one}, \mathsf{keep}\}$. In slight abuse of notation, one writes $f = (f[1], \ldots, f[n])$.

For a function f in $\mathcal{F}_{\mathsf{bit}}$ or $\mathcal{F}_{\mathsf{copy}}$, let $A(f) := \{i \mid i \in \{\mathsf{zero}, \mathsf{one}\}\}$ and $B(f) := \{i \mid i \in \{\mathsf{keep}, \mathsf{flip}\}\}$. Moreover, set $a(f) := |A(f)|$. Finally, let $\mathsf{val}(\mathsf{zero}) := \mathsf{val}(\mathsf{keep}) := 0$ and $\mathsf{val}(\mathsf{one}) := \mathsf{val}(\mathsf{flip}) := 1$.

Tampering with multiple encodings. The (sequences of) function classes corresponding to adaptive bit-wise tampering with and without bit flips are $\bar{\mathcal{F}}_{\mathsf{bit}} = (\mathcal{F}^{(i)}_{\mathsf{bit}})_{i\geq 1}$ and $\bar{\mathcal{F}}_{\mathsf{copy}} = (\mathcal{F}^{(i)}_{\mathsf{copy}})_{i\geq 1}$, respectively.

A function $f : (\{0,1\}^n)^i \to \{0,1\}^n$ in $\mathcal{F}^{(i)}_{\mathsf{bit}}$ is characterized by a vector $(f[1], \ldots, f[n])$, where $f[i] : \{0,1\} \to \{0,1\}$ is the function applied to the i^{th} bit. That is, $f[v] \in \{\mathsf{zero}, \mathsf{one}, \mathsf{keep}_1, \ldots, \mathsf{keep}_i, \mathsf{flip}_1, \ldots \mathsf{flip}_i\}$ with the meaning that the v^{th} bit is either set to 0 (zero), set to 1 (one), the v^{th} bit of the j^{th} encoding (keep_j), or the flipped v^{th} bit of the j^{th} encoding (flip_j). Similarly, a function $f \in \mathcal{F}_{\mathsf{copy}}$ is restricted to being $f[v] \in \{\mathsf{zero}, \mathsf{one}, \mathsf{keep}_1, \ldots, \mathsf{keep}_i\}$.

5.2 From Single-Bit to Multi-Bit Channels

This section presents the constructive approach to PKE domain extension.

Single-bit PKE viewed constructively. As argued in Section 4.3, one can show that a 1-bit IND-SDA-secure PKE scheme can be used to design a protocol $\mathsf{1\text{-}pke}'$ that achieves the construction

$$[\mathsf{1\text{-}AUTH}_{BA}, \mathsf{INSEC}_{AB}] \quad \overset{\mathsf{1\text{-}pke}'}{\Longmapsto} \quad \mathsf{CONF}^{\text{1-bit}}_{AB} \tag{5.4}$$

Using the composition theorem, one then obtains

$$[\mathsf{1\text{-}AUTH}_{BA}, \mathsf{INSEC}_{AB}]^n \quad \overset{\mathsf{1\text{-}pke}''}{\Longmapsto} \quad [\mathsf{CONF}^{\text{1-bit}}_{AB}]^n,$$

where $\mathsf{1\text{-}pke}'' = (\mathsf{1\text{-}enc}'', \mathsf{1\text{-}dec}'')$ and where $\mathsf{1\text{-}enc}''$ and $\mathsf{1\text{-}dec}''$ are the n-fold parallel composition of $\mathsf{1\text{-}enc}'$ and $\mathsf{1\text{-}dec}'$, respectively. A slight modification of protocol $\mathsf{1\text{-}pke}''$ yields a protocol $\mathsf{1\text{-}pke}$ for construction

$$[\mathsf{1\text{-}AUTH}_{BA}, \mathsf{INSEC}_{AB}] \quad \overset{\mathsf{1\text{-}pke}}{\Longmapsto} \quad [\mathsf{CONF}^{\text{1-bit}}_{AB}]^n, \tag{5.5}$$

Essentially, all public keys are concatenated and sent via a single $\longleftarrow\!\bullet$. A proof of security is straight-forward.

Tying the channels together. Using an adaptive (continuously) non-malleable (k, n)-code (cf. Section 5.1.2), a (single) k-bit confidential channel can be constructed from the n independent single-bit confidential channels. This is achieved by having the sender encode the message with the non-malleable code and sending the resulting codeword over the 1-bit channels (bit-by-bit), while the receiver decodes all n-bit strings received

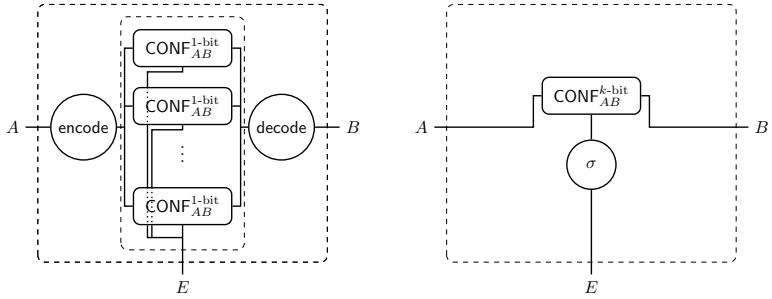

Figure 5.4: *Left: The assumed resource* $[\mathsf{CONF}_{AB}^{1\text{-bit}}]^n$ *with protocol converters* encode *and* decode *attached to interfaces* A *and* B*, denoted* $\mathsf{encode}^A\mathsf{decode}^B[\mathsf{CONF}_{AB}^{1\text{-bit}}]^n$*. Right: The constructed resource* $\mathsf{CONF}_{AB}^{k\text{-bit}}$ *with simulator* σ *attached to the* E*-interface, denoted* $\sigma^E\mathsf{CONF}_{AB}^{k\text{-bit}}$*. In particular,* σ *must simulate the* E*-interfaces of* $[\mathsf{CONF}_{AB}^{1\text{-bit}}]^n$*. The protocol is secure if the two systems are indistinguishable.*

via these channels. Additionally, due to the self-destruct property of continuously non-malleable codes, the receiver must stop decoding once an invalid codeword has been received.

More precisely, let $\mathrm{CS} = (\mathrm{Enc}, \mathrm{Dec})$ be a (k, n)-coding scheme and consider the following protocol $\mathsf{nmc} = (\mathsf{encode}, \mathsf{decode})$: Converter encode encodes every message $m \in \{0,1\}^k$ input at its outside interface with fresh randomness, resulting in an n-bit encoding $c = c[1] \cdots c[n] \leftarrow \mathrm{Enc}(m)$. Then, for $i = 1, \ldots, n$, it outputs bit $c[i]$ to the i^{th} channel at the inside interface. Converter decode, whenever it receives an n-bit string $c' = c'[1] \cdots c'[n]$ (where the i^{th} bit $c'[i]$ was received on the i^{th} channel), it computes $m' \leftarrow \mathrm{Dec}(c')$ and outputs m' at the outside interface. If $m' = \perp$, it implements the self-destruct mode, i.e., it halts.

The goal is now to show that protocol nmc achieves the construction

$$[\mathsf{CONF}_{AB}^{1\text{-bit}}]^n \quad \overset{\mathsf{nmc}}{\Longrightarrow} \quad \mathsf{CONF}_{AB}^{k\text{-bit}}. \tag{5.6}$$

The required non-malleability. By inspecting both sides of Figure 5.4, it becomes apparent why adaptive (continuously) non-malleable codes are the proper choice to achieve construction (5.6): On the left-hand side, the distinguisher can repeatedly input messages $m^{(i)}$ at interface A, which results in encodings $c^{(i)}$ being input (bit-by-bit) into the single-bit channels. Using the E-interfaces of these channels, the distinguisher can repeatedly see the decoding of an n-bit string $c' = c'[1] \cdots c'[n]$ at interface B, where

each bit $c'[j]$ results from either forwarding one of the bits already in the j^{th} channel or from injecting a fresh bit that is either 0 or 1.

Put differently, the distinguisher can effectively launch tampering attacks using functions from $\bar{\mathcal{F}}_{\mathsf{copy}}$ (cf. Section 5.1.2).

On the right-hand side, the distinguisher may again input messages $m^{(i)}$ at interface A—to the k-bit confidential channel. At interface E, this channel only allows to either deliver entire k-bit messages already sent by A or to inject independent messages. The simulator σ required to prove (5.6) needs to simulate the E-interfaces of the single-bit confidential channels at its outside interface and, based solely on what is input at these interfaces, decide whether to forward or inject a message, which corresponds exactly to the task of the simulator σ in the non-malleability experiment (cf. Section 5.1.2).

Theorem 5.1 below formalizes this correspondence; its proof is essentially a technicality—one merely needs to "translate" between the channel settings and the non-malleability experiment—and is therefore omitted.

Theorem 5.1. *For any $\ell, q \in \mathbb{N}$, if a (k,n)-coding scheme* $(\mathrm{Enc}, \mathrm{Dec})$ *is $(\bar{\mathcal{F}}_{\mathsf{copy}}, \varepsilon, \ell, q)$-adaptive non-malleable, there exists a simulator σ such that*

$$[\langle \mathsf{CONF}_{AB}^{1\text{-bit}} \rangle_{\ell,q}]^n \overset{\mathsf{nmc},\sigma,(0,\varepsilon)}{\Longrightarrow} \langle \mathsf{CONF}_{AB}^{k\text{-bit}} \rangle_{\ell,q},$$

where $\langle\cdot\rangle_{\ell,q}$ denotes a channel that only processes the first ℓ queries at the A-interface and only the first q queries at the E-interface.

Plugging it together. The composition theorem of constructive cryptography (cf. Section 2.6) implies that the protocol $\mathsf{m\text{-}pke} = \mathsf{nmc} \circ \mathsf{1\text{-}pke}$ resulting from composing the protocols 1-pke and nmc for transformations (5.5) and (5.6), respectively, achieves

$$[\mathsf{1\text{-}AUTH}_{BA}, \mathsf{INSEC}_{AB}] \overset{\mathsf{m\text{-}pke}}{\Longrightarrow} \mathsf{CONF}_{AB}^{k\text{-bit}}. \tag{5.7}$$

Protocol m-pke corresponds (in a straight-forward manner) to a PKE scheme Π that achieves IND-SDA security. A direct, game-based proof of this fact can be obtained as a special case of the game-based proof for NM-SDA security provided in Section 5.3. The resulting proof is a hybrid argument and can be obtained by "unwrapping" the concatenation of the statements in this section. The modular nature and the intuitive simplicity of the proofs are lost, however.

5.3 Domain Extension for NM-SDA-Secure PKE

This section shows how to combine a single-bit PKE NM-SDA scheme with a non-malleable code resilient against parallel tampering to obtain a multi-bit NM-SDA PKE scheme. Due to the nature of the transformation, the resulting scheme is only *replayable* NM-SDA-secure (cf. Section 3.2.2 about replayable security). However, there is a generic transformation from replayable to full security (cf. [CKN03]) that can also be applied here.

All results in this section translate to the notion of NM-CPA in a straight-forward manner.

5.3.1 Replayable NM-SDA Security

Recall the relaxation IND-RCCA of full IND-CCA-security presented in Section 3.2.2. This idea carries over seamlessly to the definition of NM-SDA security; the corresponding distinguishing game $\mathbf{G}_b^{\text{NM-RSDA}}$ is obtained by changing $\mathbf{G}_b^{\text{NM-SDA}}$ (cf. Figure 4.2 in Section 4.1) via the valid-predicate as shown in Section 2.8.

Definition 5.6. *A PKE scheme* Π *is* replayable (t, p, ε)-NM-SDA-secure (NM-RSDA) *if for all distinguishers* $\mathbf{D}$ *with running time at most* t*,*

$$\Delta^{\mathbf{D}}(\mathbf{G}_{p,0}^{\Pi,\text{NM-RSDA}}, \mathbf{G}_{p,1}^{\Pi,\text{NM-RSDA}}) \;\leq\; \varepsilon.$$

Remark. Since the statements in this section depend only on the width p of the parallel queries (but not on the number q of parallel queries), this is the only parameter made explicit. The same is the case for the non-malleable codes (resilient against parallel tampering) used below.

5.3.2 Combining Non-Malleable Codes and PKE

The construction of a multi-bit NM-RSDA-secure PKE scheme Π' from a single-bit NM-SDA-secure scheme Π and a secret-state non-malleable (k, n)-code $(\text{Gen}, \text{Enc}, \text{Dec})$ works as follows: It encrypts a k-bit message m by first computing an encoding $c = (c[1], \ldots, c[n])$ of m and then encrypting each bit $c[j]$ under an independent public key of Π; it decrypts by first decrypting the individual components and then decoding the resulting codeword using the secret state of the non-malleable code; the secret state is part of the secret key. The scheme is depicted in detail in Figure 5.5.

PKE Scheme $\Pi' = (K', E', D')$

Key Generation K'
  for $i \leftarrow 1$ **to** n
    $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow K$
  $\mathsf{pk} \leftarrow (\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$
  $\mathsf{sk} \leftarrow (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
  $s \leftarrow \mathrm{Gen}$
  return $(\mathsf{pk}, (\mathsf{sk}, s))$

Encryption $E'_{\mathsf{pk}}(m)$
  $c = (c[1], \ldots, c[n]) \leftarrow \mathrm{Enc}(m)$
  for $i \leftarrow 1$ **to** n
    $e_i \leftarrow E_{\mathsf{pk}_i}(c[i])$
  return $e = (e_1, \ldots, e_n)$

Decryption $D'_{(\mathsf{sk},s)}(e)$
  $(e_1, \ldots, e_n) \leftarrow e$
  for $i \leftarrow 1$ **to** n
    $c[i] \leftarrow D_{\mathsf{sk}_i}(e_i)$
    if $c[i] = \perp$
      return $\perp$
  $m \leftarrow \mathrm{Dec}(c[1] \cdots c[n], s)$
  return m

Figure 5.5: *The k-bit PKE scheme $\Pi' = (K', E', D')$ built from a 1-bit PKE scheme $\Pi = (K, E, D)$ and a (k, n)-coding scheme with secret state* $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$.

Theorem 5.2. *Let $p \in \mathbb{N}$ and Π be a $(t_{\mathsf{rsda}} + t_{\mathsf{1bit}}, p, \varepsilon_{\mathsf{1bit}})$-NM-SDA-secure 1-bit PKE scheme and let $\mathrm{CS} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ be $(\mathcal{F}_{\mathsf{copy}}, p, \varepsilon_{\mathsf{nmc}})$-parallel non-malleable. Then, Π' is $(t_{\mathsf{rsda}}, p, \varepsilon_{\mathsf{rsda}})$-NM-RSDA-secure PKE scheme with*

$$\varepsilon_{\mathsf{rsda}} \quad = \quad 2(n\varepsilon_{\mathsf{1bit}} + \varepsilon_{\mathsf{nmc}}),$$

where t_{1bit} represents the overhead incurred by the reductions.

In the following, let $\mathcal{F} := \mathcal{F}_{\mathsf{copy}}$. The proof follows directly from the following lemma:

Lemma 5.3. *For $b \in \{0,1\}$ and $i \in [n]$, there exist reductions $\mathbf{C}_{b,i}$ and $\mathbf{W}_b$ such that for all distinguishers $\mathbf{D}$,*

$$\begin{aligned}\Delta^{\mathbf{D}}(\mathbf{G}_{p,0}^{\Pi',\mathsf{NM\text{-}RSDA}}, \mathbf{G}_{p,1}^{\Pi',\mathsf{NM\text{-}RSDA}}) \\ \leq \sum_{b,i} \Delta^{\mathbf{DC}_{b,i}}(\mathbf{G}_{p,0}^{\Pi,\mathsf{NM\text{-}SDA}}, \mathbf{G}_{p,1}^{\Pi,\mathsf{NM\text{-}SDA}}) \\ + \sum_{b} \Delta^{\mathbf{DW}_b}(\mathbf{R}^{\mathsf{copy}}, \mathbf{S}^{\mathsf{trivial}}),\end{aligned}$$

where σ is the simulator for the non-malleable code.

Towards a proof of Lemma 5.3, consider the following hybrids for $b \in \{0,1\}$ and $i \in [n]$: $\mathbf{H}_{b,i}$ proceeds as $\mathbf{G}_{p,b}^{\Pi',\mathsf{NM\text{-}RSDA}}$ except that the challenge query (chall, m_0) and decryption queries $(\mathsf{dec}, e^{(1)}, \ldots, e^{(p)})$ are handled differently:

- **Challenge query:** The first i bits of the encoding $c = (c[1], \ldots, c[n])$ of m_b are replaced by uniformly random and independent bits. The

resulting n-bit string is then encrypted bit-wise (as done by E'). This results in the challenge ciphertext $e^* = (e_1^*, \ldots, e_n^*)$.

- **Decryption query:** Every component $e^{(l)} = (e'_1, \ldots, e'_n)$ is answered as follows: Hybrid $\mathbf{H}_{b,i}$ computes $c' = (c'[1], \ldots, c'[n])$, where

$$c'[i] \quad = \quad \begin{cases} c[j] & \text{if } e'_j = e_j^*, \text{ and} \\ D_{\mathsf{sk}_j}(e'_j) & \text{otherwise.} \end{cases}$$

  Then, $\mathbf{H}_{b,i}$ outputs $\mathrm{Dec}(c', s)$ as the answer to the component of the decryption query.[5]

Let $\mathbf{H}_{b,0} := \mathbf{G}_{p,b}^{\Pi',\mathsf{NM\text{-}RSDA}}$.

Lemma 5.4. *For all $b \in \{0,1\}$ and $i \in [n]$, there exist reductions $\mathbf{C}_{b,i}$ such that for all $\mathbf{D}$*

$$\Delta^{\mathbf{D}}(\mathbf{H}_{b,i-1}, \mathbf{H}_{b,i}) \quad = \quad \Delta^{\mathbf{D}\mathbf{C}_{b,i}}(\mathbf{G}_{p,0}^{\Pi,\mathsf{NM\text{-}SDA}}, \mathbf{G}_{p,1}^{\Pi,\mathsf{NM\text{-}SDA}}).$$

Proof. Fix b and i. Hybrid $\mathbf{C}_{b,i}$ works as follows: Initially, it generates the secret state $s \leftarrow \mathrm{Gen}$ and $n-1$ key pairs $(\mathsf{pk}_j, \mathsf{sk}_j)$ for $j \in [n] \setminus \{i\}$, obtains pk_i (but not sk_i) from the oracle (from $\mathbf{G}_{p,0}^{\Pi,\mathsf{NM\text{-}SDA}}$ or $\mathbf{G}_{p,0}^{\Pi,\mathsf{NM\text{-}SDA}}$), and outputs $\mathsf{pk} := (\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$. When it receives (chall, m_0), it chooses $m_1 \leftarrow \{0,1\}^k$ and computes an encoding $c = (c[1], \ldots, c[n]) \leftarrow \mathrm{Enc}(m_b)$. Then, it chooses i random bits $\tilde{c}[1], \ldots, \tilde{c}[i]$ and computes

$$e_j^* \quad = \quad \begin{cases} E_{\mathsf{pk}_j}(\tilde{c}[j]) & \text{for } j < i, \text{ and} \\ E_{\mathsf{pk}_j}(c[j]) & \text{for } j > i. \end{cases}$$

Moreover, it outputs $(\mathsf{chall}, c[i])$ to its oracle and obtains a ciphertext e_i^*. It finally returns $e^* = (e_1^*, \ldots, e_n^*)$.

When $\mathbf{C}_{b,i}$ receives a (parallel) decryption query, for each component $e' = (e'_1, \ldots, e'_n)$ it proceeds as follows: For $j \neq i$, it computes $c'[j]$ as $\mathbf{H}_{b,i}$ does. Moreover, if $e'_i = e_i^*$, it sets $c'[i] \leftarrow c[i]$. Otherwise, it outputs (dec, e'_i) to its oracle and obtains the answer $c'[i]$.[6] Then, it computes $m' \leftarrow \mathrm{Dec}(c')$. The answer to the component of the decryption query is m', unless $m' \in \{m_0, m_1\}$, in which case the it is test. If one of the

[5] Assume here and below that $\mathrm{Dec}(c') = \perp$ if any of the bits $c'[j]$ equal $\perp$.

[6] In fact, it is important that $\mathbf{C}_{b,i}$ output a single parallel decryption query containing all e'_i for the individual components; but it is less cumbersome to describe how individual components are handled.

component answers is $\perp$, $\mathbf{C}_{b,i}$ implements the self-destruct mode, i.e., it halts.

Consider $\mathbf{C}_{b,i}\mathbf{G}_{p,0}^{\Pi,\mathsf{NM\text{-}SDA}}$ and $\mathbf{H}_{b,i-1}$. Both generate the public key in the same fashion. As to the challenge ciphertext, the first $i-1$ ciphertext components e_j generated by $\mathbf{C}_{b,i}\mathbf{G}_{p,0}^{\Pi,\mathsf{NM\text{-}SDA}}$ are encryptions of random bits $\tilde{c}[j]$, whereas the i^{th} and the remaining components are encryptions of the corresponding bits of an encoding of m_b (generated by $\mathbf{G}_{p,0}^{\Pi,\mathsf{NM\text{-}SDA}}$ and $\mathbf{C}_{b,i}$, respectively). The same is true for $\mathbf{H}_{b,i-1}$. The answer to a decryption query component sent to $\mathbf{C}_{b,i}\mathbf{G}_{p,0}^{\Pi,\mathsf{NM\text{-}SDA}}$ is $\mathrm{Dec}(c')$ for $c' = (c'[1],\ldots,c'[n])$, where $c'[j] = D_{\mathsf{sk}_j}(e'_j)$ unless $j < i$ and $e'_j = e_j$, in which case $c'[j] = \tilde{c}[j]$. Again, the same holds for $\mathbf{H}_{b,i-1}$. Moreover, both $\mathbf{C}_{b,i}\mathbf{G}_{p,0}^{\Pi,\mathsf{NM\text{-}SDA}}$ and $\mathbf{H}_{b,i-1}$ answer test if $\mathrm{Dec}(c') \in \{m_0, m_1\}$. Thus, they behave identically.

$\mathbf{C}_{b,i}\mathbf{G}_{p,1}^{\Pi,\mathsf{NM\text{-}SDA}}$ and $\mathbf{H}_{b,i}$ are compared similarly. This concludes the proof. □

Lemma 5.5. *For $b \in \{0,1\}$, there exists a wrapper $\mathbf{W}_b$ such that*

1. *$\mathbf{W}_b\mathbf{R}^{\mathsf{copy}}$ behaves as $\mathbf{H}_{b,n}$, and*
2. *$\mathbf{W}_0\mathbf{S}^{\mathsf{trivial}}$ and $\mathbf{W}_1\mathbf{S}^{\mathsf{trivial}}$ behave identically.*

Proof. Wrapper $\mathbf{W}_b$ works as follows: Initially, it generates n key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$ for $i \in [n]$ and outputs $\mathsf{pk} := (\mathsf{pk}_1,\ldots,\mathsf{pk}_n)$. When it receives $(\mathsf{challenge}, m_0)$, it picks n random values $\tilde{c}[1],\ldots,\tilde{c}[n]$, computes $e_i^* \leftarrow E_{\mathsf{pk}}(\tilde{c}[i])$ for $i = 1,\ldots,n$, and returns $e = (e_1,\ldots,e_n)$. Additionally, it outputs m_b to its inside A-interface.

When it gets a (parallel) decryption query, for every component $e' = (e'_1,\ldots,e'_n)$, it proceeds as follows: First, it creates a tamper query $f = (f[1],\ldots,f[n])$ where

$$f[i] \quad = \quad \begin{cases} \mathsf{zero} & \text{if } e'_i \neq e_i^* \text{ and } D_{\mathsf{sk}_i}(e'_i) = 0, \\ \mathsf{one} & \text{if } e'_i \neq e_i^* \text{ and } D_{\mathsf{sk}_i}(e'_i) = 1, \text{ and} \\ \mathsf{keep} & \text{if } e'_i = e_i^*. \end{cases}$$

Then, it outputs (tamper, f) to the inside E-interface and obtains an answer m' at the inside B-interface. If $m' \in \{m_0, m_1\}$, the answer to the component query test.[7] Otherwise, it is m'. If one of the component answers is $\perp$, $\mathbf{W}_b$ implements the self-destruct mode, i.e., it halts.

[7] Again, $\mathbf{W}_b$ needs to output a single parallel tamper query containing the tamper functions f for the individual components.

Consider $\mathbf{W}_b\mathbf{R}^{\mathsf{copy}}$ and $\mathbf{H}_{b,n}$. Both generate the public key in the same fashion. Furthermore, in either case, the challenge ciphertext consists of n encryptions of random bits. Finally, both answer a decryption query by applying the same tamper function to an encoding of m_b before decoding it. When the decoding of the tampered codeword results in m_0 or m_1, both answer test. Therefore, they behave identically.

Due to the fact that test is output when a decryption query results in m_0 or m_1, the observable behavior is the same in $\mathbf{W}_0\mathbf{S}^{\mathsf{trivial}}$ and $\mathbf{W}_1\mathbf{S}^{\mathsf{trivial}}$.[8] □

Proof (of Lemma 5.3). Lemma 5.3 follows using a triangle inequality:

$$\begin{aligned}
&\Delta^{\mathbf{D}}(\mathbf{G}_{p,0}^{\Pi',\mathsf{NM\text{-}RSDA}}, \mathbf{G}_{p,1}^{\Pi',\mathsf{NM\text{-}RSDA}}) \\
&\quad\leq \sum_i \Delta^{\mathbf{D}}(\mathbf{H}_{0,i-1}, \mathbf{H}_{0,i}) + \Delta^{\mathbf{D}}(\mathbf{W}_0\mathbf{R}^{\mathsf{copy}}, \mathbf{W}_0\mathbf{S}^{\mathsf{trivial}}) \\
&\qquad + \Delta^{\mathbf{D}}(\mathbf{W}_1\mathbf{S}^{\mathsf{trivial}}, \mathbf{W}_1\mathbf{R}^{\mathsf{copy}}) + \sum_i \Delta^{\mathbf{D}}(\mathbf{H}_{1,i-1}, \mathbf{H}_{1,i}) \\
&\quad\leq \sum_{b,i} \Delta^{\mathbf{D}}(\mathbf{C}_{b,i}\mathbf{G}_{p,0}^{\Pi,\mathsf{NM\text{-}SDA}}, \mathbf{C}_{b,i}\mathbf{G}_{p,1}^{\Pi,\mathsf{NM\text{-}SDA}}) \\
&\qquad + \sum_b \Delta^{\mathbf{D}\mathbf{W}_b}(\mathbf{R}^{\mathsf{copy}}, \mathbf{S}^{\mathsf{trivial}})
\end{aligned}$$

for any distinguisher $\mathbf{D}$. □

5.4 Efficiency of the Transformations

This section compares the efficiency of the PKE schemes resulting from the EtEb paradigm to the efficiency of domain-extension transformations for *full* IND-CCA security and to the straight-forward approach via IND-CPA security and the CDMW construction (cf. Section 4.4).

5.4.1 Comparison to Full-CCA Transformations

The work of Hohenberger *et al.* [HLW12]—building on the work of Myers and shelat [MS09]—describes a multi-bit IND-CCA-secure (CCA for short) encryption scheme from a single-bit CCA-secure one, an IND-CPA-secure (CPA) one, and a 1-query-bounded CCA-secure one. Their scheme is rather sophisticated and has a somewhat circular structure, requiring a complex security proof. The public key is of the form $\mathsf{pk} = (\mathsf{pk}_{in}, \mathsf{pk}_A, \mathsf{pk}_B)$, where

[8] This is where the proof reflects that Π' is only NM-RSDA secure.

the "inner" public key pk_{in} is the public key of a so-called DCCA secure PKE scheme, and the "outer" public keys pk_A and pk_B are, respectively, the public key of a 1-bounded CCA and a CPA secure PKE scheme. To encrypt a k-bit message m, one first encrypts a tuple (r_A, r_B, m), using the "inner" public key, obtaining a ciphertext e_{in}, where r_A and r_B are thought of as being the randomness for the "outer" encryption schemes. Next, one has to encrypt e_{in} under the "outer" public key pk_A (resp. pk_B) using randomness r_A (resp. r_B) and thus obtaining a ciphertext e_A (resp. e_B). The output ciphertext is $e = (e_A, e_B)$.

To use the above scheme, one has to instantiate the DCCA, 1-bounded CCA and CPA components. As argued in [HLW12], all schemes can be instantiated using a single-bit CCA-secure PKE scheme yielding a fully black-box construction of a multi-bit CCA-secure PKE from a single-bit CCA-secure PKE. Denote by l_p (resp., l_e) the bit-length of the public key (resp., the ciphertext) for the single-bit CCA-secure PKE scheme. If one refers to the construction of [CHH+07] for the 1-bounded CCA component, one obtains a public key of size roughly $(3 + 16s) \cdot l_p$ for the public key and $(k + 2s) \cdot 4s \cdot l_e^2$ for the ciphertext, for security parameter s.[9]

In contrast, a scheme based on the EtEb paradigm instantiated with an NMC of rate ρ has ciphertexts and public keys of length $\rho^{-1}k \cdot l_e$ and $\rho^{-1}k \cdot l_p$, respectively.

5.4.2 Comparison to the CDMW Construction

Any single-bit IND-SDA, NM-CPA, or NM-SDA PKE scheme is (of course) IND-CPA-secure. It follows from a straight-forward, standard hybrid argument that the parallel repetition (i.e., encrypting each bit of a k-bit message individually—even under a single public key) is IND-CPA-secure. Therefore, to obtain an NM-SDA-secure PKE scheme, the CDMW construction can be used. If a rate-ρ LECSS is used, the resulting PKE scheme has ciphertext size $\rho^{-1}k^3 \cdot l_e$ and public-key size $\rho^{-1}k^2 \cdot l_e$.

[9] For simplicity, it is assumed here that the random strings r_A, r_B are computed by stretching the seed (of length s) of a pseudo-random generator.

Chapter 6

Non-Malleability against Bit-Wise Tampering

This section provides non-malleable codes (NMCs) resilient against bit-wise tampering. These codes can be used in the domain-extension constructions presented in Chapter 5.

The first code, in Section 6.1, is an NMC resilient against simple tampering (cf. Section 5.1.1) and is used for the case of IND-SDA security (cf. Section 5.2). Moreover, Section 6.2 shows that any NMC resilient against bit-wise tampering is also secure w.r.t. multiple encodings (cf. Section 5.1.2). Section 6.1.3 shows that the self-destruct mode is necessary.

The second code, in Section 6.3, is a *secret-state* NMC secure against parallel tampering (cf. Section 5.1.3) and is used for the cases of NM-CPA and NM-SDA (cf. Section 5.3). Section 6.3.3 shows that the secret state is necessary to obtain security against parallel tampering.

This chapter uses notation established in Chapter 5.1.

6.1 Simple Tampering

It turns out that a code construction by Dziembowski *et al.* [DPW10], which combines an AMD code and a LEDSS (cf. Section 2.9), already is a (continuously) non-malleable code resilient against tampering from the class $\mathcal{F}_{\mathsf{bit}}$.[1] This fact is proved in this section by first showing that the

[1] Dziembowski *et al.* [DPW10] only proved *basic* non-malleability of said code.

LEDSS is a non-malleable reduction from $\mathcal{F}_{\mathsf{bit}}$ to an intermediate class $\mathcal{F}_{\mathsf{xor}}$ (defined below) and then that the AMD code is a non-malleable reduction from $\mathcal{F}_{\mathsf{xor}}$ to $\mathcal{F}_{\mathsf{triv}}$. Hence, combining the two yields a non-malleable code against $\mathcal{F}_{\mathsf{bit}}$.

6.1.1 From Bit-Wise Tampering to Algebraic Manipulation

Consider the class $\mathcal{F}_{\mathsf{xor}}$ that consists of all functions $f : \{0,1\}^n \to \{0,1\}^n$ of the following types:

- *Algebraic functions:* $f(x) = x + \Delta$ for some $\Delta \in \{0,1\}^n$; such functions are denoted by "$\oplus\Delta$".
- *Constant functions:* $f(x) = z$ for some $z \in \{0,1\}^n$; such functions are denoted by "$\circledcirc z$".

A LEDSS with sufficiently high minimum distance constitutes a non-malleable reduction from $\mathcal{F}_{\mathsf{bit}}$ to $\mathcal{F}_{\mathsf{xor}}$:

Theorem 6.1. *Let* $\mathrm{CS} = (\mathsf{E}, \mathsf{D})$ *be a* (k, n, δ, τ)*-LEDSS with* $\delta > 1/4$ *and* $\delta \geq \tau$.[2] *Then,* CS *is an* $(\mathcal{F}_{\mathsf{bit}}, \mathcal{F}_{\mathsf{xor}}, \varepsilon)$*-non-malleable reduction for*

$$\varepsilon = 2^{-(\tau n - 1)} + \left(\frac{\tau}{(\delta - 1/4)^2}\right)^{\tau n/2}.$$

Security Proof

The reader is referred to Sections 5.1.1 and 5.1.4 for security definitions of non-malleable codes and notation surrounding bit-wise tampering, respectively.

Let $\mathbf{R}^{\mathsf{bit}} := \mathbf{R}^{\mathcal{F}_{\mathsf{bit}},\mathrm{CS}}$ and $\mathbf{S}^{\mathsf{xor}} := \mathbf{S}^{\mathcal{F}_{\mathsf{xor}},\sigma}$ be as in Definition 5.2 (for a simulator σ to be determined) and fix some distinguisher $\mathbf{D}$. The theorem is proved conditioned on the message m encoded by $\mathbf{D}$.

In the following, queries $f \in \mathcal{F}_{\mathsf{bit}}$ with $0 \leq a(f) \leq \tau n$, $\tau n < a(f) < (1-\tau)n$, and $(1-\tau)n \leq a(f) \leq n$ are called *low queries*, *middle queries*, and *high queries*, respectively.

On a high level, the proof proceeds as follows: First, one shows that *middle* queries are rejected with high probability. Then, one proves that issuing *low* and *high* queries actually corresponds to guessing bits of the encoding that is being tampered with. Using the secrecy property of the

[2] Note that the requirement $\delta \geq \tau$ can always be achieved by "ignoring" some of the secrecy.

LEDSS, one can show that only with negligible probability, some attacker can guess sufficiently many of those bits before the self-destruct in order to be able to distinguish tampering with an actual encoding from tampering with uniformly random bits, which leads to a simulation strategy.

Analyzing low and high queries. Consider the system $\mathbf{R}^{\mathsf{bit}}$ and let $c = c[1] \cdots c[n] = \mathsf{E}(m; r)$ be the encoding of the message m initially specified by $\mathbf{D}$, where r are the random bits used by E. Moreover, for a query f, let $\tilde{c} = \tilde{c}[1] \cdots \tilde{c}[n] = f(\mathsf{E}(m; r))$ be the tampered encoding. By the linearity of the LEDSS,

$$\mathsf{D}(\tilde{c}) = \mathsf{D}(c) + \mathsf{D}(d),$$

where $d = \tilde{c} - c$.

- Consider a *low query* f. It fully determines the bits $i \in B(f)$ of d; namely, $d[i] = \mathsf{val}(f[i])$. Let d^* be a codeword such that $d^*[i] = \mathsf{val}(f[i])$ for all $i \in B(f)$. Due to the fact that the LEDSS has distance $\delta \geq \tau$ and $|B(f)| \geq (1 - \tau)n$, d^* is unique (and determined solely by f).

  Therefore, $\mathsf{D}(\tilde{c}) \neq \perp$ if and only if for all $i \in A(f)$, $d[i] = d^*[i]$ or, equivalently, $\mathsf{val}(f[i]) - c[i] = d^*[i]$.

- Consider a *high query* f. It fully determines the bits $i \in A(f)$ of $\tilde{c}$; namely, $\tilde{c}[i] = \mathsf{val}(f[i])$. Let $\tilde{c}^*$ be a codeword such that $\tilde{c}^*[i] = \mathsf{val}(f[i])$ for all $i \in A(f)$. Due to the fact that the LEDSS has distance $\delta \geq \tau$ and $|A(f)| \geq (1 - \tau)n$, $\tilde{c}^*$ is unique (and determined solely by f).

  Therefore, $\mathsf{D}(\tilde{c}) \neq \perp$ if and only if for all $i \in B(f)$, $\tilde{c}[i] = \tilde{c}^*[i]$ or, equivalently, $c[i] + \mathsf{val}(f[i]) = \tilde{c}^*[i]$.

Handling middle queries. Consider the hybrid system $\mathbf{H}$ that proceeds as $\mathbf{R}^{\mathsf{bit}}$ except that as soon as $\mathbf{D}$ specifies a middle query, it outputs $\perp$ and self-destructs.

Lemma 6.2. $\Delta^{\mathbf{D}}(\mathbf{R}^{\mathsf{bit}}, \mathbf{H}) \leq 2^{-\tau n} + \left(\frac{\tau}{(\delta - 1/4)^2}\right)^{\tau n/2}$.

The proof of Lemma 6.2 follows a generic paradigm, at whose core is the so-called *self-destruct lemma*, which deals with the indistinguishability of hybrids with the self-destruct property and is explained in detail in Section 7. Roughly, this lemma applies whenever the first hybrid (in this

case $\mathbf{R}^{\mathsf{bit}}$) can be turned into the second one (in this case $\mathbf{H}$) by changing ("bending") the answers to a subset (the "bending set") of the possible queries to always be $\perp$, and when additionally non-bent queries have a unique answer (cf. the statement of Lemma 7.1). Intuitively, the lemma states that adaptivity does not help distinguish in such cases.[3]

Proof. To use the self-destruct lemma, note that both $\mathbf{R}^{\mathsf{bit}}$ and $\mathbf{H}$ answer queries from $\mathcal{X} := \mathcal{F}$ by values from $\mathcal{Y} := \{0,1\}^k \cup \{\perp\}$. Moreover, note that their internal randomness is an element uniformly chosen from the space $\mathcal{R}$ of random strings r for the encoding algorithm E.

Let $g : \mathcal{X} \times \mathcal{R} \to \mathcal{Y}$ be the function according to which $\mathbf{R}^{\mathsf{bit}}$ answers queries, i.e.,

$$g(f, r) := \mathsf{D}(f(\mathsf{E}(m; r))).$$

Hence, $\mathbf{R}^{\mathsf{bit}}$ is a PSSD game and $\mathbf{H}$ is its $\mathcal{B}$-bending (cf. Definition 7.2) where $\mathcal{B} \subseteq \mathcal{F}_{\mathsf{bit}}$ is the set of middle queries.

Moreover, given the above it is easy to see that queries $f \notin \mathcal{B}$, i.e., low and high queries, can only be answered by a unique value y_f or $\perp$. For

- *low queries* that value is $y_f := m + \mathsf{D}(d^*)$ and for
- *high queries* that value is $y_f := \mathsf{D}(\tilde{c}^*)$.

Finally, note that by the original analysis of middle queries f in [DPW10],

$$\mathsf{P}[\mathsf{D}(f(\mathsf{E}(m; r))) \neq \perp] \leq \left(\frac{\tau}{(\delta - 1/4)^2}\right)^{\tau n/2}.$$

□

Bit-guessing. Consider the hybrid system $\mathbf{H}$. Making tamper queries to this system essentially amounts to trying to "guess" the bits of the encoding $\mathsf{E}(m)$ with the caveat that an incorrect guess leads to the self-destruct. This intuition is formalized by defining a system $\mathbf{B}$ capturing the bit-guessing and a wrapper system $\mathbf{W}$ such that $\mathbf{WB} \equiv \mathbf{H}$.

System $\mathbf{B}$ works as follows: Initially, it takes a value $m \in \{0,1\}^k$, computes an encoding $c[1] \cdots c[n] \leftarrow \mathsf{E}(m)$ of it, and outputs λ (where the symbol λ indicates an empty output). Then, it repeatedly accepts guesses $g_i = (j, b)$, where (j, b) is a guess b for c_j. If a guess g_i is correct, $\mathbf{B}$ returns $a_i = 1$. Otherwise, it outputs $a_i = \perp$ and self-destructs (i.e., all future answers are $\perp$).

[3] Note that Lemma 7.1 is state for games that accept *parallel* queries. This is not needed here, i.e., $p = 1$ in the statement of the lemma.

Wrapper System **W**

```
init
  ∀i ∈ [n] : c[i] ← ∅

on first m at A
  output m at in

on (tamper, f) with 0 ≤ a(f) ≤ τn at E
  if
  ∃codeword d*: ∀i ∈ B(f) : val(f[i]) = d*[i]
    for i where f[i] ∈ A(f)
      g ← val(f[i]) − d*[i]
      if c[i] = ∅
        output (i, g) at in
        get a ∈ {⊥, 1} at in
        if a = ⊥
          self-destruct
        c[i] ← g
      else
        if c[i] ≠ g
          self-destruct
    if D(d*) = ⊥
      self-destruct
    else
      output m + D(d*) at out
  else
    self-destruct

on (tamper, f) with τn < a(f) < n − τn at out
  self-destruct

on (tamper, f) with n − τn ≤ a(f) ≤ n at out
  if ∃codeword c̃* : ∀i ∈ A(f) : val(f[i]) = c̃*[i]
    for i where f[i] ∈ B(f)
      g ← c̃*[i] − val(f[i])
      if c[i] = ∅
        output (i, g) at in
        get a ∈ {⊥, 1} at in
        if a = ⊥
          self-destruct
        c[i] ← g
      else
        if c[i] ≠ g
          self-destruct
    if D(c̃*) = ⊥
      self-destruct
    else
      output D(c̃*) at out
  else
    self-destruct
```

Figure 6.1: *The wrapper system* **W**. *The command* **self-destruct** *causes* **W** *to output* $\perp$ *at* B *and to halt.*

The wrapper system **W** (cf. Figure 6.1) initially forwards the message m the distinguisher wishes to encode to **B**, which internally creates an encoding $c[1] \cdots c[n]$ of m. Then, **W** follows the intuition already built above:

- A *low* query f results in $m + \mathsf{D}(d^*)$ if $c[i] = \mathsf{val}(f[i]) - d^*[i]$ for all $i \in A(f)$.
- A *middle* query f results in $\perp$.
- A *high* query f results in $\mathsf{D}(\tilde{c}^*)$ if $c[i] = \tilde{c}^*[i] - \mathsf{val}(f[i])$ for all $i \in B(f)$.

Hence, upon receiving a low or a high query, **W** issues the corresponding guesses to **B**. If all guesses succeed, **W** outputs $m + \mathsf{D}(d^*)$ resp. $\mathsf{D}(\tilde{c}^*)$. Otherwise, it outputs $\perp$ and self-destructs.

Lemma 6.3. $\mathbf{WB} \equiv \mathbf{H}$.

Proof. By inspection and the above arguments. □

Simulation. Consider the system $\mathbf{B}'$ behaves as $\mathbf{B}$ except that the initial input m is ignored and the $c_1, \ldots, c_n$ are chosen uniformly at random and independently.

Lemma 6.4. $\Delta^{\mathbf{D}}(\mathbf{B}, \mathbf{B}') \leq 2^{-\tau n}$.

Proof. The behavior of $\mathbf{B}$ (and similarly that of $\mathbf{B}'$) is described by a sequence $(\mathsf{p}^{\mathbf{B}}_{A^i|G^i})_{i\geq 0}$ of conditional probability distributions (cf. Section 2.3), where $\mathsf{p}^{\mathbf{B}}_{A^i|G^i}(a^i, g^i)$ is the probability of observing the outputs $a^i = (\lambda, a_1, \ldots, a_i)$ given the inputs $g^i = (m, g_1, \ldots, g_i)$. For simplicity, assume that g^i is such that no position is guessed twice (a generalization is straightforward) and that a^i is of the form $\{\lambda\}\{1\}^*\{\perp\}^*$ (as otherwise it has probability 0 anyway).

For system $\mathbf{B}$, all i, and any g^i, $\mathsf{p}^{\mathbf{B}}_{A^i|G^i}(a^i, g^i) = 2^{-(s+1)}$ if a^i has $s < \min(i, \tau n)$ leading 1's; this follows from the τn-wise independence of the bits of $\mathsf{E}(m)$. All remaining output vectors a^i, i.e., those with at least $\min(i, \tau n)$ preceding 1's, share a probability mass of $2^{-\min(i,\tau n)}$, in a way that depends on the code in use and on m. (It is easily verified that this yields a valid probability distribution.) The behavior of $\mathbf{B}'$ is obvious given the above (simply replace "τn" by "n" in the above description).

On both systems $\mathbf{B}$ and $\mathbf{B}'$, one can define an MBO $\mathcal{B}$ (cf. Section 2.3) that is zero as long as *less* than τn positions have been guessed correctly. In the following, $\hat{\mathbf{B}}$ and $\hat{\mathbf{B}}'$ denote $\mathbf{B}$ and $\mathbf{B}'$ with the MBO, respectively.

Analogously to the above, the behavior of $\hat{\mathbf{B}}$ (and similarly that of $\hat{\mathbf{B}}'$) is described by a sequence $(\mathsf{p}^{\hat{\mathbf{B}}}_{A^i,B_i=0|G^i})_{i\geq 0}$ of conditional probability distributions, where $\mathsf{p}^{\hat{\mathbf{B}}}_{A^i,B_i=0|G^i}(a^i, g^i)$ is the probability of observing the outputs $a^i = (\lambda, a_1, \ldots, a_i)$ and $b_0 = b_1 = \ldots = b_i = 0$ given the inputs $g^i = (m, g_1, \ldots, g_i)$. One observes that due to the τn-wise independence of $\mathsf{E}(m)$'s bits, for $i < \tau n$,

$$\mathsf{p}^{\hat{\mathbf{B}}}_{A^i,B_i=0|G^i}(a^i, g^i) = \mathsf{p}^{\hat{\mathbf{B}}'}_{A^i,B_i=0|G^i}(a^i, g^i)$$
$$= \begin{cases} 2^{-(s+1)} & \text{if } a^i \text{ has } s < i \text{ leading 1's,} \\ 2^{-i} & \text{if } a^i \text{ has } i \text{ leading 1's, and} \\ 0 & \text{otherwise,} \end{cases}$$

Simulator σ

init
  $\forall i \in [n] : c[i] \leftarrow \{0,1\}$

on (tamper, f) *with* $0 \le a(f) \le \tau n$ *at* out
  for i *where* $f[i] \in A(f)$
    $d'[i] \leftarrow \mathsf{val}(f[i]) \oplus c[i]$
  for i *where* $f[i] \in B(f)$
    $d'[i] \leftarrow \mathsf{val}(f[i])$
  $d' \leftarrow d'[1] \cdots d'[n]$
  if $\mathsf{D}(d') = \perp$
    output (tamper, $\perp$) at in
  else
    output $\oplus\mathsf{D}(d')$ at in

on (tamper, f) *with* $\tau n < a(f) < (1-\tau)n$ *at* out
  output (tamper, $\perp$) at in

on (tamper, f) *with* $(1-\tau)n \le a(f) \le n$ *at* out
  for i *where* $f[i] \in A(f)$
    $c'[i] \leftarrow \mathsf{val}(f[i])$
  for i *where* $f[i] \in B(f)$
    $c'[i] \leftarrow c[i] \oplus \mathsf{val}(f[i])$
  $c' \leftarrow c'[1] \cdots c'[n]$
  output (tamper, $\circledcirc\mathsf{D}(c')$) at in

Figure 6.2: *The simulator σ.*

and for $i \ge \tau n$,

$$\mathsf{p}^{\hat{\mathbf{B}}}_{A^i, B_i=0 | G^i}(a^i, g^i) = \mathsf{p}^{\hat{\mathbf{B}}'}_{A^i, B_i=0 | G^i}(a^i, g^i) = \begin{cases} 2^{-(s+1)} & \text{if } a^i \text{ has } s < \tau n \text{ leading 1's,} \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, $\hat{\mathbf{B}} \overset{g}{\equiv} \hat{\mathbf{B}}'$ and $\Delta^{\mathbf{D}}(\mathbf{B}, \mathbf{B}') \le \Gamma^{\mathbf{D}}(\hat{\mathbf{B}}')$. Observe that by an argument similar to the one above, adaptivity does not help in provoking the MBO of $\hat{\mathbf{B}}'$. Thus, $\Gamma^{\mathbf{D}}(\hat{\mathbf{B}}') \le 2^{-\tau n}$, since an optimal non-adaptive strategy simply tries to guess distinct positions. □

Consider now the system $\mathbf{WB}'$. Due to the nature of $\mathbf{B}'$, the behavior of $\mathbf{WB}'$ is independent of the value m that is initially encoded. This allows to easily design a simulator σ as such that $\mathbf{WB}' \equiv \mathbf{S}^{\mathsf{xor}}$. It internally creates a simulated encoding consisting of uniformly random bits (just as $\mathbf{WB}'$) and then follows the intuition above. The simulator is described in Figure 6.2. By inspection, one easily verifies:

Lemma 6.5. $\mathbf{WB}' \equiv \mathbf{S}^{\mathsf{xor}}$.

The proof of Theorem 6.1 now follows from a simple triangle inequality.

Proof (of Theorem 6.1). From Lemmas 6.2-6.5, one obtains that for all

distinguishers $\mathbf{D}$,

$$\begin{aligned}
&\Delta^{\mathbf{D}}(\mathbf{R}^{\mathsf{bit}}, \mathbf{S}^{\mathsf{xor}}) \\
&\leq \Delta^{\mathbf{D}}(\mathbf{R}^{\mathsf{bit}}, \mathbf{H}) + \underbrace{\Delta^{\mathbf{D}}(\mathbf{H}, \mathbf{WB})}_{=0} + \underbrace{\Delta^{\mathbf{D}}(\mathbf{WB}, \mathbf{WB}')}_{=\Delta^{\mathbf{DW}}(\mathbf{B},\mathbf{B}')} + \underbrace{\Delta^{\mathbf{D}}(\mathbf{WB}', \mathbf{S}^{\mathsf{xor}})}_{=0} \\
&\leq 2^{-\tau n} + \left(\frac{\tau}{(\delta - 1/4)^2}\right)^{\tau n/2} + 2^{-\tau n} \\
&\leq 2^{-(\tau n - 1)} + \left(\frac{\tau}{(\delta - 1/4)^2}\right)^{\tau n/2}.
\end{aligned}$$

□

Security against $\mathcal{F}_{\mathsf{copy}}$. By inspecting the above proof, one can easily see that if only functions from $\mathcal{F}_{\mathsf{copy}} \subseteq \mathcal{F}_{\mathsf{bit}}$ are used, no queries $\oplus\Delta$ for $\Delta \neq 0^n$ are issued by σ. This implies that the LEDSS is a non-malleable code resilient against $\mathcal{F}_{\mathsf{copy}}$.

Theorem 6.6. *Let* $\mathrm{CS} = (\mathsf{E}, \mathsf{D})$ *be a* (k, n, δ, τ)*-LEDSS with* $\delta > 1/4$ *and* $\delta \geq \tau$*. Then,* CS *is an* $(\mathcal{F}_{\mathsf{bit}}, \varepsilon)$*-non-malleable code for*

$$\varepsilon = 2^{-(\tau n - 1)} + \left(\frac{\tau}{(\delta - 1/4)^2}\right)^{\tau n/2}.$$

6.1.2 From Algebraic Manipulation to Non-Malleability

An AMD code already constitutes a non-malleable reduction from $\mathcal{F}_{\mathsf{xor}}$ to $\mathcal{F}_{\mathsf{triv}}$:

Theorem 6.7. *Let* $\mathrm{AMD} = (\mathsf{A}, \mathsf{V})$ *be a* ρ*-AMD code. Then,* AMD *is an* $(\mathcal{F}_{\mathsf{xor}}, \mathcal{F}_{\mathsf{triv}}, \rho)$*-non-malleable reduction.*

Proof. Consider the following simulator σ: Upon receiving (tamper, f) for $f \in \mathcal{F}_{\mathsf{xor}}$ at the outside interface, it outputs (tamper, f) at the inside interface if $f = \circledcirc z$ for some $z \in \{0,1\}^n$ or if $f = \oplus 0^n$. Otherwise, it outputs $(\mathsf{tamper}, \perp)$ at the inside interface.

Consider $\mathbf{R}^{\mathsf{xor}} := \mathbf{R}^{\mathcal{F}_{\mathsf{xor}}, \mathrm{AMD}}$ and $\mathbf{S}^{\mathsf{trivial}} := \mathbf{S}^{\mathcal{F}_{\mathsf{triv}}, \sigma}$. These systems only differ in behavior if the first query $(\mathsf{tamper}, \oplus\Delta)$ with $\Delta \neq 0^n$ does not trigger the self-destruct. Since all other queries (replacing by a constant or the identity) clearly do not reveal any information on the encoding, the security of the AMD code guarantees that this happens with probability at most ρ. □

6.1.3 On the Necessity of Self-Destruct

No (k, n)-coding scheme (Enc, Dec) is (continuously) non-malleable against $\mathcal{F}_{\mathsf{copy}}$ without self-destruct. This fact is reminiscent of the negative result by Gennaro *et al.* [GLM+04], and was already observed by Faust *et al.* [FMNV14] (without a proof) for the easier case of so-called *strong* continuous non-malleability. The distinguisher $\mathbf{D}$ provided by Theorem 6.8 is universal, i.e., it breaks any coding scheme (if given oracle access to its decoding algorithm).

For the remainder of this section, let $\mathbf{R}^{\mathsf{copy}} := \mathbf{R}^{\mathcal{F}_{\mathsf{copy}},\mathrm{CS}}$ and $\mathbf{S}^{\mathsf{trivial}} := \mathbf{S}^{\mathcal{F}_{\mathsf{triv}},\sigma}$ for an *arbitrary* simulator σ. However, both $\mathbf{R}^{\mathsf{copy}}$ and $\mathbf{S}^{\mathsf{trivial}}$ are stripped of the self-destruct mode.

Theorem 6.8. *There exists a distinguisher* $\mathbf{D}$ *such that for all coding schemes* $\mathrm{CS} = (\mathrm{Enc}, \mathrm{Dec})$ *and all simulators* σ*,*

$$\Delta^{\mathbf{D}}(\mathbf{R}^{\mathsf{copy}}, \mathbf{S}^{\mathsf{trivial}}) \;\geq\; 1 - \frac{n+1}{2^k}.$$

Proof. Distinguisher $\mathbf{D} := \mathbf{D}_{\mathsf{Ext}}$ uses an algorithm Ext that always extracts the encoded message when interacting with system $\mathbf{R}^{\mathsf{copy}}$ and does so with small probability only when interacting with system $\mathbf{S}^{\mathsf{trivial}}$.

The extraction algorithm. Consider the following algorithm, denoted Ext, which repeatedly issues tamper queries (tamper, f) with $f \in \mathcal{F}_{\mathsf{copy}}$, expects an answer in $\{0,1\}^k \cup \{\bot, \mathsf{same}\}$, and eventually outputs a value $m' \in \{0,1\}^k$: Initially, it initializes variables $f[1], \ldots, f[n] \leftarrow \varnothing$ (where the value $\varnothing$ stands for "undefined"). Then, for $i = 1, \ldots, n$ it proceeds as follows: It queries (tamper, f) with $f = (f[1], \ldots, f[i-1], \mathsf{zero}, \mathsf{keep}, \ldots, \mathsf{keep})$. If the answer is same, it sets $f[i] \leftarrow \mathsf{zero}$ and otherwise $f[i] \leftarrow \mathsf{one}$. In the end Ext outputs $m' \leftarrow \mathrm{Dec}(\mathsf{val}(f[1]) \cdots \mathsf{val}(f[n]))$.

The distinguisher. Consider the following distinguisher $\mathbf{D}_{\mathsf{Ext}}$: Initially, it chooses $m \leftarrow \{0,1\}^k$ and outputs m to the A-interface of the system it is connected to. Then, it lets Ext interact with that system, forwarding the tamper queries to the E-interface and the answers from the B-interface, replacing an answer by same whenever it is m. When Ext terminates and outputs a value m', $\mathbf{D}_{\mathsf{Ext}}$ outputs 1 if $m' = m$ and 0 otherwise.

Real world. Assume that before the i^{th} iteration of Ext, asking the query (tamper, f) with $f = (f[1], \ldots, f[i-1], \mathsf{keep}, \mathsf{keep}, \ldots, \mathsf{keep})$ to $\mathbf{R}^{\mathsf{copy}}$ yields the answer m. From this it follows that either $(f[1], \ldots, f[i -$

$1], \mathsf{zero}, \mathsf{keep}, \ldots, \mathsf{keep})$ or $(f[1], \ldots, f[i-1], \mathsf{one}, \mathsf{keep}, \ldots, \mathsf{keep})$ leads to the answer m; Ext sets $f[i]$ appropriately (the fact that the answer m is replaced by same plays no role here). Thus, in the end, computing $\mathrm{Dec}(\mathsf{val}(f[1]) \cdots \mathsf{val}(f[n]))$ yields m. Therefore,

$$\mathsf{P}[\mathbf{D}_{\mathsf{Ext}}\mathbf{R}^{\mathsf{copy}} = 1] = 1.$$

Ideal world. Consider the following modified distinguisher $\hat{\mathbf{D}}_{\mathsf{Ext}}$ that works as $\mathbf{D}_{\mathsf{Ext}}$ except that it does *not* modify the answers received by the system it is connected to. Moreover, let $\hat{\mathbf{S}}^{\mathsf{trivial}}$ be the the system that ignores all messages m input at interface A and handles queries (tamper, f) by inputting them to σ at interface $\mathtt{out}$ and outputting σ's answer at interface $\mathtt{in}$, replacing $(\mathsf{tamper}, \mathrm{id})$ by same.

Note that in both experiments, Ext's view is identical unless it causes σ to output m (the value encoded by $\mathbf{D}$), which happens with probability at most $\frac{n}{2^k}$. Thus,

$$|\mathsf{P}^{\mathbf{D}_{\mathsf{Ext}}\mathbf{S}^{\mathsf{trivial}}}[\mathsf{Ext} \text{ outputs } m] - \mathsf{P}^{\hat{\mathbf{D}}_{\mathsf{Ext}}\hat{\mathbf{S}}^{\mathsf{trivial}}}[\mathsf{Ext} \text{ outputs } m]| \leq \frac{n}{2^k}.$$

Furthermore, in experiment $\hat{\mathbf{D}}_{\mathsf{Ext}}\hat{\mathbf{S}}^{\mathsf{trivial}}$, Ext's view is independent of m, and therefore, m is output by Ext with probability $\frac{1}{2^k}$. Hence,

$$\mathsf{P}[\mathbf{D}_{\mathsf{Ext}}\mathbf{S}^{\mathsf{trivial}} = 1] \leq \frac{n+1}{2^k}.$$

This proves the theorem. □

6.2 Achieving Adaptive Non-Malleability

If a coding scheme is non-malleable w.r.t. $\mathcal{F}_{\mathsf{bit}}$ (or $\mathcal{F}_{\mathsf{copy}}$), then it is also *adaptive* non-malleable w.r.t. $\bar{\mathcal{F}}_{\mathsf{bit}}$ (or $\bar{\mathcal{F}}_{\mathsf{copy}}$).

Theorem 6.9. *Let $\varepsilon \geq 0$. If a (k, n)-coding scheme* $(\mathrm{Enc}, \mathrm{Dec})$ *is $(\mathcal{F}_{\mathsf{bit}}, \varepsilon)$-non-malleable (resp. $(\mathcal{F}_{\mathsf{copy}}, \varepsilon)$-non-malleable), it is also $(\bar{\mathcal{F}}_{\mathsf{bit}}, 2\ell\varepsilon + \frac{q\ell}{2^k}, \ell, q)$-adaptive non-malleable (resp. $(\bar{\mathcal{F}}_{\mathsf{copy}}, 2\ell\varepsilon + \frac{q\ell}{2^k}, \ell, q)$-adaptive non-malleable), for all $\ell, q \in \mathbb{N}$.*

Left-or-right non-malleability. The proof of Theorem 6.9 is facilitated by considering the notion of *left-or-right (LOR)* non-malleability, for which the transition from single-encoding to multi-encoding security is less cumbersome than for the standard definition.

System $\mathbf{L}_b^{\bar{\mathcal{F}},\mathrm{CS}}$

init
  $i \leftarrow 0$

on $(\mathsf{lor\text{-}enc}, m_0, m_1)$ *at* A
  $i \leftarrow i+1$
  $m_0^{(i)} \leftarrow m_0$
  $m_1^{(i)} \leftarrow m_1$
  $c^{(i)} \leftarrow \mathrm{Enc}(m_b)$

on (tamper, f) *with* $f \in \mathcal{F}^{(i)}$ *at* E
  $c' \leftarrow f(c^{(1)}, \ldots, c^{(i)})$
  $m' \leftarrow \mathrm{Dec}(c')$
  if $m' = \bot$
    self-destruct
  if $\exists j : m' \in \{m_0^{(j)}, m_1^{(j)}\}$
    $m' \leftarrow \mathrm{id}^{(j)}$
  output m' at B

Figure 6.3: *Systems* $\mathbf{L}_0^{\bar{\mathcal{F}},\mathrm{CS}}$ *and* $\mathbf{L}_1^{\bar{\mathcal{F}},\mathrm{CS}}$ *defining LOR-non-malleability of* $(\mathrm{Enc}, \mathrm{Dec})$. *The* **self-destruct** *command causes the system to output* $\bot$ *at interface* B *and halt.*

In the LOR variant,[4] the A-interface takes as input pairs of messages and encodes either always the first or always the second message. The goal of the attacker is to find out which is the case. Formally, LOR-non-malleability is defined using the two random $\{A, B, E\}$-systems $\mathbf{L}_0^{\bar{\mathcal{F}},\mathrm{CS}}$ and $\mathbf{L}_1^{\bar{\mathcal{F}},\mathrm{CS}}$, shown in Figure 6.3.[5]

When processing a tamper query, if there are multiple indices j for which $\mathrm{id}^{(j)}$ could be output, $\mathbf{L}_b^{\bar{\mathcal{F}},\mathrm{CS}}$ outputs the largest such j.

Definition 6.1. *A coding scheme* $\mathrm{CS} = (\mathrm{Enc}, \mathrm{Dec})$ *is* $(\bar{\mathcal{F}}, \varepsilon, \ell, q)$-adaptive LOR non-malleable *if*

$$\Delta^{\mathbf{D}}(\langle \mathbf{L}_0^{\bar{\mathcal{F}},\mathrm{CS}} \rangle_{\ell,q}, \langle \mathbf{L}_1^{\bar{\mathcal{F}},\mathrm{CS}} \rangle_{\ell,q}) \leq \varepsilon$$

for all distinguishers $\mathbf{D}$, *where* $\langle \cdot \rangle_{\ell,q}$ *denotes that only the first* ℓ *queries at the* A*-interface and only the first* q *queries at the* E*-interface are processed.*

The two definitions are equivalent, however, as shown by Lemmas 6.10 and 6.11.

Lemma 6.10. *If* $\mathrm{CS} = (\mathrm{Enc}, \mathrm{Dec})$ *is* $(\bar{\mathcal{F}}, \varepsilon, \ell, q)$*-adaptive non-malleable, it is also* $(\bar{\mathcal{F}}, 2\varepsilon, \ell, q)$*-adaptive LOR-non-malleable.*

Proof. Fix ℓ, q, and a simulator σ, and let $\mathbf{R} := \langle \bar{\mathbf{R}}^{\bar{\mathcal{F}},\mathrm{CS}} \rangle_{\ell,q}$, $\mathbf{S} := \langle \bar{\mathbf{S}}^{\sigma} \rangle_{\ell,q}$, $\mathbf{L}_0 := \langle \mathbf{L}_0^{\bar{\mathcal{F}},\mathrm{CS}} \rangle_{\ell,q}$, and $\mathbf{L}_1 := \langle \mathbf{L}_1^{\bar{\mathcal{F}},\mathrm{CS}} \rangle_{\ell,q}$. For $b \in \{0,1\}$, consider the following reduction $\mathbf{C}_b$: Upon the i^{th} query $(\mathsf{lor\text{-}enc}, x_0, x_1)$ at the outside

[4] One should not confuse the above LOR variant with *strong* non-malleability, the difference being that for strong non-malleability $\mathbf{L}_b^{\bar{\mathcal{F}},\mathrm{CS}}$ would output $\mathrm{id}^{(j)}$ iff $c' = c^{(j)}$. In fact, being equivalent to non-malleability, our LOR variant is strictly weaker.

[5] The same LOR variant was already considered in [DPW10, Definition A.1] (and referred to as "alternative" non-malleability). In this sense Lemma 6.10 and 6.11 below are a generalization of [DPW10, Theorem A.1] to the adaptive and continuous case.

A-interface, it stores $m_0^{(i)} := x_0$ and $m_1^{(i)} := x_1$ internally and outputs x_b at the inside A-interface. Upon a query (tamper, f) at the outside E-interface, $\mathbf{C}_b$ outputs (tamper, f) at the inside E-interface and subsequently receives a value m' at the inside interface. If there exist indices i' such that $m' \in \{m_0^{(i')}, m_1^{(i')}\}$, $\mathbf{C}_b$ outputs $\mathrm{id}^{(i')}$ for the largest such index at the outside B-interface. Otherwise, it outputs m'.

One observers that

$$\mathbf{C}_0\mathbf{R} \equiv \mathbf{L}_0 \qquad \text{and} \qquad \mathbf{C}_1\mathbf{R} \equiv \mathbf{L}_1 \qquad \text{and} \qquad \mathbf{C}_0\mathbf{S} \equiv \mathbf{C}_1\mathbf{S},$$

where the third equivalence follows from the fact that the observable behavior of $\mathbf{C}_b\mathbf{S}$ is independent of the messages $\mathbf{C}_b$ outputs to $\mathbf{S}$. Hence, for all distinguishers $\mathbf{D}$,

$$\begin{aligned}\Delta^{\mathbf{D}}(\mathbf{L}_0, \mathbf{L}_1) &= \Delta^{\mathbf{D}}(\mathbf{C}_0\mathbf{R}, \mathbf{C}_1\mathbf{R}) \\ &\le \Delta^{\mathbf{D}}(\mathbf{C}_0\mathbf{R}, \mathbf{C}_0\mathbf{S}) + \Delta^{\mathbf{D}}(\mathbf{C}_0\mathbf{S}, \mathbf{C}_1\mathbf{S}) + \Delta^{\mathbf{D}}(\mathbf{C}_1\mathbf{S}, \mathbf{C}_1\mathbf{R}) \\ &\le \Delta^{\mathbf{D}\mathbf{C}_0}(\mathbf{R}, \mathbf{S}) + \Delta^{\mathbf{D}\mathbf{C}_1}(\mathbf{R}, \mathbf{S}) \\ &\le 2\varepsilon.\end{aligned}$$

□

Lemma 6.11. *If* $(\mathrm{Enc}, \mathrm{Dec})$ *is* $(\bar{\mathcal{F}}, \varepsilon, \ell, q)$*-adaptive LOR-non-malleable, it is also* $(\bar{\mathcal{F}}, \varepsilon + \frac{q\ell}{2^k}, \ell, q)$*-adaptive non-malleable.*

Proof. Fix ℓ and q, and let $\mathbf{R} := \langle \bar{\mathbf{R}}^{\bar{\mathcal{F}},\mathrm{CS}} \rangle_{\ell,q}$, $\mathbf{S} := \langle \bar{\mathbf{S}}^{\sigma} \rangle_{\ell,q}$ (for a simulator σ to be defined next), $\mathbf{L}_0 := \langle \mathbf{L}_0^{\bar{\mathcal{F}},\mathrm{CS}} \rangle_{\ell,q}$, and $\mathbf{L}_1 := \langle \mathbf{L}_1^{\bar{\mathcal{F}},\mathrm{CS}} \rangle_{\ell,q}$. Consider the following simulator σ: It internally keeps a counter $i \leftarrow 0$. When invoked on (i', f) with $f \in \mathcal{F}^{(i')}$, if $i' > i$, it samples $m_1^{(j)} \leftarrow \{0,1\}^k \setminus \{m_1^{(1)}, \ldots, m_1^{(j-1)}\}$ and computes $c_1^{(j)} \leftarrow \mathrm{Enc}(m_1^{(j)})$ for all $i < j \le i'$ and sets $i \leftarrow i'$. Then, it computes the tampered codeword $c' \leftarrow \mathrm{Dec}(f(c_1^{(1)}, \ldots, c_1^{(i)}))$ and decodes it to $m' \leftarrow \mathrm{Dec}(c')$. If $m' = m_1^{(j)}$ for some indices j, σ returns (same, j) for the largest such j. Otherwise, it returns m'.

Consider the following reduction $\mathbf{C}$: Upon the i^{th} message m at the outside A-interface, it chooses $m_1^{(i)} \leftarrow \{0,1\}^k \setminus \{m_1^{(1)}, \ldots, m_1^{(i-1)}\}$, stores $m_0^{(i)} := m$ internally, and outputs $(\mathsf{lor\text{-}enc}, m_0^{(i)}, m_1^{(i)})$ at the inside A-interface. Upon a query (tamper, f) at the outside E-interface, $\mathbf{C}$ outputs (tamper, f) at the inside E-interface and subsequently receives a value m' at the inside B-interface. If $m' = \mathrm{id}^{(j)}$ for some j, $\mathbf{C}$ outputs $m_0^{(j)}$ at the outside B-interface. Otherwise, it outputs m'.

Observe that $\mathbf{CL}_1 \equiv \mathbf{S}$. In both cases, the i^{th} message m is treated by sampling fresh values $m_1^{(i)}$ distinct from all $m_1^{(1)}, \ldots, m_1^{(i-1)}$ and computing $c_1^{(i)}$ as an encoding of $m_1^{(i)}$. (This is delayed in $\mathbf{S}$, but that does not change the distribution.) A query (tamper, f) with some function $f \in \mathcal{F}^{(i)}$ is answered by evaluating $f(c_1^{(1)}, \ldots, c_1^{(i)})$, decoding the resulting codeword to obtain a message m', and if $m' = m_1^{(j)}$ for some $j \in \{1, \ldots, i\}$, returning $m_0^{(j)}$ and m' otherwise.

The systems $\mathbf{CL}_0$ and $\mathbf{R}$ are, however, not equivalent. The reason is that if, in $\mathbf{CL}_0$, $\mathrm{Dec}(f(c_0^{(1)}, \ldots, c_0^{(i)})) = m_1^{(j)}$ for some $j \in \{1, \ldots, i\}$, then $\mathbf{L}_0$ returns (same, j), which $\mathbf{C}$ replaces by $m_0^{(j)}$. There is no comparable behavior in $\mathbf{R}$. Provoking this event, however, corresponds to "non-adaptively guessing" one of the values $m_1^{(j)}$, which occurs with probability at most $\frac{i}{2^k}$ in each query.

Hence, for all distinguishers $\mathbf{D}$,

$$\begin{aligned}
\Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{S}) &= \Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{CL}_1) \\
&\leq \Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{CL}_0) + \Delta^{\mathbf{D}}(\mathbf{CL}_0, \mathbf{CL}_1) \\
&\leq \tfrac{q\ell}{2^k} + \Delta^{\mathbf{DC}}(\mathbf{L}_0, \mathbf{L}_1) \\
&\leq \tfrac{q\ell}{2^k} + \varepsilon.
\end{aligned}$$

□

From single to multiple encodings. It remains to show that LOR non-malleability implies adaptive LOR non-malleability.

Lemma 6.12. *If* $(\mathrm{Enc}, \mathrm{Dec})$ *is* $(\bar{\mathcal{F}}_{\mathsf{copy}}, \varepsilon, 1, q)$*-adaptive LOR-non-malleable, it is also* $(\bar{\mathcal{F}}_{\mathsf{copy}}, \ell \cdot \varepsilon, \ell, q)$*-adaptive LOR-non-malleable, for all* $\ell \in \mathbb{N}$.

Proof. Fix ℓ and q, let $\bar{\mathcal{F}} := \bar{\mathcal{F}}_{\mathsf{copy}}$, and set $\mathbf{L}'_b := \langle \mathbf{L}_b^{\bar{\mathcal{F}}, \mathrm{CS}} \rangle_{\ell, q}$ and $\mathbf{L}_b := \langle \mathbf{L}_b^{\bar{\mathcal{F}}, \mathrm{CS}} \rangle_{1, q}$ for $b \in \{0, 1\}$.

The distinguishing advantage between $\mathbf{L}'_0$ and $\mathbf{L}'_1$ is bounded via a hybrid argument, where the i^{th} hybrid $\mathbf{H}^{(i)}$ picks x_0 when processing the first i encode queries $(\mathsf{lor\text{-}enc}, x_0, x_1)$ and x_1 afterwards. For each i, the distinguishing advantage between successive hybrids $\mathbf{H}^{(i-1)}$ and $\mathbf{H}^{(i)}$ is bounded by exhibiting a system $\mathbf{C}_i$ that reduces distinguishing $\mathbf{L}_0$ and $\mathbf{L}_1$ to distinguishing the hybrids.

For $i = 0, 1, \ldots, \ell$, hybrid $\mathbf{H}^{(i)}$ works as follows: Initialization and (tamper, f) are defined as with $\mathbf{L}'_0$ and $\mathbf{L}'_1$. The first i queries $(\mathsf{lor\text{-}enc}, x_0, x_1)$

are handled by encoding x_0, i.e., $c^{(j)} \leftarrow \mathrm{Enc}(x_0)$ for the j^{th} encoding. For all later queries, x_1 is encoded, i.e., $c^{(j)} \leftarrow \mathrm{Enc}(x_1)$.

One observes that

$$\mathbf{H}^{(\ell)} \equiv \mathbf{L}'_0 \qquad \text{and} \qquad \mathbf{H}^{(0)} \equiv \mathbf{L}'_1.$$

For $i = 1, \ldots, n$, reduction $\mathbf{C}_i$ works as follows: For the first $i-1$ encode queries $(\mathsf{lor\text{-}enc}, x_0, x_1)$ (at the outside interface), it computes and stores an encoding of x_0, i.e., $c^{(j)} \leftarrow \mathrm{Enc}(x_0)$ for the j^{th} encoding. Upon the i^{th} query $(\mathsf{lor\text{-}enc}, x_0, x_1)$, it outputs $(\mathsf{lor\text{-}enc}, x_0, x_1)$ at the inside interface. (Note that as a consequence, a target encoding $c \leftarrow \mathrm{Enc}(x_b)$ is generated, depending on whether $\mathbf{C}_i$ is connected to $\mathbf{L}_0$ or $\mathbf{L}_1$.) The remaining encode queries are handled by encoding the second message x_1, i.e., $c^{(j)} \leftarrow \mathrm{Enc}(x_1)$.

System $\mathbf{C}_i$ maintains a counter j that keeps track of the number of encode queries it has encountered. When a tamper query (tamper, f) with $f \in \mathcal{F}^{(j)}_{\mathsf{copy}}$ and $f = (f[1], \ldots, f[n])$ is received at the outside interface, it computes $f[1]', \ldots, f[n]'$, where

$$f[v]' := \begin{cases} f[v] & \text{if } f[v] \in \{\mathsf{zero}, \mathsf{one}\}, \\ \mathsf{zero} & \text{if } f[v] = \mathsf{keep}_w \text{ for } w \neq i, \text{ and } c^{(w)}[v] = 0, \\ \mathsf{one} & \text{if } f[v] = \mathsf{keep}_w \text{ for } w \neq i, \text{ and } c^{(w)}[v] = 1, \\ \mathsf{keep}_1 & \text{if } f[v] = \mathsf{keep}_i. \end{cases}$$

Then, it outputs (tamper, f') at the inside interface, where f' is the function in $\mathcal{F}^{(1)}_{\mathsf{copy}}$ with $f' = (f[1]', \ldots, f[n]')$.[6] Let m' be the answer to the tamper query at the inside interface. $\mathbf{C}_i$ computes the set of indices j for which m' matches one of the two messages of the j^{th} encode query. Moreover, if $m' = \mathsf{same}$, index i is added to that set as well. Then, it outputs $\mathrm{id}^{(j)}$ for the largest index j in the set. If the set is empty, m' is output.

One observes that

$$\mathbf{C}_i\mathbf{L}_0 = \mathbf{H}^{(i)} \qquad \text{and} \qquad \mathbf{C}_i\mathbf{L}_1 = \mathbf{H}^{(i-1)}.$$

[6] For simplicity, we assume here that $\mathbf{L}_0$ and $\mathbf{L}_1$ answer tamper queries consisting of zero and one instructions only even before a message has been encoded.

Thus, for all distinguishers $\mathbf{D}$,

$$\Delta^{\mathbf{D}}(\mathbf{L}'_0, \mathbf{L}'_1) = \Delta^{\mathbf{D}}(\mathbf{H}^{(\ell)}, \mathbf{H}^{(0)}) \leq \sum_{i=1}^{\ell} \Delta^{\mathbf{D}}(\mathbf{H}^{(i)}, \mathbf{H}^{(i-1)})$$
$$\leq \sum_{i=1}^{\ell} \Delta^{\mathbf{D}}(\mathbf{C}_i\mathbf{L}_0, \mathbf{C}_i\mathbf{L}_1) \leq \sum_{i=1}^{\ell} \Delta^{\mathbf{D}\mathbf{C}_i}(\mathbf{L}_0, \mathbf{L}_1) \leq \ell \cdot \varepsilon.$$

□

Proof (of Theorem 6.9). Follows from Lemmas 6.10, 6.11, and 6.12 in a straight-forward manner. □

Lemma 6.13. *If* $(\mathrm{Enc}, \mathrm{Dec})$ *is* $(\bar{\mathcal{F}}_{\mathsf{bit}}, \varepsilon, 1, q)$*-adaptive LOR-non-malleable, it is also* $(\bar{\mathcal{F}}_{\mathsf{bit}}, \ell \cdot \varepsilon, \ell, q)$*-adaptive LOR-non-malleable, for all* $\ell \in \mathbb{N}$*.*

Proof. The proof is analogous to the proof of Lemma 6.12, except that the reduction system $\mathbf{C}_i$ computes $f[v]'$ as follows:

$$f[v]' := \begin{cases} f[v] & \text{if } f[v] \in \{\mathsf{zero}, \mathsf{one}\}, \\ \mathsf{zero} & \text{if } f[v] = \mathsf{keep}_w \text{ for } w \neq i, \text{ and } c^{(w)}[v] = 0, \\ \mathsf{one} & \text{if } f[v] = \mathsf{keep}_w \text{ for } w \neq i, \text{ and } c^{(w)}[v] = 1, \\ \mathsf{keep}_1 & \text{if } f[v] = \mathsf{keep}_i, \\ \mathsf{one} & \text{if } f[v] = \mathsf{flip}_w \text{ for } w \neq i, \text{ and } c^{(w)}[v] = 0, \\ \mathsf{zero} & \text{if } f[v] = \mathsf{flip}_w \text{ for } w \neq i, \text{ and } c^{(w)}[v] = 1, \\ \mathsf{flip}_1 & \text{if } f[v] = \mathsf{flip}_i. \end{cases}$$

□

6.3 Parallel Tampering

6.3.1 A Code Non-Malleable against Parallel Tampering

This section shows a non-malleable code with secret state resilient against parallel tampering. The intuition behind the construction is the following: If a code has the property (as is the case with, e.g., the scheme from Section 6.1.1 secure against non-parallel bit-wise tampering) that changing a single bit of a valid encoding results in an invalid codeword, then the tamper function that fixes a particular bit of the encoding and leaves the remaining positions unchanged can be used to determine the value of that bit (cf. Section 6.1.3); this attack is parallelizable, and thus a code of this

type cannot provide security against parallel tampering. A similar attack is also possible if the code corrects a fixed (known) number of errors. To circumvent this issue, the construction presented here uses a—for the lack of a better word—"dynamic" error-correction bound: The secret state (which is initially chosen at random) is used to determine the positions of the encoding in which (a certain amount of) errors is tolerated.

Construction. Let $\mathbb{F} = \mathrm{GF}(2)$ and $\alpha > 0$. Let (E, D) be a (k, n, δ, τ)-LECSS (cf. Definition 2.5 in Section 2.9) with minimum distance δ and secrecy τ over $\mathbb{F}$ such that:[7]

- *Minimum distance:* $\delta > 1/4 + 2\alpha$ and $\delta/2 > 2\alpha$.
- *Constant rate:* $k/n = \Omega(1)$.
- *Constant secrecy:* $\tau = \Omega(1)$.

In the following, it is assumed that $\alpha \geq \tau$, an assumption that can always be made by ignoring some of the secrecy. Consider the following (k, n)-code with secret state $\mathrm{CS} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$:

- Gen: Choose a subset T of $[n]$ of size τn uniformly at random and output it.
- $\mathrm{Enc}(m)$ for $m \in \{0,1\}^k$: Compute $c = \mathsf{E}(m)$ and output it.
- $\mathrm{Dec}(c, T)$ for $c \in \{0,1\}^n$: Find a codeword $w = (w[1], \ldots, w[n])$ with $d_{\mathsf{H}}(w, c) \leq \alpha n$. If no such w exists, output $\perp$. Moreover, if $w[j] \neq c[j]$ for some $j \in T$, output $\perp$ as well. Otherwise, decode w to its corresponding plaintext m and output it.

The above code is resilient against (continuous) parallel tampering:

Theorem 6.14. *For all $p \in \mathbb{N}$, (k, n)-code* $\mathrm{CS} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *based on a (k, n, δ, τ)-LECSS satisfying the three conditions above is $(\mathcal{F}_{\mathsf{copy}}, p, \varepsilon_{\mathsf{nmc}})$-non-malleable with*

$$\varepsilon_{\mathsf{nmc}} \;=\; p(\mathcal{O}(1) \cdot e^{-\tau n/16} + e^{-\tau^2 n/4}) + p e^{-\tau^2 n}.$$

[7] The reasons for these restrictions become apparent in the proof; of course, α must be chosen small enough in order for these constraints to be satisfiable.

Instantiating the construction. Section 6.3.2 details how a LECSS satisfying the above properties can be constructed by combining high-distance binary codes with a recent result by Cramer *et al.* [CDD+15] in order to "add" secrecy. The resulting LECSS has secrecy $\tau = \Omega(1)$ and rate $\rho = \Omega(1)$ (cf. Corollary 6.23 in Section 6.3.2). The secrecy property depends on the random choice of a universal hash function. Thus, the instantiated code can be seen as a construction in the CRS model. When combined with the single-bit PKE as described above, the description of the hash function can be made part of the public key.

Security Proof

The reader is referred to Sections 5.1.3 and 5.1.4 for security definitions of non-malleable codes and notation surrounding bit-wise tampering, respectively.

Let $\mathbf{R}^{\mathsf{copy}} := \mathbf{R}_p^{\mathcal{F}_{\mathsf{copy}},\mathrm{CS}}$ and $\mathbf{S}^{\mathsf{trivial}} := \mathbf{S}_p^{\mathcal{F}_{\mathsf{triv}},\sigma}$ be as in Definition 5.5 (for a simulator σ to be determined) and fix some distinguisher $\mathbf{D}$.

In the following, queries $f \in \mathcal{F}_{\mathsf{bit}}$ with $0 \leq a(f) \leq \tau n$, $\tau n < a(f) < (1-\tau)n$, and $(1-\tau)n \leq a(f) \leq n$ are called *low queries*, *middle queries*, and *high queries*, respectively.

On a high level, the proof proceeds as follows: First, it shows that *middle* queries are rejected with high probability. For *low* and *high* queries, one can show that their effect on the decoding process can always be determined from the query itself and the bits of the encoding at the positions indexed by the secret trigger set T. Since the size of T is τn, these symbols are uniformly random and independent of the encoded message, which immediately implies a simulation strategy for σ.

Analyzing query types. The following lemma states that an isolated middle query is rejected with high probability.

Lemma 6.15. *Let $f \in \mathcal{F}_{\mathsf{copy}}$ be a middle query. Then, for any $m \in \{0,1\}^k$,*

$$\mathsf{P}[\mathrm{Dec}(f(\mathrm{Enc}(m)), T) \neq \perp] \quad \leq \quad \mathcal{O}(1) \cdot e^{-\tau n/16} + e^{-\tau^2 n/4}$$

where the probability is over the randomness of Enc *and the choice of the secret trigger set T.*

Proof. Fix $m \in \{0,1\}^k$ and a middle query $f = (f[1], \ldots, f[n])$. Suppose first that $a(f) \geq n/2$. Define

$$\mathcal{W} := \{w \in \mathbb{F}^n \mid w \text{ is codeword} \wedge \exists r : d_{\mathsf{H}}(f(\mathsf{E}(x;r)), w) \leq \alpha n\},$$

where r is the randomness of E. That is, $\mathcal{W}$ is the set of all codewords that could possibly be considered while decoding an encoding of x tampered with via f. Consider two distinct codewords $w, w' \in \mathcal{W}$. From the definition of $\mathcal{W}$ it is apparent that $w[j] \neq \mathsf{val}(f[j])$ for at most αn positions $j \in A(f)$ (and similarly for w'), which implies that w and w' differ in at most $2\alpha n$ positions $j \in A(f)$. Therefore, w and w' differ in at least $(\delta - 2\alpha)n$ positions $j \notin A(f)$.

For $w \in \mathcal{W}$, let $\tilde{w}$ be the projection of w onto the unfixed positions $j \notin A(f)$ and set $\tilde{\mathcal{W}} := \{\tilde{w} \mid w \in \mathcal{W}\}$. The above distance argument implies that $|\mathcal{W}| = |\tilde{\mathcal{W}}|$. Moreover, $\tilde{\mathcal{W}}$ is a binary code with block length $n - a(f)$ and relative distance at least

$$\frac{(\delta - 2\alpha)n}{n - a(f)} \geq \frac{(\delta - 2\alpha)n}{n/2} = 2\delta - 4\alpha > 1/2,$$

where the last inequality follows from the fact that δ and α are such that $\delta - 2\alpha > 1/4$. Therefore, by the Plotkin bound (Theorem 2.3 in Section 2.12),[8]

$$|\mathcal{W}| = |\tilde{\mathcal{W}}| \leq \mathcal{O}(1).$$

Denote by $c = (c[1], \ldots, c[n])$ and $\tilde{c} = (\tilde{c}[1], \ldots, \tilde{c}[n])$ the (random variables corresponding to the) encoding $c = \mathrm{Enc}(m)$ and the tampered encoding $\tilde{c} = f(c)$, respectively. For an arbitrary (n-bit) codeword $w \in \mathcal{W}$,

$$\mathbf{E}[d_{\mathsf{H}}(\tilde{c}, w)] \quad = \quad \sum_{j=1}^{n} \mathbf{E}[d_{\mathsf{H}}(\tilde{c}[j], w[j])] \quad \geq \quad \sum_{j \in J} \mathbf{E}[d_{\mathsf{H}}(\tilde{c}[j], w[j])],$$

where $J \subseteq [n]$ is the set containing the indices of the first τn bits *not* fixed by f. Note that by the definition of middle queries, there are at least that many, i.e., $|J| = \tau n$.

Observe that for $j \in J$, $d_{\mathsf{H}}(\tilde{c}[j], w[j])$ is an indicator variable with expectation $\mathbf{E}[d_{\mathsf{H}}(\tilde{c}[j], w[j])] \geq \frac{1}{2}$, since $c[j]$ is a uniform bit. Thus,

$$\mathbf{E}[d_{\mathsf{H}}(\tilde{c}, w)] \geq \frac{\tau n}{2}.$$

Additionally, $(d_{\mathsf{H}}(\tilde{c}[j], w[j]))_{j \in J}$ are independent. Therefore, using a Chernoff bound (Theorem 2.2 in Section 2.11), for $\varepsilon > 0$

$$\mathsf{P}[d_{\mathsf{H}}(\tilde{c}, w) \quad < \quad (1 - \varepsilon)\tau n/2] \quad \leq \quad e^{-\tau \varepsilon^2 n/4}.$$

[8] The size constant absorbed by $\mathcal{O}(1)$ here depends on how close $2\delta - 4\alpha$ is to $1/2$.

Therefore, the probability that there exists $w \in \mathcal{W}$ for which the above does not hold is at most

$$|\mathcal{W}| \cdot e^{-\tau\varepsilon^2 n/4} \quad \leq \quad \mathcal{O}(1) \cdot e^{-\tau\varepsilon^2 n/4},$$

by a union bound.

Suppose now that $d_{\mathsf{H}}(\tilde{c}, w) \geq (1-\varepsilon)\tau n/2$ for all codewords $w \in \mathcal{W}$. Then, over the choice of T,[9]

$$\mathsf{P}[\forall j \in T : d_{\mathsf{H}}(\tilde{c}[j], w[j]) = 0] \quad \leq \quad (1-(1-\varepsilon)\tau/2)^{\tau n} \quad \leq \quad e^{-(1-\varepsilon)\tau^2 n/2}.$$

The lemma now follows by setting $\varepsilon := \frac{1}{2}$.

If $a(f) < n/2$ an analogous argument can be made for the difference $d := \tilde{c} - c$ between the encoding and the tampered codeword, as such a query f fixes at least half of the bits of d (to 0, in fact) and $\mathsf{D}(d) \neq \perp$ implies $\mathsf{D}(\tilde{c}) \neq \perp$. □

It turns out that low and high queries always result in $\perp$ or one other value.

Lemma 6.16. *Low queries $f \in \mathcal{F}_{\mathsf{copy}}$ can result only in $\perp$ or the originally encoded message $m \in \{0,1\}^k$. High queries $f \in \mathcal{F}_{\mathsf{copy}}$ can result only in $\perp$ or one other value $m_f \in \{0,1\}^k$, which solely depends on f. Furthermore, m_f, if existent, can be found efficiently given f.*

Proof. The statement for low queries is trivial, since a low query f cannot change the encoding beyond the error correction bound αn.

Consider now a high query f and the following efficient procedure:

1. Compute $\tilde{c}_f \leftarrow f(0^n)$.
2. Find a codeword w_f with $d_{\mathsf{H}}(w_f, \tilde{c}_f) \leq 2\alpha n$ (which is possible since $2\alpha < \delta/2$).
3. Output w_f or $\perp$ if none exists.

Consider an arbitrary encoding c and let $\tilde{c} \leftarrow f(c)$ be the tampered encoding. Assume there exists w with $d_{\mathsf{H}}(w, \tilde{c}) \leq \alpha n$. Since a high query f fixes all but τn bits, $d_{\mathsf{H}}(\tilde{c}, \tilde{c}_f) \leq \tau n \leq \alpha n$, and, thus, $d_{\mathsf{H}}(w, \tilde{c}_f) \leq 2\alpha n$, by the triangle inequality. Hence, $w = w_f$.

In other words, if the decoding algorithm Dec on $\tilde{c}$ finds a codeword $w = w_f$, one can find it using the above procedure, which also implies that high queries can only result in $\perp$ or one other message $m_f = \mathsf{D}(w_f)$. □

[9] Recall that $|T| = \tau n$.

Handling middle queries. Consider the hybrid game $\mathbf{H}_1$ that behaves as $\mathbf{R}^{\mathsf{copy}}$, except that it answers all middle queries by $\perp$.

Lemma 6.17. $\Delta^{\mathbf{D}}(\mathbf{R}^{\mathsf{copy}}, \mathbf{H}_1) \quad \leq \quad p(\mathcal{O}(1) \cdot e^{-\tau n/16} + e^{-\tau^2 n/4})$.

The proof of Lemma 6.17 follows a generic paradigm, at whose core is the so-called *self-destruct lemma*, which deals with the indistinguishability of hybrids with the self-destruct property and is explained in detail in Section 7. Roughly, this lemma applies whenever the first hybrid (in this case $\mathbf{R}^{\mathsf{copy}}$) can be turned into the second one (in this case $\mathbf{H}_1$) by changing ("bending") the answers to a subset (the "bending set") of the possible queries to always be $\perp$, and when additionally non-bent queries have a unique answer (cf. the statement of Lemma 7.1). Intuitively, the lemma states that parallelism and adaptivity do not help distinguish (much) in such cases, which allows using Lemma 6.15.

Proof. The lemma is proved conditioned on the message m encoded by $\mathbf{D}$. To use the self-destruct lemma, note first that both $\mathbf{R}^{\mathsf{copy}}$ and $\mathbf{H}_1$ answer parallel tamper queries in which each component is from the set $\mathcal{X} := \mathcal{F}$ by vectors whose components are in $\mathcal{Y} := \{0,1\}^k \cup \{\perp\}$. Moreover, both hybrids use as internal randomness a uniformly chosen element from $\mathcal{R} := \{0,1\}^\rho \times \mathcal{S}$, where ρ is an upper bound on the number of random bits used by Enc and $\mathcal{S}$ is the set of all τn-subsets T of $[n]$. $\mathbf{R}^{\mathsf{copy}}$ answers each component of a query $f \in \mathcal{X}$ by

$$g(f,(r,T)) \quad := \quad \mathrm{Dec}(f(\mathrm{Enc}(m;r)),T).$$

Define $\mathcal{B} \subseteq \mathcal{X}$ to be the set of all middle queries; $\mathbf{H}_1$ is the $\mathcal{B}$-bending of $\mathbf{R}^{\mathsf{copy}}$ (cf. Definition 7.2).

Observe that queries $f \notin \mathcal{B}$ are either low or high queries. For low queries f, the unique answer is $y_f = m$, and for high queries f, $y_f = m_f$ (cf. Lemma 6.16). Thus, by Lemmas 7.1 and 6.15,

$$\begin{aligned} \Delta^{\mathbf{D}}(\mathbf{R}^{\mathsf{copy}}, \mathbf{H}_1) \quad &\leq \quad p \cdot \max_{f \in \mathcal{B}} \mathsf{P}[g(f,(r,T)) \neq \perp] \\ &\leq \quad p(\mathcal{O}(1) \cdot e^{-\tau n/16} + e^{-\tau^2 n/4}), \end{aligned}$$

where the probability is over the choice of (r,T). □

Handling high queries. Consider the following hybrid game $\mathbf{H}_2$: It differs from $\mathbf{H}_1$ in the way it decodes high queries f. Instead of applying the normal decoding algorithm to the tampered codeword $\tilde{c}$, it proceeds as follows:

1. Find w_f (as in the proof of Lemma 6.16).
2. If w_f does not exist, return $\perp$.
3. If $\tilde{c}[j] = w_f[j]$ for all $j \in T$, return $\mathrm{Dec}(w)$. Otherwise, return $\perp$.

Lemma 6.18. $\Delta^{\mathbf{D}}(\mathbf{H}_1, \mathbf{H}_2) \quad \leq \quad pe^{-\tau^2 n}$.

Proof. The lemma is proved conditioned on the message m encoded by $\mathbf{D}$ *and* the randomness r of the encoding. For the remainder of the proof, r is therefore considered fixed inside $\mathbf{H}_1$ and $\mathbf{H}_2$. The proof, similarly to that of Lemma 6.17, again uses the self-destruct lemma.

Set $\mathcal{X} := \mathcal{F}$ and $\mathcal{Y} := \{0,1\}^k \cup \{\perp\}$. However, this time, let $\mathcal{R} := \mathcal{S}$. For $f \in \mathcal{X}$ and $T \in \mathcal{R}$, define

$$g(f, T) \quad := \quad \mathrm{Dec}(\tilde{c}, T),$$

where $\tilde{c} := f(\mathrm{Enc}(m; r))$. The bending set $\mathcal{B} \subseteq \mathcal{X}$ is the set of all high queries f such that w_f exists and $d_{\mathsf{H}}(w_f, \tilde{c}) > \alpha n$.[10] It is readily verified that $\mathbf{H}_2$ is a parallel stateless self-destruct game (cf. Definition 7.1) that behaves according to g, and that $\mathbf{H}_1$ is its $\mathcal{B}$-bending.

Consider a query $f \notin \mathcal{B}$. If f is a low query, the unique answer is $y_f = m$; if it is a middle query, $y_f = \perp$; if it is a high query, $y_f = m_f$ (cf. Lemma 6.16). Therefore,

$$\Delta^{\mathbf{D}}(\mathbf{H}_1, \mathbf{H}_2) \quad \leq \quad \max_{f \in \mathcal{B}} \mathsf{P}[g(f, T) \neq \perp] \quad \leq \quad pe^{-\tau^2 n},$$

where the first inequality follows from the self-destruct lemma (Lemma 7.1) and the second one from the fact that $d_{\mathsf{H}}(m_f, \tilde{c}) > \tau n$ for queries $f \in \mathcal{B}$, and therefore the probability over the choice of T that it is accepted is at most $(1-\tau)^{\tau n} \leq e^{-\tau^2 n}$. □

Simulation. By analyzing hybrid $\mathbf{H}_2$, one observes that low and high queries can now be answered knowing only the query itself and the symbols of the encoding indexed by the secret trigger set $T \in \mathcal{S}$.

Lemma 6.19. *Consider the random experiment of distinguisher* $\mathbf{D}$ *interacting with* $\mathbf{H}_2$. *There is an efficiently computable function* $\mathrm{Dec}' : \mathcal{F}_{\mathsf{copy}} \times \mathcal{S} \times \{0,1\}^{\tau n} \to \{0,1\}^k \cup \{\mathsf{same}, \perp\}$ *such that for any low or high query* f, *any fixed message* m, *any fixed encoding* c *thereof, and any output* T *of* Gen,

$$\left[\mathrm{Dec}'(f, T, (c[j])_{j \in T})\right]_{\mathsf{same}/m} \quad = \quad \mathrm{Dec}(f(c)),$$

[10] These are queries potentially accepted by $\mathbf{H}_2$ but not by $\mathbf{H}_1$.

where $[\cdot]_{\mathsf{same}/m}$ *is the identity function except that* same *is replaced by* m *and where* $(c[j])_{j\in T}$ *are the symbols of* c *specified by* T*.*

Proof. Consider a low query f. Due to the error correction, $\mathrm{Dec}(f(c))$ is the message originally encoded if no bit indexed by T is changed and $\perp$ otherwise. Which one is the case can clearly be efficiently computed from f, T, and $(c[j])_{j\in T}$.

For high queries f the statement follows by inspecting the definition of $\mathbf{H}_2$ and Lemma 6.16. □

In $\mathbf{H}_2$, by the τn-secrecy of the LECSS, the distribution of the symbols indexed by T is independent of the message m encoded by $\mathbf{D}$. Moreover, the distribution of T is trivially independent of m. This suggests the following simulator σ: Initially, it chooses a random subset T from $\binom{[n]}{\tau n}$ and chooses τn random symbols $(c[j])_{j\in T}$. Every component f of any tamper query is handled as follows: If f is a low or a high query, the answer is $\mathrm{Dec}'(f, T, (c[j])_{j\in T})$; if f is a middle query, the answer is $\perp$. This implies:

Lemma 6.20. $\mathbf{H}_2 \equiv \mathbf{S}^{\mathsf{trivial}}$.

Proof of Theorem 6.14. Follows from Lemmas 6.17, 6.18, and 6.20 and a triangle inequality. □

6.3.2 LECSS for the Non-Malleable Code

Let $\mathbb{F} = \mathrm{GF}(2)$ and $\alpha > 0$. This section shows shows how to construct a (k, n, δ, τ)-LECSS (E, D) (cf. Definition 2.5 in Section 2.9) with minimum distance δ and secrecy τ over $\mathbb{F}$ and the following properties (as required in Section 6.3.1):

- *Minimum distance:* $\delta > 1/4 + 2\alpha$ and $\delta/2 > 2\alpha$.
- *Constant rate:* $k/n = \Omega(1)$.
- *Constant secrecy:* $\tau = \Omega(1)$.

The construction combines high-distance binary codes with a recent result by Cramer *et al.* [CDD+15], which essentially allows to "add" secrecy to any code of sufficient rate.

Let $\mathcal{C}$ be a (n, l)-code with rate $R = \frac{l}{n}$ over $\mathbb{F}$. In the following we write $\mathcal{C}(x)$ for the codeword corresponding to $x \in \mathbb{F}^l$ and $\mathcal{C}^{-1}(c, e)$ for the output of the efficient error-correction algorithm attempting to correct up

to e errors on c, provided that $e < \delta n/2$;[11] the output is $\perp$ if there is no codeword within distance e of c.

Adding secrecy. Let l be such that $k < l < n$. The construction by [CDD+15] combines a surjective linear universal hash function $\mathsf{h} : \mathbb{F}^l \to \mathbb{F}^k$ with $\mathcal{C}$ to obtain a LECSS (E, D) as follows:[12]

- $\mathsf{E}(m)$ for $m \in \{0,1\}^k$: Choose $s \in \{0,1\}^l$ randomly such that $\mathsf{h}(s) = x$ and output $c = \mathcal{C}(s)$.
- $\mathsf{D}(c, e)$ for $c \in \{0,1\}^n$ and $e < \delta n/2$: Compute $s = \mathcal{C}^{-1}(c, e)$. If $s = \perp$, output $\perp$. Otherwise, output $x = \mathsf{h}(s)$.

The resulting LECSS has rate $\rho = \frac{k}{ln}$ and retains all distance and error-correction properties of $\mathcal{C}$. Additionally, if R is not too low, the LECSS has secrecy. More precisely, Cramer *et al.* prove the following theorem:

Theorem 6.21 ([CDD+15])**.** *Let $\tau > 0$ and $\eta > 0$ be constants and $\mathcal{H}$ be a family of linear universal hash functions $\mathsf{h} : \mathbb{F}^l \to \mathbb{F}^k$. Given that $R \geq \rho + \eta + \tau + h(\tau)$, there exists a function $\mathsf{h} \in \mathcal{H}$ such that (E, D) achieves secrecy τ. Moreover, such a function h can be chosen randomly with success probability $1 - 2^{-\eta n}$.*

It should be pointed out that the version of the above theorem in [CDD+15] does not claim that any τn bits of an encoding are uniform and independent but merely that they are independent of the message encoded. However, by inspecting their proof, it can be seen that uniformity is guaranteed if $\tau n \leq l - k$, which is the case if and only if $\tau \leq \frac{l}{n} - \frac{k}{n} = R - \rho$, which is clearly implied by the precondition of the theorem.

Zyablov bound. For code $\mathcal{C}$, we use concatenated codes reaching the Zyablov bound:

Theorem 6.22. *For every $\delta < 1/2$ and all sufficiently large n, there exists a code $\mathcal{C}$ that is*

- *linear,*

[11] This assumes that $\mathcal{C}$ is efficiently decodable up to relative distance $\delta/2$. However, while the codes we consider here have this property, for our non-malleable code construction, it would be sufficient to have efficient error correction up to distance 2α for whatever particular choice of the constant α.

[12] Note that we switched the roles of l and k here in order to remain consistent with the notation in this paper.

- *efficiently encodable,*
- *of distance at least δn,*
- *allows to efficiently correct up to $\delta n/2$ errors,*

and has rate

$$R \geq \max_{0 \leq r \leq 1-h(\delta+\varepsilon)} r\left(1 - \frac{\delta}{h^{-1}(1-r)-\varepsilon}\right),$$

for $\varepsilon > 0$ and where $h(\cdot)$ is the binary entropy function.

The Zyablov bound is achieved by concatenating Reed-Solomon codes with linear codes reaching the Gilbert-Varshamaov bound (which can be found by brute-force search in this case). Alternatively, Shen [She93] showed that the bound is also reached by an explicit construction using algebraic geometric codes.

Choice of parameters. Set $\alpha := 1/200$ and $\delta := 1/4 + 2\alpha + \varepsilon$ for $\varepsilon := 1/500$, say. Then, $\delta - 2\alpha > 1/4$, as required. Moreover, the rate of the Zyablov code with said distance δ can be approximated to be $R \geq 0.0175$. Setting, $\tau := 1/1000$ yields $\tau + h(\tau) \leq 0.0125$, leaving a possible rate for the LECSS of up to $\rho \approx 0.005 - \eta$. Hence:

Corollary 6.23. *For any $\alpha > 0$ there exists a (k, n, δ, τ)-LECSS (E, D) with the following properties:*

- Minimum distance: $\delta > 1/4 + 2\alpha$ *and* $\delta/2 > 2\alpha$.
- Constant rate: $k/n = \Omega(1)$.
- Constant secrecy: $\tau = \Omega(1)$.

6.3.3 On the Necessity of Secret State

No (k, n)-coding scheme (Enc, Dec) *without secret state* can be non-malleable against even a single *parallel* tampering query from $\mathcal{F}_{\mathsf{copy}}$.

For the remainder of this section, let $\mathbf{R}^{\mathsf{copy}} := \mathbf{R}_p^{\mathcal{F}_{\mathsf{copy}},\mathrm{CS}}$ and $\mathbf{S}^{\mathsf{trivial}} := \mathbf{S}_p^{\mathcal{F}_{\mathsf{triv}},\sigma}$ for an *arbitrary* simulator σ and $p \geq n$.

Note that the attacker in Theorem 6.24 below is not efficient, and therefore it remains an open question whether security against parallel tampering can be obtained via cryptographically secure codes.

Theorem 6.24. *There exists a distinguisher* $\mathbf{D}$ *such that for all coding schemes* $\mathrm{CS} = (\mathrm{Enc}, \mathrm{Dec})$ *and all simulators* σ*,*

$$\Delta^{\mathbf{D}}(\mathbf{R}^{\mathsf{copy}}, \mathbf{S}^{\mathsf{trivial}}) \;\geq\; 1 - \frac{1}{2^k - 2n}.$$

Proof. Distinguisher $\mathbf{D} := \mathbf{D}_{\mathsf{Ext}}$ uses an algorithm Ext that always extracts the encoded message when interacting with system $\mathbf{R}^{\mathsf{copy}}$ and does so with small probability only when interacting with system $\mathbf{S}^{\mathsf{trivial}}$.

Distinguishing strings. For a position $i \in [n]$, let $c_i, c_i' \in \{0,1\}^n$ be two strings that differ exactly in the i^{th} bit and decode to two different values. Clearly, one can without loss of generality assume that for all positions in a coding scheme such strings exist.

The extraction algorithm. Consider the following algorithm, denoted Ext: It issues a *single* parallel tamper query $(\mathsf{tamper}, f^{(1)}, \ldots, f^{(n)})$, where the i^{th} function $f^{(i)} \in \mathcal{F}_{\mathsf{copy}}$ is determined as follows: For $j \neq i$,

$$f^{(i)}[j] := \begin{cases} \mathsf{zero} & \text{if } c_i[j] = 0, \\ \mathsf{one} & \text{otherwise,} \end{cases}$$

and $f^{(i)}[i] = \mathsf{keep}$.

Upon receiving an answer $(m^{(1)}, \ldots, m^{(n)})$ with $m^{(i)} \in \{0,1\}^k \cup \{\bot\}$, it sets

$$x[i] := \begin{cases} c_i[i] & \text{if } m^{(i)} = \mathrm{Dec}(c_i), \\ c_i'[i] & \text{otherwise.} \end{cases}$$

The algorithm finally outputs $\mathrm{Dec}(x[1], \ldots, x[n])$.

The distinguisher. Consider the following distinguisher $\mathbf{D}_{\mathsf{Ext}}$: Initially, it chooses

$$m \leftarrow \{0,1\}^k \setminus \{\mathrm{Dec}(c_i), \mathrm{Dec}(c_i') \mid i \in [n]\}$$

and outputs m to the A-interface of the system it is connected to. Then, it lets Ext interact with that system, forwarding the tamper queries to the E-interface and the answers from the B-interface. If at some point one of the answers is the initially encoded message, the algorithm outputs 0 and terminates. Otherwise, when Ext terminates and outputs a value m', $\mathbf{D}_{\mathsf{Ext}}$ outputs 1 if $m' = m$ and 0 otherwise.

Real world. It can be easily verified that in the real world, the output of Ext is always the message initially encoded by $\mathbf{D}_{\mathsf{Ext}}$. Moreover, all of the messages $m^{(i)}$ are clearly in $\{\mathrm{Dec}(c_i), \mathrm{Dec}(c'_i) \mid i \in [n]\}$ and $\mathbf{D}_{\mathsf{Ext}}$ never outputs 0. Therefore,

$$\mathsf{P}[\mathbf{D}_{\mathsf{Ext}}\mathbf{R}^{\mathsf{copy}} = 1] = 1.$$

Ideal world. By definition, if one of the answers $m^{(i)}$ equals the originally encoded message, $\mathbf{D}_{\mathsf{Ext}}$ outputs 0. Thus, whenever the output of Ext influences the output of $\mathbf{D}_{\mathsf{Ext}}$, Ext's view is independent of the original message m. Therefore, the probability that Ext outputs m is at most $\frac{1}{2^k - 2n}$. Hence,

$$\mathsf{P}[\mathbf{D}_{\mathsf{Ext}}\mathbf{S}^{\mathsf{trivial}} = 1] \leq \frac{1}{2^k - 2n}.$$

This proves the theorem. □

Chapter 7

A General Indistinguishability Paradigm

A recurring issue in this paper are proofs that certain self-destruct games answering successive parallel decryption/tampering queries are indistinguishable. Such games are formalized as *parallel stateless self-destruct games*.

Definition 7.1. *A system* $\mathbf{U}$ *is a* parallel stateless self-destruct (PSSD) game *if*

- *it accepts parallel queries in which each component is from some set* $\mathcal{X}$ *and answers them by vectors with components from some set* $\mathcal{Y}$*,*
- $\perp \in \mathcal{Y}$*,*
- *there exists a function* $g : \mathcal{X} \times \mathcal{R} \to \mathcal{Y}$ *such that every query component* $x \in \mathcal{X}$ *is answered by* $g(x, r)$*, where* $r \in \mathcal{R}$ *is the internal randomness of* $\mathbf{U}$*, and*
- *the game self-destructs, i.e., after the first occurrence of* $\perp$ *in an answer vector all* further *outputs are* $\perp$*.*

A PSSD game can be transformed into a related one by "bending" the answers to some of the queries $x \in \mathcal{X}$ to the value $\perp$. This is captured by the following definition:

Definition 7.2. *Let* $\mathbf{U}$ *be a PSSD game that behaves according to* g *and let* $\mathcal{B} \subseteq \mathcal{X}$*. The* $\mathcal{B}$-bending of $\mathbf{U}$*, denoted by* $\mathbf{U}'$*, is the PSSD game that behaves according to* g'*, where*

$$g'(x,r) \quad = \quad \begin{cases} \perp & \text{if } x \in \mathcal{B}, \\ g(x,r) & \text{otherwise.} \end{cases}$$

The *self-destruct lemma* below states that in order to bound the distinguishing advantage between a PSSD and its bending, one merely needs to analyze a single, non-parallel query, provided that all non-bent queries x can only be answered by a unique value y_x or $\perp$.

Lemma 7.1. *Let* $\mathbf{U}$ *be a PSSD game and* $\mathbf{U}'$ *its* $\mathcal{B}$*-bending for some* $\mathcal{B} \subseteq \mathcal{X}$*. If for all* $x \notin \mathcal{B}$ *there exists* $y_x \in \mathcal{Y}$ *such that*

$$\{g(x,r) \mid r \in \mathcal{R}\} \quad = \quad \{y_x, \perp\},$$

then, for all distinguishers $\mathbf{D}$*,*

$$\Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{U}') \quad \leq \quad p \cdot \max_{x \in \mathcal{B}} \mathsf{P}[g(x,R) \neq \perp],$$

where the probability is over the choice of R *and where* p *is the number of components the largest parallel query.*

Proof. Fix a distinguisher $\mathbf{D}$ and denote by R and R' the random variables corresponding to the internal randomness of $\mathbf{U}$ and $\mathbf{U}'$, respectively. Call a value $x \in \mathcal{X}$ *dangerous* if $x \in \mathcal{B}$ and a query dangerous if it contains a dangerous value.

In the random experiment corresponding to the interaction between $\mathbf{D}$ and $\mathbf{U}$, define the event E that the first dangerous query contains a dangerous value X with $g(X,R) \neq \perp$ and that the self-destruct has not been provoked yet. Similarly, define the event E' for the interaction between $\mathbf{D}$ and $\mathbf{U}'$ that the first dangerous query contains a dangerous value X' with $g(X',R') \neq \perp$ and that the self-destruct has not been provoked yet.[1]

Clearly, $\mathbf{U}$ and $\mathbf{U}'$ behave identically unless E resp. E' occur. Thus, it remains to bound $\mathsf{P}[E] = \mathsf{P}[E']$. To that end, note that adaptivity does not help in provoking E. For any distinguisher $\mathbf{D}$, there exists a *non-adaptive* distinguisher $\tilde{\mathbf{D}}$ such that whenever $\mathbf{D}$ provokes E, so does $\mathbf{D}'$. $\mathbf{D}'$ proceeds as follows: First, it interacts with $\mathbf{D}$ only. Whenever $\mathbf{D}$

[1] Note that the function g is the *same* in the definitions of either event.

asks a non-dangerous query, $\mathbf{D}'$ answers every component $x \notin \mathcal{B}$ by y_x. As soon as $\mathbf{D}$ specifies a dangerous query, $\mathbf{D}'$ stops its interaction with $\mathbf{D}$ and sends all queries to $\mathbf{U}$.

Fix all randomness in experiment $\mathbf{D}'\mathbf{U}$, i.e., the coins of $\mathbf{D}$ (inside $\mathbf{D}'$) and the randomness r of $\mathbf{U}$. Suppose $\mathbf{D}$ would provoke E in the direct interaction with $\mathbf{U}$. In such a case, all the answers by $\mathbf{D}'$ are equal to the answers by $\mathbf{U}$, since, by assumption, the answers to components $x \notin \mathcal{B}$ in non-dangerous queries are y_x or $\perp$ and the latter is excluded if E is provoked. Thus, whenever $\mathbf{D}$ provokes E, $\mathbf{D}'$ provokes it as well.

The success probability of non-adaptive distinguishers $\mathbf{D}$ is bounded from above by the probability over R that their first dangerous query provokes E, which is at most $p \cdot \max_{x \in \mathcal{B}} \mathsf{P}[g(x, R) \neq \perp]$. □

Bibliography

[ADKO15a] Divesh Aggarwal, Yevgeniy Dodis, Tomasz Kazana, and Maciej Obremski. Non-malleable reductions and applications. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 459–468, 2015.
(Cited on page 12.)

[ADKO15b] Divesh Aggarwal, Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Leakage-resilient non-malleable codes. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 398–426, 2015.
(Cited on pages 2, 12, and 62.)

[ADL14] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 774–783, 2014.
(Cited on page 12.)

[AGM+15a] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Explicit non-malleable codes against bit-wise tampering and permutations. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 538–557, 2015.
(Cited on page 12.)

[AGM+15b] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. A rate-optimizing compiler

for non-malleable codes against bit-wise tampering and permutations. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 375–397, 2015.
(Cited on page 12.)

[BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, pages 259–274, 2000.
(Cited on page 57.)

[BDJR97] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A Concrete Security Treatment of Symmetric Encryption. In *38th FOCS*, pages 394–403, 1997.
(Cited on page 21.)

[Bea91] Donald Beaver. Foundations of secure interactive computing. In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *LNCS*, pages 377–391, Heidelberg, 1991. Springer.
(Cited on page 11.)

[BHK09] Mihir Bellare, Dennis Hofheinz, and Eike Kiltz. Subtleties in the Definition of IND-CCA: When and How Should Challenge-Decryption be Disallowed? Cryptology ePrint Archive 2009/418, 2009.
(Cited on pages 7, 33, and 37.)

[BPW07] Michael Backes, Birgit Pfitzmann, and Michael Waidner. The Reactive Simulatability (RSIM) Framework for Asynchronous Systems. *Information and Computation*, 205(12):1685–1720, December 2007.
(Cited on page 11.)

[BS99] Mihir Bellare and Amit Sahai. Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 519–536, 1999.
(Cited on page 8.)

[Can00] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. *IACR Cryptology ePrint Archive*, 2000:67, 2000.
(Cited on pages 11, 12, and 15.)

[CDD+15] Ronald Cramer, Ivan Bjerre Damgård, Nico Döttling, Serge Fehr, and Gabriele Spini. Linear secret sharing schemes from error correcting codes and universal hash functions. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, pages 313–336, 2015.
(Cited on pages 93, 98, and 99.)

[CDF+08] 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 *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, pages 471–488, 2008.
(Cited on page 23.)

[CDMW08] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Black-box construction of a non-malleable encryption scheme from any semantically secure one. In *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008.*, pages 427–444, 2008.
(Cited on pages vii, xi, 2, 8, 9, 11, 50, 51, 52, 53, 54, 57, and 60.)

[CDTV16] Sandro Coretti, Yevgeniy Dodis, Björn Tackmann, and Daniele Venturi. Non-malleable encryption: Simpler, shorter, stronger. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 306–335, 2016.
(Cited on page 4.)

[CG14a] Mahdi Cheraghchi and Venkatesan Guruswami. Capacity of non-malleable codes. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-*

14, 2014, pages 155–168, 2014.
(Cited on page 12.)

[CG14b] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, pages 440–464, 2014.
(Cited on pages 12 and 58.)

[CHH+07] Ronald Cramer, Goichiro Hanaoka, Dennis Hofheinz, Hideki Imai, Eike Kiltz, Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Bounded cca2-secure encryption. In *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, pages 502–518, 2007.
(Cited on pages 2, 9, 58, and 75.)

[CK02] Ran Canetti and Hugo Krawczyk. Universally Composable Notions of Key Exchange and Secure Channels. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 337–351, Heidelberg, 2002. Springer.
(Cited on pages 11 and 40.)

[CKM11] Seung Geol Choi, Aggelos Kiayias, and Tal Malkin. Bitr: Built-in tamper resilience. In *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, pages 740–758, 2011.
(Cited on page 12.)

[CKN03] Ran Canetti, Hugo Krawczyk, and Jesper Buus Nielsen. Relaxing Chosen-Ciphertext Security. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 565–582, Heidelberg, 2003. Springer.
(Cited on pages 7, 11, 22, 32, 33, 34, 35, 39, and 70.)

[CMT13] Sandro Coretti, Ueli Maurer, and Björn Tackmann. Constructing confidential channels from authenticated channels - public-key encryption revisited. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the*

Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I, pages 134–153, 2013.
(Cited on page 4.)

[CMTV15] Sandro Coretti, Ueli Maurer, Björn Tackmann, and Daniele Venturi. From single-bit to multi-bit public-key encryption via non-malleable codes. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 532–560, 2015.
(Cited on page 4.)

[CS98] Ronald Cramer and Victor Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In Hugo Krawczyk, editor, *CRYPTO 1998*, volume 1462 of *LNCS*, pages 13–25, Heidelberg, 1998. Springer.
(Cited on page 2.)

[CS01] Ronald Cramer and Victor Shoup. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. *SIAM Journal on Computing*, 33:167–226, 2001.
(Cited on page 2.)

[CZ14] Eshan Chattopadhyay and David Zuckerman. Non-malleable codes against constant split-state tampering. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 306–315, 2014.
(Cited on page 12.)

[Dac14] Dana Dachman-Soled. A black-box construction of a CCA2 encryption scheme from a plaintext aware (spa1) encryption scheme. In *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*, pages 37–55, 2014.
(Cited on page 2.)

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.
(Cited on pages 7 and 8.)

[DF14] Yevgeniy Dodis and Dario Fiore. Interactive encryption and message authentication. In *Security and Cryptography for Networks - 9th International Conference, SCN 2014, Amalfi, Italy, September 3-5, 2014. Proceedings*, pages 494–513, 2014.
(Cited on page 2.)

[DFMV13] Ivan Damgård, Sebastian Faust, Pratyay Mukherjee, and Daniele Venturi. Bounded tamper resilience: How to go beyond the algebraic barrier. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II*, pages 140–160, 2013.
(Cited on page 12.)

[DFMV15] Ivan Damgård, Sebastian Faust, Pratyay Mukherjee, and Daniele Venturi. The chaining lemma and its application. In *Information Theoretic Security - 8th International Conference, ICITS 2015, Lugano, Switzerland, May 2-5, 2015. Proceedings*, pages 181–196, 2015.
(Cited on page 12.)

[DH76] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.
(Cited on page 1.)

[DKO13] Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable codes from two-source extractors. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, pages 239–257, 2013.
(Cited on page 12.)

[DLSZ15] Dana Dachman-Soled, Feng-Hao Liu, Elaine Shi, and Hong-Sheng Zhou. Locally decodable and updatable non-malleable codes and their applications. In *Theory of Cryptography -*

12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I, pages 427–450, 2015.
(Cited on page 12.)

[DPW10] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 434–452, 2010.
(Cited on pages vi, x, xi, 2, 10, 11, 12, 22, 58, 60, 61, 62, 77, 80, and 87.)

[ElG84] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, pages 10–18, 1984.
(Cited on page 2.)

[FMNV14] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. Continuous non-malleable codes. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, pages 465–488, 2014.
(Cited on pages vi, x, 3, 10, 61, and 85.)

[FMNV15] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. A tamper and leakage resilient von neumann architecture. In *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, pages 579–603, 2015.
(Cited on page 63.)

[FMVW14] Sebastian Faust, Pratyay Mukherjee, Daniele Venturi, and Daniel Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 111–128, 2014.
(Cited on page 12.)

[GL90] Shafi Goldwasser and Leonid A. Levin. Fair computation of general functions in presence of immoral majority. In *CRYPTO*, pages 77–93, 1990.
(Cited on page 11.)

[GLM+04] Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, pages 258–277, 2004.
(Cited on pages 12 and 85.)

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.
(Cited on pages 1 and 9.)

[GMM07] Yael Gertner, Tal Malkin, and Steven Myers. Towards a Separation of Semantic and CCA Security for Public Key Encryption. In *TCC*, pages 434–455, 2007.
(Cited on page 2.)

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract). In *17th ACM STOC*, pages 291–304, 1985.
(Cited on page 11.)

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. In *19th ACM STOC*, pages 218–229, 1987.
(Cited on page 11.)

[HK09] Dennis Hofheinz and Eike Kiltz. Practical Chosen Ciphertext Secure Encryption from Factoring. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 313–332, Heidelberg, 2009. Springer.
(Cited on page 2.)

[HLW12] Susan Hohenberger, Allison B. Lewko, and Brent Waters. Detecting dangerous queries: A new approach for chosen ciphertext security. In *EUROCRYPT*, pages 663–681, 2012.
(Cited on pages vi, x, 2, 12, 60, 74, and 75.)

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 44–61, 1989.
(Cited on page 2.)

[JW15] Zahra Jafargholi and Daniel Wichs. Tamper detection and continuous non-malleable codes. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 451–480, 2015.
(Cited on pages 3 and 12.)

[KMO+13] Markulf Kohlweiss, Ueli Maurer, Cristina Onete, Björn Tackmann, and Daniele Venturi. Anonymity-preserving public-key encryption: A constructive approach. In Emiliano De Cristofaro and Matthew Wright, editors, *Privacy Enhancing Technologies — 13th International Symposium*, volume 7981 of *Lecture Notes in Computer Science*, pages 19–39. Springer, July 2013.
(Cited on pages 12 and 18.)

[LL12] Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 517–532, 2012.
(Cited on page 12.)

[LT13] Huijia Lin and Stefano Tessaro. Amplification of chosen-ciphertext security. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 503–519, 2013.
(Cited on page 2.)

[Mau02] Ueli Maurer. Indistinguishability of random systems. In Lars Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 110–132. Springer-Verlag, May 2002.
(Cited on page 15.)

[Mau11] Ueli Maurer. Constructive cryptography – a new paradigm for security definitions and proofs. In S. Moedersheim and C. Palamidessi, editors, *Theory of Security and Applications (TOSCA 2011)*, volume 6993 of *Lecture Notes in Computer Science*, pages 33–56. Springer-Verlag, April 2011.
(Cited on pages 3, 6, and 17.)

[Mau13] Ueli Maurer. Conditional equivalence of random systems and indistinguishability proofs. In *2013 IEEE International Symposium on Information Theory Proceedings (ISIT)*, pages 3150–3154, July 2013.
(Cited on pages 15 and 16.)

[MR91] Silvio Micali and Phillip Rogaway. Secure computation (abstract). In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *LNCS*, pages 392–404, Heidelberg, 1991. Springer.
(Cited on page 11.)

[MR11] Ueli Maurer and Renato Renner. Abstract cryptography. In Bernard Chazelle, editor, *The Second Symposium on Innovations in Computer Science, ICS 2011*, pages 1–21. Tsinghua University Press, January 2011.
(Cited on pages v, ix, 3, 6, 12, 13, and 15.)

[MRT12] Ueli Maurer, Andreas Rüedlinger, and Björn Tackmann. Confidentiality and Integrity: A Constructive Perspective. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 209–229, Heidelberg, 2012. Springer.
(Cited on page 12.)

[MS78] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. North-holland Publishing Company, 2nd edition, 1978.
(Cited on page 24.)

[MS09] Steven Myers and Abhi Shelat. Bit encryption is complete. In *FOCS*, pages 607–616, 2009.
(Cited on pages vi, x, 2, 12, 60, and 74.)

[MSS12] Steven Myers, Mona Sergi, and Abhi Shelat. Blackbox construction of a more than non-malleable CCA1 encryption scheme from plaintext awareness. In *Security and Cryptography for Networks - 8th International Conference, SCN 2012,*

Amalfi, Italy, September 5-7, 2012. Proceedings, pages 149–165, 2012.
(Cited on page 2.)

[MT10] Ueli Maurer and Björn Tackmann. On the soundness of authenticate-then-encrypt: Formalizing the malleability of symmetric encryption. In Angelia D. Keromytis and Vitaly Shmatikov, editors, *Proceedings of the 17th ACM Conference on Computer and Communication Security*, pages 505–515. ACM, ACM, October 2010.
(Cited on page 18.)

[MTC13] Ueli Maurer, Björn Tackmann, and Sandro Coretti. Key Exchange with Unilateral Authentication: Composable Security Definition and Modular Protocol Design. Cryptology ePrint Archive, Report 2013/555, 2013.
(Cited on pages 2 and 33.)

[PSV06] Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Construction of a non-malleable encryption scheme from any semantically secure one. In *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, pages 271–289, 2006.
(Cited on pages 8 and 43.)

[PSV07] Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Relations among notions of non-malleability for encryption. In *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, pages 519–535, 2007.
(Cited on page 8.)

[PW01] Birgit Pfitzmann and Michael Waidner. A Model for Asynchronous Reactive Systems and its Application to Secure Message Transmission. In *IEEE Symposium on Security and Privacy*, pages 184–200, 2001.
(Cited on pages 11 and 12.)

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A Method for Obtaining Digital Signatures and Public-Key

Cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
(Cited on page 1.)

[She93] Ba-Zhong Shen. A justesen construction of binary concatenated codes that asymptotically meet the zyablov bound for low rate. *IEEE Transactions on Information Theory*, 39(1):239–242, 1993.
(Cited on page 100.)

Curriculum Vitae

Sandro Coretti
Citizen of Bregaglia GR, Switzerland
Born on 23 December 1986, in Samedan GR, Switzerland

Doctoral Studies *06/2010 - present*
ETH Zurich, Department of Computer Science
Thesis title: *Non-Malleable Codes and Public-Key Encryption*
Advisor: Prof. Dr. Ueli Maurer
Co-examiner: Prof. Dr. Yevgeniy Dodis
Degree: Doctor of Sciences (Dr. sc. ETH)

Undergraduate Studies *10/2005 - 05/2010*
ETH Zurich, Department of Computer Science
Thesis title: *Constant-Round Asynchronous Multi-Party Computation*
Degree: Master of Science ETH in Computer Science

High School *08/1999 - 05/2005*
School: Academia Engiadina, Samedan GR, Switzerland