Restricted types of malleability

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Restricted Types of Malleability

Master Thesis

A. Rüedlinger

April 19, 2011

Advisors: Prof. Dr. U. Maurer, B. Tackmann

Department of Computer Science, ETH Zürich

Abstract

An encryption scheme is a cryptographic protocol ensuring confidentiality of communication. At the same time, encryption does not generally protect the integrity of the resulting encrypted communication. The potential ability of an adversary to modify the encrypted communication is called the *malleability* of the encryption scheme.

Several different notions of security have been introduced to capture not only confidentiality of an encryption scheme, but also to restrict its malleability. This set of notions was mainly formalized using a game-based approach and includes notions like *non-malleability* [17], *integrity of plaintexts* [7], or *plaintext-uncertainty* [19]. However, the model in which they are presented allow no intuitive and meaningful comparison of the notions, and their practical relevance often remains unclear.

A different model aiming at an abstract and natural way of defining security of cryptographic protocols is considered in the context of constructive cryptography [26, 29]. In this model, communication is abstracted as a channel and cryptographic protocols are considered as transformation of a particular type of channel into a more secure channel. This model also allows to describe the malleability properties of an encryption scheme in a meaningful way by a set of transformations that can be applied to a sent message.

In this thesis, I adapt several game-based security notions to the model of constructive cryptography, describing the corresponding restricted type of malleability as a set of transformations on the plaintext space. The resulting types of malleability are compared and examined with regard to their practical meaning for the use in secure communication. As a result of the analysis of the different types of malleability, I translate the natural definition of confidentiality (i.e. constructing a confidential channel) from the constructive framework to the game-based model and define "pure" confidentiality based on a game, resulting in a new game-based notion.

As a further application of the constructive model of malleability, I specify a restricted type of malleability that is sufficient to render the encryption-with-redundancy composition secure. In other words, an encryption scheme that conforms to this type of malleability can be combined with a keyless redundancy mechanism to guarantee both confidentiality and authenticity of communication. As it is shown in [2], such a sufficient condition is not provided by the traditional game-based security notions.

Contents

Chapter 1

Introduction

Secure communication is one of today's most important concepts in cryptography. In applications, such as online banking or e-commerce, or when sending sensitive data over the public internet, one would like to assume *secure* point-to-point connections. The term *secure* usually refers to *confidentiality* and *authenticity*, meaning that the communication cannot be read by a third party, and that a received message cannot originate from someone else than the legitimate sender.

Since this security is usually not ensured by standard communication channels, such as communication over the internet, cryptographic encryption schemes and authentication mechanisms, or a combination of both, are typically employed to secure communication.

This thesis mainly focuses on encryption schemes and their contribution to the overall goal, that is, secure and confidential communication.

1.1 Security of Encryption Schemes

Research in cryptography has a long history of studying and discussing the security properties of encryption schemes. The traditional approach, which is adopted in numerous papers, is to define security properties in a game-based model where an adversary plays a game with a (hypothetical) "challenger" and the scheme satisfies the property if no adversary can win the game with substantial probability.

The first game-based security notions were proposed for public-key encryption schemes where messages to a certain receiver are encrypted by using the same publicly known key by any party, and ciphertexts are decrypted by using the corresponding private key that is only known to the receiver.

Goldwasser and Micali [21] introduced *semantic security* as a way of formalizing the confidentiality of a public-key encryption scheme. Semantic security means that any information that an adversary can obtain from a ciphertext about the corresponding plaintext can also be obtained without knowledge of the ciphertext. Intuitively, semantic security means that the ciphertext leaks no information about the plaintext.

A notion that is related to semantic security is *indistinguishability*, also aiming at defining confidentiality by using the game-based approach. Many different definitions of this were introduced in the literature. However, these definitions converge on the idea that an adversary is unable to distinguish between two different types of encryptions (e.g. the encryptions of two chosen messages or the encryption of a chosen message and a random message).

It was shown that indistinguishability is equivalent to both semantic security and a notion of privacy based on computational entropy [30, 20]. This equivalence and the simple formalization of indistinguishability induced people to conclude that indistinguishability is the *right* formalization of confidentiality [8].

Non-malleability, a somewhat stronger notion not only covering confidentiality, was introduced by Dolev, Dwork and Naor [17, 18]. The following scenario should be considered: An auctioneer distributes his public key and accepts bids in the form of an encrypted amount (signed by the bidder). Confidentiality of the encryption scheme prevents other bidders from learning the bidden amounts of their opponents. A perfectly confidential channel can however not prevent a malicious bidder from transforming the ciphertext of the opponent's bid into a ciphertext which corresponds to a bid that is just slightly higher (e.g. increasing the bid by one dollar), signing it to be his own bid, and winning the auction. The notion of non-malleability disables this undesired behavior, i.e. that an adversary can generate a new ciphertext from a given ciphertext so that the corresponding plaintexts are "meaningfully related". The *malleability* of an encryption scheme is thus the potential adversarial influence on the outcome of the decryption that can be achieved by modifying given ciphertexts.

For secret-key encryption schemes, where a secret-key is shared among sender and receiver and used for encryption and decryption, both indistinguishability [5, 23, 7] and non-malleability [19, 23, 7] were adapted and game-based formalizations were provided.

1.2 The Weaknesses of the Game-Based Model

Defining security in the game-based model has a long tradition, which resulted in a large variety of different notions and definitions. However, these notions and definitions, as well as the game-based approach itself, have many disadvantages that cannot simply be justified by their popularity.

These disadvantages mainly arise from the fact that security is defined by the properties of the primitives that a constructed functionality is based on (e.g. the indistinguishability property of an encryption scheme), rather than by the security of the functionality itself (e.g. confidentiality of communication). This leads to unnecessary complexity of the definitions and different versions of game-based definitions for the same security properties of the functionality.

The drawbacks of the game-base approach can be illustrated by means of the notion of indistinguishability, whose goal is to capture confidentiality. On the level of constructed functionality, i.e. secure communication, the security goal is clear and simple: Communication does not leak any information about sent messages. On the level of the encryption scheme, this is definable in various ways. In [5], four versions of indistinguishability, differing in the way the adversary is challenged, are defined and proved to be equivalent. Other variations in the definitions are due to the number of challenge queries that are allowed ([5, 23] vs. [7]), or the different types of encryption schemes considered (stateless vs. stateful, allowance of message replay [31]). In a addition to the non-trivial transition from the game-based definition to a security statement for the functionality, it is a rather complex and subtle undertaking to compare different game-based definitions and notions. On the level of secure communication, the meaning of numerous observations and results about indistinguishability definitions remains rather unclear.

Moreover, another problem is that a seemingly minor detail of a game-based definition can lead to a significant change of the security statement on the functionality level. The example of indistinguishability is considered once more: If one restricts the challenge encryption to messages of equal length, the security statement translates into confidential communication, leaking no information about the sent messages *except for their length*. Without this restriction, no information is leaked and communication is perfectly confidential.

The problem of "translating" statements from the functionality level to the level of the primitive also affects the model of the adversary. For functionality, like secure communication, different models of adversaries exist (e.g. only passive attacker or the Dolev-Yao model [16]). In a game-based approach, the capabilities of an adversary are specified in an *attack model*. However, the translation between the levels for such adversary models is

not clear, which is why there is no simple model correspondence between the levels that is widely accepted. It is often argued that the strongest possible attack model should be used because the more powerful the adversary is in the game, the more secure is the encryption scheme that meets the requirements [13, 14]. This paradigm to use the strongest possible security property for the primitive can, however, suppress the fact that in order to achieve the strongest possible security for the constructed functionality (e.g. secure communication), a weaker variant of security would suffice for the primitive (e.g. the encryption scheme) [12], and possibly result in a more efficient encryption scheme.

An additional drawback of the game-based approach is that guarantees provided by a game-based notion are not generically preserved under composition. As pointed out by Krawczyk [24], combining an encryption scheme that satisfies a game-based definition of indistinguishability with an authentication scheme does necessarily result in an authenticated encryption scheme that still satisfies the game-based properties of each component. The authenticate-then-encrypt composition paradigm (AtE) is an example of how the complexity of such a composition is underestimated and not analyzed properly. For many years, AtE was assumed to preserve the security properties of the composed schemes and was used in practice (TLS [15]), before Krawczyk [24] disabused the community.

In conclusion, security definitions in the game-based model are hard to compare and often have an unclear meaning for the scenario in which they are applied. Therefore, a more promising approach is to define security on the level of constructed functionality and so that composition preserves the defined security properties.

1.3 Changing Perspective

With regard to the disadvantages mentioned in the previous section, one can question whether the traditional game-based security modeling is the right approach to take. One could argue that it would be more natural to define security by changing the viewpoint and defining the security properties of the constructed functionality itself. For secure communication, this means that one defines what communication must fulfill in order to be called secure.

Constructive cryptography, introduced by Maurer [26], follows this approach. To be more precise, a security goal is characterized by an ideal functionality (e.g. a secure channel allowing two honest parties to communicate with each other). The term secure refers to the abilities (or inabilities) of the adversary in the ideal functionality (e.g. a secure channel does not leak any

information[1] to the adversary and does only allow forwarding or deleting messages). A cryptographic protocol (e.g. an encryption scheme) is seen as a transformation of a given functionality into a stronger functionality. Such a protocol is thus called secure if it transforms an insecure functionality into the ideal, secure functionality that was specified. The formal definition of this type of security will be given in Chapter 2.

A similar approach is taken in [33, 3], introducing reactive simulatability, and in Canetti's UC framework [11] .

Following the ideas of constructive cryptography, Chapter 3 introduces how ideal functionalities should be modeled in the case of secure communication. As the corresponding primitives are communication channels, the basic functionality is a channel without security guarantees, called *insecure channel*. Capturing the notions of authenticity, confidentiality as well as the combination of both in an ideal functionality, the *authenticated channel*, the *confidential channel* and the *secure channel* are defined. Especially in the case of the confidential channel, a new definition that covers pure confidentiality and does—in contrast to existing game-based notions—not restrict the malleability in any way, is given. The malleability of a channel is formalized on the basis of the definition of malleability given in [29], allowing the categorization of confidential channels according to the type of malleability allowed.

The system concept, introduced by Maurer and Renner in [28], provides the framework for the discussions of definitions, statements and proofs in this thesis. Starting on a very abstract level, a system simply has a set of interfaces through which systems can be connected in order to establish a new composed system. A communication channel is thus a system providing an interface to the sender A, one to the receiver B and an additional one to the adversary E.

On a second, more concrete level, systems do not only have interfaces but also can communicate via the interfaces with each other by taking inputs and producing outputs in discrete steps, often called rounds. For example, a simple communication channel, without an adversary being present, takes as input a message at the sender interface A and produces the message as output at the receiver interface B.

As mentioned in [28], statements should be made on the highest level of abstraction possible. On the one hand, a theorem proven on such a high level is completely precise and, on the other hand, very powerful as it holds for any instantiation on a lower-level that complies with the requirements made in the theorem (such as required properties or axioms). In this thesis,

[1] Except for the length of the message.

most of the statements and proofs are made on the level of discrete systems, using the formal model of random systems introduced in [25].

1.4 Formalizing Restricted Types of Malleability

The main aim of this thesis is to examine confidentiality and non-malleability related security notions, with a special focus on the malleability characteristics. Section 1.1 underlined the need to examine notions that are stronger than confidentiality on the example of an online auction. Such notions (e.g. non-malleability) are also interesting in the case of secret-key cryptography. Consideration should be given to the authenticate-then-encrypt (AtE) composition paradigm, where security is established by encrypting an authenticated message. It has been shown that confidentiality alone does not suffice to render the composition secure as the encryption scheme could allow a type of malleability that can be exploited to break the whole system [24, 7]. An interesting question in this context is what kind of malleability is sufficient for AtE to be sound [29].

Chapter 4 presents a selection of game-based security notions from the literature that address non-malleability and integrity properties of encryption schemes. The game-based notions are defined using the formalization of a game as a random system [27] and they are characterized in an unified description framework using pseudo-code.

In order to both have a meaningful statement of each game-based notion and to be able to compare the notions among each other, they are "translated" into channel-based security statements. For each game-based notion, I specify a restricted type of malleability and show that an encryption scheme is secure with respect to the notion if, and only if, it constructs a confidential channel with the specified type of malleability.

The first considered notion is *non-malleability* [17]. I show that a scheme which is secure in the sense of non-malleability[2] allows the following type of malleability: An adversary can forward, replay and delete messages and insert "constant" messages (that are independent of the sent messages). In the case of public-key cryptography, the term non-malleability seems to be appropriate as this type of malleability is the strongest that is achievable (inserting constant messages can always be achieved by encrypting such a message using the known public key). In the case of secret-key cryptography, using the term non-malleability is unfortunate and it seems interesting to investigate even more restricted types of malleability.

[2]The naming may seem counter-intuitive as a non-malleable scheme actually can be malleable.

One type of notions that refers to a more restricted type of malleability is the one addressing the integrity of encryption. Definitions of such notions are given in [7, 19] in the form of *plaintext integrity*, in [7] *ciphertext integrity* is introduced, in [22] *existential unforgeability* is defined and *existential forgery* is given in [19]. I will show that all these definitions capture essentially the same security statement on the level of secure communication, that is, that a scheme secure in their sense constructs a confidential channel that allows no malleability at all, i.e. a secure channel.

In [19], a set of additional notions concerning the integrity of encryption is proposed. I will deal with the two closely related notions of *plaintext-uncertainty* and *chosen-plaintext forgery* and show that the corresponding confidential channel allows malleability where the result of the modification done by the adversary must be unpredictable in a computational sense.

In the process of showing the results for plaintext-uncertainty and chosen-plaintext forgery, a new attack model for game-based notions that does not restrict the malleability of the channel induced by such a notion is introduced. While traditional, existing game-based definitions, such as those of indistinguishability, are always coupled with such a restriction to the malleability, the new attack model leads to a game-based definition of pure confidentiality.

Chapter 5 proposes a new restricted type of malleability called *subset-blurring*. The purpose of this type of malleability is that an encryption scheme that constructs a subset-blurring confidential channel can be used in a general AtE composition, called encryption-with-redundancy (EwR), such that the composition with a public redundancy code constructs a secure channel. As outlined out in [2], such a sufficient condition for a sound EwR composition with keyless authentication (i.e. publicly known redundancy) cannot be formalized using the traditional game-based notions.

1.5 Related Work

Non-malleability of public-key schemes Non-malleability notions and malleability characteristics of encryption schemes were so far primarily studied using game-based models. The notion of non-malleability was introduced by Dolev, Dwork and Naor [17, 18] using a simulation-based formalization.

In [6], both indistinguishability and non-malleability definitions are given and related to each other. For non-malleability, a new indistinguishability-based definition is given.

Bellare and Sahai, in [8, 9], present the equivalence of the two definitions of non-malleability. By doing so, they come up with another definition, a

"pure" indistinguishability definition, that is shown to be equivalent to the other two definitions. In [32], the comparison of the different definitions is extended to less restricted models and equivalence relations and separations are examined with regard to specific adversary types.

Relaxing the attack model The equivalence of non-malleability and indistinguishability (or semantic security) in the strongest attack model (CCA) was pointed out in [18, 6]. In [12], it is argued that CCA is too strict and that a weaker attack model is sufficient "for most practical purposes". They introduce the *replayable chosen-ciphertext attack model* (RCCA) and corresponding definitions of indistinguishability and non-malleability are given. The equivalence of the two definitions is shown to hold true even under this type of attack model. Weaker attack models than CCA that are still sufficient "for practical purposes" have also been studied in [35, 31, 1].

Notions for secret-key schemes The first study of game-based notions for the secret-key case was provided in [5]. Definitions of indistinguishability against chosen-plaintext attacks are given and the relations are analyzed. Since CPA is a weaker attack model than the one considered here, these definitions are not considered in this thesis.

Katz and Yung [22] introduce a new notion called *encryption unforgeability*, addressing the integrity property of secret-key encryption.

In [19], non-malleability was adapted to the setting of secret-key schemes for the first time. The stated definition of non-malleability is, however, not fully precise and considers only weak attack models. In addition to non-malleability, many other definitions addressing the integrity of encryption that potentially capture interesting malleability characteristics of encryption were proposed.

In [23], both indistinguishability and non-malleability are examined in detail and similar results about the relations, as in the public-key case, are stated. Bellare and Namprempre [7] made a similar analysis of the two notions using concrete security statements. Additionally, two definitions addressing the integrity of encryption were introduced and the individual notions were examined with respect to the preservation of security under the three main composition paradigms (EtA, AtE, E&A).

An and Bellare [2] introduce the *encryption-with-redundancy paradigm* where an encryption scheme is used along with either public redundancy (e.g. a hash of the message) or with secret redundancy (e.g. a MAC). They further examine the authentication properties of encryption schemes that satisfy the properties of existing notions and that are extended according to their paradigm.

Constructive cryptography The paradigm of constructive cryptography was introduced by Maurer [26]. Maurer and Tackmann [29] applied the paradigm to formalize the malleability of an encryption scheme precisely. Further, they not only showed a malleability condition that is sufficient for the AtE composition to be sound, but also that encryption in TLS satisfies this condition.

1.6 Future Work

In this thesis, I express game-based security notions in the channel-based framework of constructive cryptography where the malleability of an encryption scheme is formalized as a set of transformations on the plaintext history. This results in a collection of restricted types of malleability. An interesting question is whether other types that are not captured by game-based security notions exist, which are useful and of practical relevance (e.g. a type that is both necessary and sufficient for AtE to be sound).

Particularly in the case of *plaintext-uncertainty*, the given formalization is still very close to the original game-based definition. It remains unclear if there is a more natural way of defining the type of malleability, or even if an information-theoretic definition, based on min-entropy properties of the transformations, results in a more useful type of malleability.

Regarding the introduced game-based definition of "pure" confidentiality, the given formalization "borrows" concepts from the constructive model and thus differs slightly in style from traditional game-based notions. It would be interesting to analyze if this difference is inherent or if there exists a definition that is closer to the traditional games.

Chapter 2

Model and Notation

This chapter introduces the model and notation used in this thesis. The model considered here is an adapted version of the theory of systems introduced by Maurer and Renner in [28, Section 6], a system model defining different layers separated by their level of abstraction. As opposed to the bottom-up approach widely used in cryptography, they use a top-down approach, starting at the most abstract level. Therefore, they avoid many technical details and are able to state more general results. For a more detailed comparison of the top-down and the bottom-up approach, see [28, Section 1.4].

In the following, the two most principal levels of abstraction of this approach are introduced, namely the level of abstract systems and the level of discrete systems. The concepts and definitions are adapted from [28] and the lecture notes on cryptography of Maurer [27] in order to suit the needs of this thesis.

2.1 Abstract Systems

On the highest level of abstraction, the focus lies on abstract systems that are characterized solely by their set of interfaces. Let $\mathcal{I}$ be the set of interfaces that characterizes a system. Such a system is called an *$\mathcal{I}$-system*. A system—both abstract and discrete—will in the following be denoted by a bold-face capital letter, e.g. $\mathbf{S}$.

Systems can be connected via their interfaces to build new systems. The topology of such composed systems is the main focus on this level. Two systems, $\mathbf{S}$ and $\mathbf{T}$, can be composed by connecting an interface of one system, e.g. the interface i_S of $\mathbf{S}$, with an interface of the other system, e.g. interface i_T of $\mathbf{T}$, resulting in the system denoted by $\mathbf{S}^{i_S - i_T}\mathbf{T}$, or simply $\mathbf{ST}$ if it is clear

which interfaces are connected. This kind of composition is called *sequential composition*.

Parallel composition of systems is defined if the two systems to be composed have the same set of interfaces $\mathcal{I}$. The parallel composition of two $\mathcal{I}$-systems, $\mathbf{S}$ and $\mathbf{T}$, is denoted by $\mathbf{S}\|\mathbf{T}$. The result of such a composition is the $\mathcal{I}$-system where each interface (of $\mathbf{S}\|\mathbf{T}$) provides parallel access to the corresponding interfaces of both $\mathbf{S}$ and $\mathbf{T}$.

In the considered model, compositions are assumed to have a fundamental property called *composition-order independence*. This property is a kind of generalized associativity for the operation of sequential composition.

2.1.1 Resources and Converters

In the following, the focus is restricted to two special types of systems, i.e. resources and converters.

Resources A *resource system* (or simply *resource*) with interface set $\mathcal{I}$, also called $\mathcal{I}$-resource, is a system with $|\mathcal{I}|$ interfaces labeled by elements of $\mathcal{I}$. In every consideration, the set of interfaces is fixed and the set of resources is thus denoted in situ by $\mathcal{R}$. Moreover, an equivalence relation $\equiv$ is assumed to be defined on the set $\mathcal{R}$ (this relation has to be made explicit on a lower level).

An $\mathcal{I}$-resource with $\mathcal{I} = \{1, \ldots, n\}$ models systems that enable an n-party interaction where each interface i is intended to be accessed by party i. For the example of secure communication, resources with the interface set $\mathcal{I} = \{A, B, E\}$ are considered. The interfaces A and B are accessible to the honest parties Alice and Bob respectively, and the interface E is accessible to the adversary Eve. Examples for resources in this context are an (insecure) communication channel or a shared secret key.

Converters A *converter system* (or simply *converter*) is a system with two interfaces, one is designated as the inside interface and the other one as the outside interface. Converters are denoted by a bold-face capital letter (e.g. $\mathbf{C}$) or by a small greek letter (e.g. π, φ, σ). The inside interface of a converter $\mathbf{C}$ can be connected to an interface $i \in \mathcal{I}$ of a $\mathcal{I}$-resource $\mathbf{S}$. The outside interface of $\mathbf{C}$ serves as the new interface of the combined system, which is again a resource of the same type as $\mathbf{S}$ and is denoted by $\mathbf{C}^i\mathbf{S}$, or simply $\mathbf{CS}$ if i is the only interface of $\mathbf{S}$. The set of converters is denoted by Σ.

Properties of Resources and Converters The following *properties* are assumed to hold for the sets $\mathcal{R}$ and Σ and for the mapping $\Sigma \times \mathcal{R} \times \mathcal{I} \to \mathcal{R}$ defining the attachment of a converter to the interface of a resource. In the following conditions, the set $\mathcal{R}$ contains $\mathcal{I}$-resources, α and β stand for any converter in the set Σ, i and j for any interface in $\mathcal{I}$, and $\mathbf{R}$ and $\mathbf{S}$ for any resource in the set $\mathcal{R}$:

(i) The order of applying converters at different interfaces is irrelevant, $\alpha^i \beta^j \mathbf{S} \equiv \beta^j \alpha^i \mathbf{S}$, $i \neq j$.

(ii) There is a special converter in the set Σ, denoted by $\mathbf{1}$, a dummy converter that refers to applying no converter at all: $\mathbf{1}^i \mathbf{S} \equiv \mathbf{S}$.

(iii) Attaching converters from Σ preserves the equivalence relation: If $\mathbf{R} \equiv \mathbf{S}$, then $\alpha^i \mathbf{R} \equiv \alpha^i \mathbf{S}$.

(iv) Serial composition of two converters α, β, denoted by $\alpha \circ \beta$, is defined by $(\alpha \circ \beta)^i \mathbf{S} := \alpha^i \beta^i \mathbf{S}$. The set Σ is closed under serial composition, $\alpha \circ \beta \in \Sigma$.

(v) Parallel composition of two converters α, β, denoted by $\alpha \| \beta$, is defined by $(\alpha \| \beta)^i (\mathbf{R} \| \mathbf{S}) := (\alpha^i \mathbf{R}) \| (\beta^i \mathbf{S})$. The set Σ is closed under parallel composition, $\alpha \| \beta \in \Sigma$.

(vi) For a resource $\mathbf{R}$, there exists a special type of converter in Σ, called *resource converter*, emulating the resource $\mathbf{R}$ and making it available in parallel to the system it is connected to at the inner interface. Such a resource converter is denoted by $(\mathbf{R} \| \cdot)$ (or $(\cdot \| \mathbf{R})$ respectively) and defined by $(\mathbf{R} \| \cdot)\mathbf{S} := (\mathbf{R} \| \mathbf{S})$.

2.1.2 Pseudo-metrics

In order to be able to express how similar resources are, a pseudo-metric is defined on the set of resources $\mathcal{R}$.

Definition 2.1. *A* pseudo-metric *on the set of resource systems $\mathcal{R}$, denoted by d, is a function $d : \mathcal{R} \times \mathcal{R} \to \mathbb{R}$ with the following properties (for any resources $\mathbf{R}, \mathbf{S}, \mathbf{T} \in \mathcal{R}$):*

(i) $d(\mathbf{S}, \mathbf{S}) = 0$,

(ii) $d(\mathbf{S}, \mathbf{T}) = d(\mathbf{T}, \mathbf{S})$,

(iii) $d(\mathbf{R}, \mathbf{T}) \leq d(\mathbf{R}, \mathbf{S}) + d(\mathbf{S}, \mathbf{T})$.

An important property of a pseudo-metric is closure under composition with converters.

Definition 2.2. *A pseudo-metric d is called* closed under composition with converters Σ *if for any two resources* $\mathbf{R}, \mathbf{S} \in \mathcal{R}$*, any converter* $\alpha \in \Sigma$ *and any interface* $i \in \mathcal{I}$*,*

$$d(\alpha^i \mathbf{R}, \alpha^i \mathbf{S}) \leq d(\mathbf{R}, \mathbf{S}).$$

2.1.3 Defining Security

Let $\mathcal{R}$ be the set of $\{A, B, E\}$-resources and Σ be the set of converters according to the properties from Section 2.1.1, and let d be a pseudo-metric according to Definition 2.1. Security is defined following the ideas of constructive cryptography for the case of secure communication by adapting the definition from [29].

The intuition behind the upcoming definition of security is as follows: A "real" resource $\mathbf{R}$ is specified, being a reasonable model of the resource used in practice. The ideal resource $\mathbf{S}$ captures the behavior of a resource one would like to have. A pair of converters $\pi = (\pi_1, \pi_2)$, called a *protocol*, is employed at the honest interfaces A and B of the real resource with the idea of transforming it into a more secure resource, called *transformed resource*, that is similar or equivalent to the ideal one. The protocol π is considered *secure* if anything that can be achieved interacting with the transformed resource $\pi_1^A \pi_2^B$ $\mathbf{R}$ can also be achieved interacting with the ideal resource $\mathbf{S}$. In order to make this informal statement more precise, a special converter σ that is applied at the E-interface of the ideal resource $\mathbf{S}$ to translate between the E-interfaces of the two resources, called the *simulator*, is introduced. It has to be mentioned that the application of a simulator can only strengthen the ideal system since its behavior can always be emulated by the adversary.

In order to measure the similarity of the transformed resource and the ideal resource, the pseudo-metric d is used. While equivalence of the transformed resource and the ideal resource translates into a distance between the two resources of 0, the state of "being similar" is reflected by a "small" distance between the two resources.

Definition 2.3. *A protocol* $\pi = (\pi_1, \pi_2)$ securely constructs resource $\mathbf{S}$ from $\mathbf{R}$ with error ε*, if there exists a converter* $\sigma \in \Sigma$ *such that the distance between the real resource—to which* π_1 *and* π_2 *are attached via the interfaces A and B—and the ideal resource—to which* σ *is attached via the E-interface—is bounded by* ε *according to the pseudo-metric* d*:*

$$\exists\, \sigma \in \Sigma : d(\pi_1^A \pi_2^B \mathbf{R},\ \sigma^E \mathbf{S}) \leq \varepsilon.$$

2.1.4 Step-wise Refinement and Composability

Step-wise refinement is a very natural and intuitive paradigm in cryptography and other constructive disciplines. Following this paradigm, one constructs complex systems from simpler components or modules (sometimes called primitives). Each of these components can be constructed from even simpler primitives, and so on. The overall construction consisting of a sequence of refinements is, however, only useful if the relevant properties of each individual refinement step are *composeable*. This means that the properties of a single step are preserved in the process of composing several refinement steps and the overall construction still has those properties.

In the context of cryptography, the definition of *securely realizing an ideal resource from a real resource* is an example of such a relevant property of a refinement step. The fundamental property of composability holds for the security definition stated in the previous section. More precisely, composability of the security definition means that whenever an ideal resource $\mathbf{S}$ is used in the composition of a new resource, it can safely be replaced by the resource $\pi_1^A \pi_2^B \, \mathbf{R}$ in this composition.

In the following, Theorem 1 of [29] is recalled using the notation of the introduced model. The theorem states that under the assumption that the considered pseudo-metric d is closed under composition with converters Σ, security is composable for both the sequential composition of converters and for the parallel composition of resources.

Theorem 2.4 (Composability). *Let* $\mathbf{R}$, $\mathbf{S}$, $\mathbf{T}$ *and* $\mathbf{U}$ *be resources from* $\mathcal{R}$ *and let* $\pi = (\pi_1, \pi_2)$ *and* $\varphi = (\varphi_1, \varphi_2)$ *be protocols so that* π *securely constructs* $\mathbf{S}$ *from the resource* $\mathbf{R}$ *with error* ε_π *and* φ *securely constructs* $\mathbf{T}$ *from* $\mathbf{S}$ *with error* ε_φ.

If the pseudo-metric d *is closed under composition with converters* Σ *according to Definition 2.2, then* $(\varphi_1 \circ \pi_1, \varphi_2 \circ \pi_2)$ *securely constructs* $\mathbf{T}$ *from* $\mathbf{R}$ *with error* $\varepsilon_\pi + \varepsilon_\varphi$, $(\pi_1 \| \mathbf{1}, \pi_2 \| \mathbf{1})$ *securely constructs* $\mathbf{S} \| \mathbf{U}$ *from* $\mathbf{R} \| \mathbf{U}$ *with error* ε_π *and* $(\mathbf{1} \| \pi_1, \mathbf{1} \| \pi_2)$ *securely constructs* $\mathbf{U} \| \mathbf{S}$ *from* $\mathbf{U} \| \mathbf{R}$ *with error* ε_π.

The proof of the Theorem can be found in [29, Theorem 1].

2.2 Discrete Systems

On the second level of abstraction, i.e. the level of discrete systems, the *behavior of systems* is focused on in addition to the structure of composed systems. Not only do systems have sets of interfaces, but their behavior is modeled by an explicit communication via the interfaces. The model in which this communication is considered is a round-based model: In each round, a system receives an input at a single interface from a discrete input

alphabet and produces an output at a single interface from a discrete output alphabet before handling the next input.

A discrete system is thus additionally characterized by the discrete input and output alphabets at each interface, as well as the probability distributions of the output given the input and output history.

2.2.1 $(\mathcal{X}, \mathcal{Y})$-Systems

On the level of discrete systems, statements about systems depend only on the observable *input-output behavior* of the system. Therefore, a system can be seen as a black-box producing, on an input X from an alphabet $\mathcal{X}$, an output Y from another alphabet $\mathcal{Y}$. If the system has multiple interfaces, the input X and the output Y can be seen as containing an identifier of the interface the input is delivered to or the interface the output is produced at respectively. A resource can thus be defined as a $(\mathcal{X}, \mathcal{Y})$-system.

Definition 2.5. *An* $(\mathcal{X}, \mathcal{Y})$-system *takes inputs* $X_1, X_2, \ldots$ *(from some discrete alphabet* $\mathcal{X}$*) and generates, for each new input* X_i*, an output* $Y_i \in \mathcal{Y}$*. The output* Y_i *depends (possibly probabilistically) on the current input* X_i *and on the internal state.*

Two discrete systems with the same input-output behavior are considered to be *equivalent*. Equivalence in this sense implies that two such systems behave identically in any environment they are plugged into. Characterizing a discrete system by conditional probability distributions allows to formalize this idea of the equivalence relation. In the following, the notation X^i, where X is a random variable and $i \geq 1$, denotes the tuple $(X_1, \ldots, X_i)$.

Definition 2.6. *An* $(\mathcal{X}, \mathcal{Y})$-random system $\mathbf{F}$ *is a (possibly infinite) sequence of conditional probability distributions* $\mathrm{p}^{\mathbf{F}}_{Y_i|X^iY^{i-1}}$ *for* $i \geq 1$.

Definition 2.7. *Two resources* $\mathbf{R}$ *and* $\mathbf{S}$ *are* equivalent, *denoted*

$$\mathbf{R} \equiv \mathbf{S},$$

if they correspond to the same random system, i.e.

$$\mathrm{p}^{\mathbf{R}}_{Y_i|X^iY^{i-1}} = \mathrm{p}^{\mathbf{S}}_{Y_i|X^iY^{i-1}}, i \geq 1,$$

where the equality of two probability distributions $\mathrm{p}^{\mathbf{R}}_{Y_i|X^iY^{i-1}}$ *and* $\mathrm{p}^{\mathbf{S}}_{Y_i|X^iY^{i-1}}$ *is defined by the equality of the probabilities for any values of the random variables.*

Therefore, all the considered resources are seen as random systems in this thesis, following the viewpoint of [25]. Although a random system can be described in many ways, a kind of pseudo-code that is introduced in Section 2.3 will be used in this thesis.

2.2.2 Distinguishers

The concepts of a *distinguisher* and of the corresponding *distinguishing advantage* provide a concrete definition of a pseudo-metric on the level of discrete systems. A distinguisher is a system that can be connected to resources and that outputs, after a certain number of rounds of interaction with the resource, a *distinguishing bit* W. The distinguishing bit can be regarded as the distinguisher's guess to which of two resources in question he is connected to.

In order to keep the complexity of definitions and proofs reasonable, resources that are connected to distinguishers are assumed to be *single-interface systems* in this thesis. If this is not the case by default, one can merge the interfaces of the resource by letting all of them be accessed via the same (single) interface. This simplification allows a distinguisher to be seen as a converter where the inside interface can be connected to a resource and the outside interface provides the distinguishing bit W.

Definition 2.8. *A* distinguisher $\mathbf{D}$ for $(\mathcal{X}, \mathcal{Y})$-systems *is a converter which, at the inside interface, behaves like a $(\mathcal{Y}, \mathcal{X})$-system that is one query ahead, meaning that it is defined by $\mathsf{p}^{\mathbf{D}}_{X_i|Y^{i-1}X^{i-1}}$ (instead of $\mathsf{p}^{\mathbf{D}}_{X_i|Y^iX^{i-1}}$) for all i. Moreover, there exists a number q of queries, after which the system outputs a bit W at the outside interface, based on the transcript (X^q, Y^q), according to a conditional distribution $\mathsf{p}^{\mathbf{D}}_{W|X^qY^q}$.*

To measure the distance between two resources $\mathbf{R}$ and $\mathbf{S}$, the following pseudo-metric is introduced:

Definition 2.9. *The* advantage *of a distinguisher $\mathbf{D}$ in distinguishing the resources $\mathbf{R}$ and $\mathbf{S}$, denoted by $\Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{S})$, is defined as*

$$\Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{S}) := \left| \mathsf{P}^{\mathbf{DR}}(W = 1) - \mathsf{P}^{\mathbf{DS}}(W = 1) \right|.$$

The advantage of a distinguisher class $\mathcal{D}$ *in distinguishing $\mathbf{R}$ and $\mathbf{S}$, denoted by $\Delta^{\mathcal{D}}(\mathbf{R}, \mathbf{S})$, is defined as*

$$\Delta^{\mathcal{D}}(\mathbf{R}, \mathbf{S}) := \sup_{\mathbf{D} \in \mathcal{D}} \Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{S}).$$

It has to be mentioned that the given definition indeed satisfies the postulated conditions of a pseudo-metric.

Remark 2.10. *For any distinguisher $\mathbf{D}$ and any distinguisher class $\mathcal{D}$, $\Delta^{\mathbf{D}}$ and $\Delta^{\mathcal{D}}$ are pseudo-metrics on the resource set $\mathcal{R}$ according to Definition 2.1.*

The condition for closure of the pseudo-metric under composition is implied in the case of Definition 2.9 by the condition that the class of distinguishers is closed under composition.

Remark 2.11. *If a distinguisher class $\mathcal{D}$ is* closed under composition with converters Σ, *namely that for any distinguisher $\mathbf{D} \in \mathcal{D}$ and any converter $\mathbf{C} \in \Sigma$, $\mathbf{DC} \in \mathcal{D}$, then the advantage of the distinguisher class $\mathcal{D}$, the pseudo-metric $\Delta^{\mathcal{D}}$, is* closed under composition with converters Σ.

The following lemma states that equivalent resources cannot be distinguished by any distinguisher. Since equivalence of two systems is defined as identical input-output behavior of the two systems, and the input-output behavior is all that a distinguisher can observe, there is nothing that can be distinguished.

Lemma 2.12. *If two systems $\mathbf{F}$ and $\mathbf{G}$ are equivalent, $\mathbf{F} \equiv \mathbf{G}$, then any distinguisher $\mathbf{D}$ has the distinguishing advantage* 0,

$$\Delta^{\mathbf{D}}(\mathbf{F}, \mathbf{G}) = 0.$$

Proof. Recall that the distinguishing advantage is defined as

$$\Delta^{\mathbf{D}}(\mathbf{F}, \mathbf{G}) = \left| \mathsf{P}^{\mathbf{DF}}(W = 1) - \mathsf{P}^{\mathbf{DG}}(W = 1) \right|.$$

The probability that $\mathbf{D}$ outputs $W = 1$ interacting with $\mathbf{F}$ can be written as the sum, over all values of inputs/outputs at the inner interface of $\mathbf{D}$, of probabilities that $\mathbf{D}$ outputs $W = 1$ conditioned on that input/output at the inner interface, multiplied by the probability of that input/output in the system $\mathbf{F}$. Since the probability distributions of $\mathbf{F}$ and $\mathbf{G}$ are equal for all inputs/outputs due to their equivalence, all terms corresponding to an input/output probability in $\mathbf{F}$ can be replaced by corresponding terms for $\mathbf{G}$ and thus the probability that $W = 1$ is equal in both cases. □

2.2.3 Monotone Event Sequences

Considering $(\mathcal{X}, \mathcal{Y})$-random systems, it is sometimes useful to examine certain conditions on the input and output of such a system. Following Maurer [25], a (possibly infinite) sequence of binary random variables $\mathcal{A} = A_0, A_1, A_2, \ldots$ denoting the state of the condition in each round is called *event sequence*. An event sequence is called a *monotone event sequence (MES)*, if $A_i = 0$ implies $A_{i+1} = 0$ for $i \geq 1$. Therefore, a random system $\mathbf{F}$ with MES $\mathcal{A}$ can informally be seen as a system with (possibly) two states: an initial state where the condition described by $\mathcal{A}$ is satisfied, and a second state where the system fails to fulfill the condition. Once the system is in the second state, it remains like this for all future rounds due to the monotonicity of $\mathcal{A}$.

Often, one is only interested in the system as long as it is in the first state satisfying the condition, denoted by $\mathbf{F}|\mathcal{A}$.

Definition 2.13. *Let* $\mathbf{F}$ *be a random system with MES* $\mathcal{A} = A_0, A_1, A_2, \ldots$ *and let* $\mathbf{G}$ *be another random system. Then,* $\mathbf{F}$ conditioned on $\mathcal{A}$ is equivalent to $\mathbf{G}$, *denoted by* $\mathbf{F}|\mathcal{A} \equiv \mathbf{G}$, *if for* $i \geq 1$

$$\mathrm{p}^{\mathbf{F}}_{Y_i|X^iY^{i-1}A_i=1} = \mathrm{p}^{\mathbf{G}}_{Y_i|X^iY^{i-1}}.$$

Recall a part of Theorem 1 of [25] where it is shown that in order to distinguish random systems $\mathbf{F}$ and $\mathbf{G}$ with $\mathbf{F}|\mathcal{A} \equiv \mathbf{G}$ in k queries, a distinguisher must provoke the event $\bar{A}_k$ to have non-zero distinguishing advantage. Here the statement is adapted for a general distinguisher without an explicit bound on the number of queries. For this, let $\bar{A}_i$ denote the complementary event of A_i, and $\mathsf{P}^{\mathbf{DF}}(\mathcal{A}) := \sup_{i\geq 1} \mathsf{P}^{\mathbf{DF}}(A_i)$ be the probability of the event condition of $\mathcal{A}$ to be eventually true. The following Lemma is thus an adaption of [25, Theorem 1 (i)].

Lemma 2.14. *Let* $\mathbf{F}$ *be a random system with MES* $\mathcal{A}$ *and let* $\mathbf{G}$ *be a random system such that* $\mathbf{F}|\mathcal{A} \equiv \mathbf{G}$, *then for any distinguisher* $\mathbf{D}$,

$$\Delta^{\mathbf{D}}(\mathbf{F}, \mathbf{G}) \leq \mathsf{P}^{\mathbf{DF}}(\bar{\mathcal{A}}).$$

Proof. Using the triangle inequality and Lemma 2.12, we get

$$\begin{aligned}\Delta^{\mathbf{D}}(\mathbf{F}, \mathbf{G}) &\leq \Delta^{\mathbf{D}}(\mathbf{F}, \mathbf{F}|\mathcal{A}) + \Delta^{\mathbf{D}}(\mathbf{F}|\mathcal{A}, \mathbf{G}) \\ &= \Delta^{\mathbf{D}}(\mathbf{F}, \mathbf{F}|\mathcal{A}) + 0.\end{aligned}$$

Since either $\mathcal{A}$ or $\bar{\mathcal{A}}$ in any random experiment,

$$\begin{aligned}&\Delta^{\mathbf{D}}(\mathbf{F}, \mathbf{F}|\mathcal{A}) = \left| \mathsf{P}^{\mathbf{DF}}(W=1) - \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1) \right| \\ &= \left| \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1) \cdot \mathsf{P}^{\mathbf{DF}}(\mathcal{A}) + \mathsf{P}^{\mathbf{DF}|\bar{\mathcal{A}}}(W=1) \cdot \mathsf{P}^{\mathbf{DF}}(\bar{\mathcal{A}}) - \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1) \right|.\end{aligned}$$

If $\mathsf{P}^{\mathbf{DF}}(W=1) \geq \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1)$, we get

$$\begin{aligned}\Delta^{\mathbf{D}}(\mathbf{F}, \mathbf{F}|\mathcal{A}) &= \mathsf{P}^{\mathbf{DF}}(W=1) - \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1) \\ &= \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1) \cdot \mathsf{P}^{\mathbf{DF}}(\mathcal{A}) + \mathsf{P}^{\mathbf{DF}|\bar{\mathcal{A}}}(W=1) \cdot \mathsf{P}^{\mathbf{DF}}(\bar{\mathcal{A}}) \\ &\qquad - \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1) \\ &\leq \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1) \cdot 1 + 1 \cdot \mathsf{P}^{\mathbf{DF}}(\bar{\mathcal{A}}) - \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1) \\ &= \mathsf{P}^{\mathbf{DF}}(\bar{\mathcal{A}}).\end{aligned}$$

If conversely $\mathsf{P}^{\mathbf{DF}}(W=1) < \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1)$, we get

$$\begin{aligned}
\Delta^{\mathbf{D}}(\mathbf{F},\mathbf{F}|\mathcal{A}) &= \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1) - \mathsf{P}^{\mathbf{DF}}(W=1) \\
&= \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1) - \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1)\cdot\mathsf{P}^{\mathbf{DF}}(\mathcal{A}) \\
&\qquad - \mathsf{P}^{\mathbf{DF}|\bar{\mathcal{A}}}(W=1)\cdot\mathsf{P}^{\mathbf{DF}}(\bar{\mathcal{A}}) \\
&\leq \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1) - \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1)\cdot\mathsf{P}^{\mathbf{DF}}(\mathcal{A}) - 0 \\
&= \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1) - \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1)\cdot(1-\mathsf{P}^{\mathbf{DF}}(\bar{\mathcal{A}})) \\
&= \mathsf{P}^{\mathbf{DF}|\mathcal{A}}(W=1)\cdot\mathsf{P}^{\mathbf{DF}}(\bar{\mathcal{A}}) \\
&\leq \mathsf{P}^{\mathbf{DF}}(\bar{\mathcal{A}}).
\end{aligned}$$

We conclude that in either case, $\Delta^{\mathbf{D}}(\mathbf{F},\mathbf{F}|\mathcal{A}) \leq \mathsf{P}^{\mathbf{DF}}(\bar{\mathcal{A}})$. □

2.2.4 Computational Security

Modeling the adversary to have unbounded computing power, one gets security statements in an information-theoretical sense. Information-theoretical security comes, however, often at a high price of inefficiency. For an encryption scheme to be perfectly secure, at least one fresh bit of a random key must be used for every bit of a message to be encrypted [34]. In what followings, the adversary's computing power is thus bounded and security notions are stated in a computational sense only.

In order to argue about computationally bounded systems and adversaries, an asymptotic complexity model has to be considered. In such a model, one defines the notions *efficient*, *feasible* and *negligible*. In this model, only bounded adversaries that meet a certain notion of *feasibility* are considered. *Efficiency* is required for resources and converters in order to assure the possibility of an efficient implementation. In this model, security is defined as follows: If a feasible adversary can break the security properties of an efficient system only with *negligible* probability, such an unlikely event (e.g. the event of correctly guessing the key) is "ignored" and the system is considered secure.

Following the hierarchy of levels of abstraction in [28], the level of discrete systems is not the right level to argue about complexity of systems (e.g. in terms of computation steps) which is why it is thus hard to define reasonable asymptotic notions. Nevertheless, in order to be able to formulate asymptotic statements on this level, a reasonable computational model is assumed, allowing to express the upper bound of computation steps (in total) of a system by a number t[1]. As a simplification, trivial operations, such as

[1] A computational model can for example be formulated using interactive Turing machines

forwarding a message or checking for a simple condition (such as if a flag is set), are assumed not to count as a computation step.

The complexity of a system is defined by the number of computation steps t and the number of queries q done by the system. Based on the complexity and the two abstract notions *efficient*[2] and *negligible*, abstract classes of *efficient systems* are introduced. A system (resource, converter or adversary) is in the class of efficient systems (resources, converters or adversaries) if its computation steps and its queries can be bounded by an efficient quantity. More precisely, the considered classes are the *class of efficient resources* $\mathcal{R}^e$, the *class of efficient converters* Σ^e and the class of efficient adversaries $\mathcal{A}^e$ (e.g. distinguishers).

The terms efficient and negligible must be instantiated in such a way that the resulting efficiency classes and negligible quantities have the following closure properties. For any efficient converters $\mathbf{C}_1, \mathbf{C}_2 \in \Sigma^e$, any efficient resources $\mathbf{R}, \mathbf{S} \in \mathcal{R}^e$, any efficient adversary $\mathbf{A} \in \mathcal{A}^e$ and any interface $i \in \mathcal{I}$:

(i) $\mathbf{R} \| \mathbf{S} \in \mathcal{R}^e$

(ii) $\mathbf{C}_1^i \mathbf{S} \in \mathcal{R}^e$

(iii) $\mathbf{A}\mathbf{C}_1^i \in \mathcal{A}^e$

(iv) $\mathbf{C}_1 \circ \mathbf{C}_2 \in \Sigma^e$,

and for any efficient quantity l and any negligible quantities ε_1 and ε_2,

$$k \cdot \varepsilon_1 \text{ and } \varepsilon_1 + \varepsilon_2 \text{ are negligible quantities.}$$

The notions efficient and negligible are usually defined with respect to a *security parameter* k.[3] The security parameter represents security relevant quantities in a system (e.g. the length of a key used in a cryptographic protocol). Thus, families of systems parametrized by the security parameter k, i.e. $\{\mathbf{S}(k)\}_{k \in \mathbb{N}}$, are considered, meaning that the class of efficient systems is a set of families of concrete systems parametrized by the security parameter.

In addition to the asymptotic view, it is interesting to observe certain system parameters for fixed values of the security parameter k (e.g. a fixed key-length of 1024 bits), and thus most of the statements are formulated in two ways:

Concrete statement The *concrete statement* assumes a fixed k and states that if a security predicate (e.g. the advantage in a game is bounded by ε) holds

[2] As simplification, the notions 'efficient' and 'feasible' are hereby assumed to be identical.

[3] Efficient in k can for example be "polynomial in k".

for any k-adversary with certain complexity bounds, then a certain other security predicate holds as well for all k-adversaries with some (potentially slightly different) complexity bounds. The set of k-adversaries satisfying the complexity bounds in a reduction is denoted by $\mathcal{A}_{q_1,q_2,\ldots,q_n,t}$ where the example refers to the class of all adversaries bounded by q_1 queries of the first type, q_2 queries of the second type, ..., q_n queries of the n-th type and bounded by t computation steps. These classes do not satisfy the closure properties.

Asymptotic statement The *asymptotic statement* shows that if a security predicate Q_1 holds for all families of efficient adversaries, then a security predicate Q_2 also holds for all these families.

In a normal reduction, the complexity bounds and the security predicate of the concrete statement are often weakened by small factors during the reduction process. Therefore, a "small amount of security" is intuitively "lost" in every reduction step. As this amount is usually only an efficient quantity, the asymptotic statement follows directly from the concrete statement because *efficient* is closed under composition.

2.2.5 The Hybrid Argument

This section recalls a popular proof technique in cryptography, called the hybrid argument. The argument provides a useful theorem on the (abstract) level of discrete systems that can be applied to shorten proofs and to make them more elegant. The theorem is taken from the lecture notes on cryptography by Maurer [27].

The hybrid argument is often used in reduction proofs when a sequence $\mathbf{Q}_0, \ldots, \mathbf{Q}_n$ of systems is considered where two subsequent systems $\mathbf{Q}_i$ and $\mathbf{Q}_{i+1}$ are assumed to be indistinguishable, and one is interested in the distinguishability of the two "extreme" systems $\mathbf{Q}_0$ and $\mathbf{Q}_n$. Applying the triangle inequality, for any distinguisher $\mathbf{D}$ one gets,

$$\Delta^{\mathbf{D}}(\mathbf{Q}_0, \mathbf{Q}_n) \leq \sum_{i=1}^{n} \Delta^{\mathbf{D}}(\mathbf{Q}_i, \mathbf{Q}_{i-1}).$$

If one assumes that $\mathbf{D}$ is able to distinguish $\mathbf{Q}_0$ and $\mathbf{Q}_n$, this contradicts at least one of our indistinguishability assumptions about a hybrid pair $\mathbf{Q}_i$ and $\mathbf{Q}_{i+1}$, but this contradiction cannot be allocated to a specific pair. This results in an unsatisfying non-constructive statement. Under a special condition, one can nevertheless obtain a constructive reduction from that triangle inequality, namely when only a single pair of systems is "embedded in" every pair of the sequence.

Theorem 2.15. *Let* $\mathbf{S}$ *and* $\mathbf{T}$ *be (single-interface) systems and let* $\mathbf{C}_1, \ldots, \mathbf{C}_n$ *be converters such that*

$$\mathbf{C}_{i+1}\mathbf{S} \equiv \mathbf{C}_i\mathbf{T}$$

for $i = 1, \ldots, n-1$*, and define* $\mathbf{F} := \mathbf{C}_1\mathbf{S}$ *and* $\mathbf{G} := \mathbf{C}_n\mathbf{T}$*. Then, for any distinguisher* $\mathbf{D}$*,*

$$\Delta^{\mathbf{D}}(\mathbf{F}, \mathbf{G}) = n \cdot \Delta^{\mathbf{DC_I}}(\mathbf{S}, \mathbf{T}),$$

where I *is chosen uniformly from* $\{1, \ldots, n\}$*.*

The proof of the theorem can be found in [27, Ch. 5.4].

2.3 Notation

Notation that is consequently used in this thesis includes $x \in_R S$ to denote the sampling of x uniformly at random from the set S. Assignment of variables is denoted by $x \leftarrow y$, x^n stands for the tuple $(x_1, \ldots, x_n)$, the appending of an element to a tuple is denoted by $T \| x$, and the i-th member of the tuple T by $T(i)$. The disjoint union of two (potentially non-disjoint) sets S_1 and S_2 is denoted by $S_1 \dot{\cup} S_2$. The relations $\in$ and $\notin$ are both used for set membership and for being part of a tuple.

Specifying Systems In this thesis, discrete systems are characterized using a pseudo-code style that is introduced in this section. As a first part of the characterization, the input and output sets $\mathcal{X}$ and $\mathcal{Y}$ are specified. Usually, there are different types of inputs and outputs. To distinguish between the different types, the convention that the input starts with a string identifier of the type followed by the actual input, as for example $[''\text{type } 1'', x]$, is used.

After the domain and range specification, the pseudo-code is given according to the following sample:

initialize
// some initialization code

on every **input** X_i **do**
// do something on every input

on input $X_i = [\text{'type 1'}, \ldots]$ **do**
// do something on an input of type 1
// output something on this input
$\vdots$

on first input $X_i = [\text{'type j'}, \ldots]$ **do**
// do something on an input of type j
// output something on this input

The initialization part is executed before the first round of communication of the system. In each round i, all the event conditions are checked top-down and their respective body is executed immediately if the condition is true. Multiple bodies can thus be executed in one round, especially the ones without a condition ("on every input X_i") are executed in every round. Bodies of conditions saying "On first input" are only executed the first time a query of the corresponding type is input.

The actual implementation of the pseudo-code is irrelevant in this context. The only two things that matter is that the pseudo-code is a fully valid characterization of a random system and that the implementation of the pseudo-code can be done efficiently (e.g. the implementation of a map using a hash map or the first-time-only condition by a simple flag).

Chapter 3

Secure Communication

Secure communication is an ideal scenario of communication where two honest parties, Alice and Bob, communicate without a (hypothetical) adversary Eve being able to interfere in their communication, be it just as eavesdropper or as an active adversary trying to modify the communication, depending on the type of security one is talking about. In the system model introduced in the previous chapter, such a scenario can be modeled by *communication channels*. A communication channel is a special type of resource with three interfaces, the sender interface A, the receiver interface B and the adversary's interface E, where the type of security is reflected by the abilities of the adversary at his interface.

This chapter is structured as follows: Firstly, the most important types of communication channels are introduced and defined, using the ideas and definitions of [26, 27]: the insecure channel, the authenticated channel and the secure channel. Secondly, the basic primitives of secret-key cryptography, i.e. authentication and encryption schemes, are formalized. In the last part of this chapter, the confidential channel is introduced because the definition needs a model of the corresponding malleability properties. Malleability properties capture the possible adversarial influence on delivered messages corresponding to modifying transmitted ciphertexts. Moreover, on the basis of the introduced confidential channel, a formalization of public-key encryption schemes and their security is given.

3.1 Communication Channels

Communication channels are $\{A, B, E\}$-resources that can be further characterized by the functionality provided to the sender and receiver (single-use vs. multiple-use channel) and by the type of access that is provided at the E-

interface to the adversary Eve. Unless specified otherwise, communication channels are always multiple-use channels and are denoted by arrow-like symbols in the following. The message space used for communication channels is denoted by $\mathcal{M}$.

On the basis of the paradigm of constructive cryptography [26], the definitions of the three most important channels are given. As existing communication channels in practice usually do not give any security guarantees (e.g. communication over the internet), the first channel is called *insecure channel* and allows an adversary full access to the channel.

3.1.1 Insecure Channel

Definition 3.1. *An* insecure channel $\longrightarrow$ *with message space* $\mathcal{M}$ *is a communication channel giving the adversary full access to the channel as follows: If no adversary is present,* $\longrightarrow$ *takes in round* i *input* $m_i \in \mathcal{M}$ *at the A-interface and immediately outputs* m_i *at the B-interface. If an adversary is present,* $\longrightarrow$ *works as follows:*

on input $X_i = m_i \in \mathcal{M}$ at A-interface **do**
  output $Y_i = m_i$ at E-interface
on input $X_i = m_i' \in (\mathcal{M} \dot{\cup} \{\perp\})$ at E-interface **do**
  output $Y_i = m_i'$ at B-interface

The symbol $\perp$ *is not part of the message space and indicates the deletion of the message in such a way that the receiver recognizes the deletion.*

Note that the adversary has three possibilities of modifying the sent message in the case of the insecure channel $\longrightarrow$: He can either insert an arbitrary message (including the one that was initially sent), he can delete the message such that Bob detects the deletion or he can simply input nothing, which is referred to as an undetected deletion of the message.[1]

3.1.2 Authenticated Channel

If the access of the adversary is restricted so that sent messages are still leaked to the adversary but the receiver receives only messages that were initially sent by the sender, the channel is called *authenticated* (i.e. the adversary can only delete and forward messages). This is represented in the channel symbol by a dot at the sender side (representing the "exclusiveness" of sending the message).

[1] Communication is assumed to be completely asynchronous without any time guarantees. Therefore, a receiver cannot detect if he does not receive a sent message.

Definition 3.2. *An* authenticated channel $\bullet\longrightarrow$ *(allowing replaying and reordering) with message space $\mathcal{M}$ is a communication channel leaking the sent messages, but giving the adversary only forward and delete access: If no adversary is present, $\bullet\longrightarrow$ takes in round i input $m_i \in \mathcal{M}$ at the A-interface and immediately outputs m_i at the B-interface. If an adversary is present, $\bullet\longrightarrow$ works as follows:*

initialize
  initialize map $M : \mathbb{N} \to \mathcal{M}$
  initialize $k \leftarrow 1$
on input $X_i = m_i \in \mathcal{M}$ at A-interface **do**
  $M[k] \leftarrow m_i$, $k \leftarrow k + 1$
  output $Y_i = m_i$ at E-interface
on input $X_i = [\text{'forward'}, j]$ at E-interface **do**
  if $j < k$ **then** output $Y_i = M[j]$ at B-interface
on input $X_i = [\text{'delete'}]$ at E-interface **do**
  output $Y_i = \perp$ at B-interface

Again, the adversary additionally has the possibility of deleting a message undetected by not providing any input for a given output at the E-interface.

3.1.3 Secure Channel

The final goal of secure communication is that the access of the adversary is restricted, so that he does not get any information about the sent messages (except for their length), and that he cannot insert any new message or modify any sent message. The channel with such restricted access is called *secure channel* and its symbol gets an additional dot at the receiver side (representing the "exclusiveness" of receiving the message).

Definition 3.3. *A* secure channel $\bullet\longrightarrow\bullet$ *(allowing replaying and reordering) with message space $\mathcal{M}$ is a communication channel leaking no information about sent messages (except for their length) and giving the adversary only forward and delete access: If no adversary is present, $\bullet\longrightarrow\bullet$ takes inputs $X_i = m_i \in \mathcal{M}$ at the A-interface and immediately outputs $Y_i = m_i$ at the B-interface. If an adversary is present, $\bullet\longrightarrow\bullet$ works as follows:*

initialize
  initialize map $M : \mathbb{N} \to \mathcal{M}$
  initialize $k \leftarrow 1$
on input $X_i = m_i \in \mathcal{M}$ at A-interface **do**
  $M[k] \leftarrow m_i$, $k \leftarrow k + 1$
  output $Y_i = |m_i|$ at E-interface

on input $X_i = [\text{'forward'}, j]$ at E-interface **do**
  if $j < k$ **then** output $Y_i = M[j]$ at B-interface
on input $X_i = [\text{'delete'}]$ at E-interface **do**
  output $Y_i = \perp$ at B-interface

Regarding undetected deletion, the statements made for the authenticated channel are also valid for the secure channel.

It has to mentioned that both the authenticated and the secure channel defined here allow the adversary to replay and reorder messages, since forwarding arbitrary messages from the history is possible. One can also define stronger versions of the channels allowing no reordering and replaying by allowing only the forwarding of sent messages in order and only once. The stronger versions are actually the standard variants used, for example, in [27, 29]. In this thesis, the weaker versions are used since the channels are compared with game-based security notions and the definitions of those notions usually do not give any guarantees about the message ordering.

The scenario where no adversary is present is denoted by a dummy converter $\perp$ at the E-interface, e.g. $\perp^E \longrightarrow$, in what follows.

3.2 Secret-key Channel Constructions

First of all, channel constructions are studied in the context of secret-key cryptography where the two honest entities have access to a shared secret key. The shared secret key is modeled as a system that outputs a key at the two honest interfaces A and B and provides *no output* at the E-interface of the adversary.

Definition 3.4. *A* shared secret key $\bullet\!\!-\!\!\bullet$ *with key space* $\mathcal{K}$ *is a* $\{A, B, E\}$*-resource providing a shared uniform random key* $\mathrm{K} \in_R \mathcal{K}$ *at the A- and B-interfaces and no output at the E-interface.*

The shared secret key can, for instance, be the result of a key exchange protocol. The key is then applied in a secret-key protocol. The two fundamental types of protocols are defined: The first protocol is aiming at authenticity of communication and is called an *authentication scheme*. The second one is a protocol aiming at privacy of communication and is called *encryption scheme*. In order to formalize the security of such a protocol, the simulation-based definition of security from the previous chapter is used (see Definition 2.3).

According to this definition, the goal of a protocol $\pi = (\pi_1, \pi_2)$ is to construct a more secure channel from a less secure one. For an *authentication*

scheme, traditional security notions (e.g. strong unforgeability against chosen message attacks) correspond to a channel construction resulting in an authenticated channel. An authentication scheme is thus called *secure* if it constructs an authenticated channel $\bullet\!\!\longrightarrow$ from an insecure channel $\longrightarrow$ and a shared secret key $\bullet\!\!=\!\!\bullet$. Accordingly, for encryption schemes, traditional notions (e.g. indistinguishability under chosen-plaintext attacks) translate into a channel transformation constructing a secure channel from an authenticated channel. An *encryption scheme* is correspondingly considered *secure* if it constructs a secure channel $\bullet\!\!\longrightarrow\!\!\bullet$ from an authenticated channel $\bullet\!\!\longrightarrow$ and a shared secret key $\bullet\!\!=\!\!\bullet$. This traditional order of the transformations to get a secure channel from an insecure channel is known as the Encrypt-then-Authenticate paradigm (EtA).

3.2.1 Authentication Schemes

Authentication schemes aim at preventing an adversary from modifying a message undetected. This is usually done by adding a message authentication code (MAC), using a secret key, and letting the receiver check the MAC, using the same secret key.

Definition 3.5. *An* authentication scheme *is a protocol $\varphi = (\varphi_1, \varphi_2)$ with key space $\mathcal{K}$, input message space $\mathcal{M}$ and output message space $\mathcal{M}'$. φ_1 and φ_2 connect with their inner interfaces to a shared secret key $\bullet\!\!=\!\!\bullet$ with key space $\mathcal{K}$ and to a channel with message space $\mathcal{M}'' \supseteq \mathcal{M}'$. The resulting resource is a channel with message space $\mathcal{M}$.*

An authentication scheme family, *denoted shortly by $\varphi(k)$, is a family of authentication schemes parametrized by the security parameter k, $\{\varphi(k)\}_{k\in\mathbb{N}}$.*

Security of an authentication scheme is defined as securely constructing an authenticated channel from an insecure channel and a shared secret key. This definition corresponds to the traditional game-based notion of weak unforgeability against chosen message attacks (WUF-CMA).

Definition 3.6. *An authentication scheme $\varphi = (\varphi_1, \varphi_2)$ is* secure with error ε *and with respect to the distinguisher class $\mathcal{D}$ and converter set Σ if it securely constructs an authenticated channel $\bullet\!\!\longrightarrow$ from $(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ with error ε:*

$$\exists \sigma \in \Sigma : \Delta^{\mathcal{D}}(\varphi_1^A \varphi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \sigma^E(\bullet\!\!\longrightarrow)) \leq \varepsilon \qquad \text{(security)}$$

and

$$\Delta^{\mathcal{D}}(\varphi_1^A \varphi_2^B \perp^E(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \perp^E(\bullet\!\!\longrightarrow)) \leq \varepsilon \qquad \text{(availability)}$$

An authentication scheme family $\varphi(k)$ is secure if for every k $\varphi(k)$ is secure with error ε and with respect to the class of efficient adversaries $\mathcal{A}^e$ and the set of efficient converters Σ^e, and the error ε is negligible in k.

3.2.2 Encryption Schemes

In a similar way, an encryption scheme can be seen as an invertible mapping of messages (so called plaintexts) onto ciphertexts. It can be defined as a protocol transforming a channel with message space containing the set of ciphertexts into a channel with a message space consisting of the set of plaintexts.

Definition 3.7. *A* secret-key encryption scheme *is a protocol $\pi = (\pi_1, \pi_2)$ with key space $\mathcal{K}$, message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$. π_1 and π_2 connect with their inner interfaces to a shared secret key $\bullet\!\!=\!\!\bullet$ with key space $\mathcal{K}$ and to a channel with message space $\mathcal{M}' \supseteq \mathcal{C}$. The resulting resource is a channel with message space $\mathcal{M}$.*

A secret-key encryption scheme family*, denoted shortly by $\pi(k)$, is a family of secret-key encryption schemes parametrized by the security parameter k, $\{\pi(k)\}_{k\in\mathbb{N}}$.*

Security of an encryption scheme is defined as constructing a secure channel from an authenticated channel and a shared secret key. The notion of indistinguishability against chosen-plaintext attacks (IND-CPA) is the corresponding security notion in the traditional game-based model.

Definition 3.8. *A secret-key encryption scheme $\pi = (\pi_1, \pi_2)$ is* secure with error ε *and with respect to the distinguisher class $\mathcal{D}$ and converter set Σ if it securely constructs a secure channel $\bullet\!\!\longrightarrow\!\!\bullet$ from $(\bullet\!\!\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ with error ε:*

$$\exists \sigma \in \Sigma : \Delta^{\mathcal{D}}(\pi_1^A \pi_2^B(\bullet\!\!\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \sigma^E(\bullet\!\!\longrightarrow\!\!\bullet)) \leq \varepsilon \qquad \text{(security)}$$

and

$$\Delta^{\mathcal{D}}(\pi_1^A \pi_2^B \perp^E(\bullet\!\!\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \perp^E(\bullet\!\!\longrightarrow\!\!\bullet)) \leq \varepsilon \qquad \text{(availability)}$$

A secret-key encryption scheme family $\pi(k)$ is secure if for every k $\pi(k)$ is secure with error ε and with respect to the class of efficient distinguishers $\mathcal{D}^e$ and the set of efficient converters Σ^e, and the error ε is negligible in k.

3.3 Confidentiality and Malleability

There are scenarios where the above channel constructions cannot be applied and one wants to make a statement about a construction of a channel with confidentiality guarantees only. The *confidential channel* is important, for example, in the context of *public-key cryptography*, or when applying the *Authenticate-then-Encrypt (AtE)* paradigm.

Defining the confidential channel is, however, not as simple as specifying the other channels. The "passive abilities" of the adversary in a confidential

channel are clearly that the adversary does not learn anything about the sent messages (except for their length). Modeling the "active abilities" is, in contrast, not straightforward. A first idea could be to let the adversary either forward or delete messages or to let him choose the message that is output at the receiver's interface. This approach does, however, not cover cases where the adversary is able to systematically modify sent messages not known to him (e.g. flipping a bit of a sent message). Intuitively, this calls for an additional ability of the adversary, that is, to specify a function that is applied to the sent message and whose result is output at the receiver's interface. A formalization of that concept, i.e. the concept of the *malleability* of a channel, is proposed by Maurer and Tackmann [29, Section 4.2].

3.3.1 Formalizing the Malleability

Since the adversary "sees" all the ciphertexts corresponding to messages sent by Alice and all ciphertexts corresponding to messages received by Bob, he can intuitively use all these ciphertexts to assemble a new ciphertext. The decryption of this constructed ciphertext thus potentially depends on all previous messages from the history. An adversary can also "invent" parts of the ciphertext, potentially resulting in a probabilistic behavior of decryption (e.g. when decryption involves a pseudo-random permutation on a "new" input).

The influence of the adversary on each sent message can thus be described as a (probabilistic) transformation that is applied to the complete message history, and the result of the transformation is delivered at the B-interface. Formally, the input of the adversary corresponds to a (probabilistic) transformation $F : \mathcal{M}^* \times \mathcal{M}^* \to \mathcal{M}$ where the parameters correspond to the previous messages at the A- respectively B-interface.

As the malleability and therefore also the set of possible transformations can change in each round, a *set of eligible transformations* is defined out of which the adversary can choose the transformation to be applied. A channel is intuitively more secure if these sets of eligible transformations are small.

Definition 3.9. *The* malleability of a communication channel *is a tuple* $\mathcal{F} := (\{F_\alpha\}_{\alpha\in\mathcal{A}}, \{\mathcal{A}_q\}_{q\in\mathbb{N}})$*, where* $\{F_\alpha\}_{\alpha\in\mathcal{A}}$ *is a family of transformations* $F_\alpha : \mathcal{M}^* \times \mathcal{M}^* \to \mathcal{M}$*. The set* $\mathcal{A}$ *specifies the set of possible transformations, whereas after* q *queries the random variable* $\mathcal{A}_q \subseteq \mathcal{A}$ *describes the* eligible *transformations.*

The formalization of the concept of the malleability as tuple $\mathcal{F}$ allows to define the confidential channel. In addition to the forwarding and deleting abilities already defined for other channels, a confidential channel, allowing reordering and replaying, must allow modifying messages based on the

malleability $\mathcal{F}$ and replaying messages that were the result of such a modification.

Definition 3.10. *A* confidential channel $\longrightarrow\!\!\bullet$ *(allowing replaying and reordering) with message space $\mathcal{M}$ is a communication channel leaking no information about sent messages (except for their length) and allowing an arbitrary malleability described by $\mathcal{F}$ where every transformation in the family $\{F_\alpha\}_{\alpha\in\mathcal{A}}$ is efficiently implementable. If no adversary is present, $\longrightarrow\!\!\bullet$ takes inputs $X_i = m_i \in \mathcal{M}$ at the A-interface and immediately outputs $Y_i = m_i$ at the B-interface. If an adversary is present, $\longrightarrow\!\!\bullet$ works as follows:*

initialize
  initialize $i_A \leftarrow 1$, $i_E \leftarrow 1$
  initialize tuples M_1, M_2
on input $X_i = m_i \in \mathcal{M}$ at A-interface **do**
  determine $\mathcal{A}_i$ according to the specification of $\mathcal{F}$
  $i_A \leftarrow i_A + 1$
  $M_1 \leftarrow M_1 \| m_i$
  output $Y_i = (|m_i|, \mathcal{A}_i)$ at E-interface
on input $X_i = [\text{'forward'}, j]$ at E-interface **do**
  if $j < i_A$ **then** output $Y_i = M_1(j)$ at B-interface
on input $X_i = [\text{'delete'}]$ at E-interface **do**
  output $Y_i = \perp$ at B-interface
on input $X_i = [\text{'modify'}, \alpha_i], \alpha_i \in \mathcal{A}_{i-1}$ at E-interface **do**
  evaluate $m_i' \leftarrow F_{\alpha_i}(M_1, M_2)$
  $i_E \leftarrow i_E + 1$
  $M_2 \leftarrow M_2 \| m_i'$
  output $Y_i = m_i'$ at B-interface
on input $X_i = [\text{'replay'}, j]$ at E-interface **do**
  if $j < i_E$ **then** output $Y_i = M_2(j)$ at B-interface

The distribution of each $\mathcal{A}_i$ depends on the lengths $|m_1|, \ldots, |m_{i-1}|$ of the messages input at the A-interface and the previous $\mathcal{A}_1, \ldots, \mathcal{A}_{i-1}$ and the previous input transformations α_i.

The set of eligible transformations can change with any query (i.e. $\mathcal{A}_{i-1} \neq \mathcal{A}_i$), but the set only has to be specified explicitly on an input at the A-interface and a corresponding output at the E-interface. On input $X_i = [\text{'modify'}, \alpha_i]$ at the E-interface, the new set of eligible transformations $\mathcal{A}_i$ is fully specified by α_i and $\mathcal{A}_{i-1}$, both known to the adversary, and does not explicitly have to be output.

3.3.2 Soundness of Authenticate-then-Encrypt

Using the paradigm of Authenticate-then-Encrypt (AtE), one wants to establish a secure channel by first authenticating the message and then encrypting the pair consisting of a message and an authentication code. In terms of channel constructions, an encryption scheme is, in a first step, used to construct a confidential channel $\longrightarrow\!\!\bullet$ from an insecure channel $\longrightarrow$ and a shared secret key $\bullet\!\!=\!\!\bullet$. In a second step, an authentication scheme is applied with the goal of constructing a secure channel $\bullet\!\!\longrightarrow\!\!\bullet$ from a confidential channel $\longrightarrow\!\!\bullet$ and a shared secret key $\bullet\!\!=\!\!\bullet$.

The composition of encryption and authentication in this order does, however, not generally construct a secure channel, as pointed out in [24, 7]. Phrased constructively, the problem is that the arbitrary malleability offered by $\longrightarrow\!\!\bullet$ potentially allows an adversary to systematically abuse the verification of authenticity to his advantage and to break confidentiality (e.g. when flipping a bit in a ciphertext results in a failed MAC-verification for only some specific messages, but not for others). Note that traditional security notions do not address this problem. To avoid such undesired behavior, the malleability has to be restricted, which is a very complex task in a game-based model since it has never been formalized. By using the above formalization of malleability, the task becomes much simpler. A sufficient restriction to the malleability rendering an AtE composition secure has been given in [29].

In contrast to the AtE composition paradigm, the counterpart paradigm Encrypt-then-Authenticate (EtA) is sound using even traditional game-based security notions and should be preferred [24]. In the context of constructive cryptography, the soundness of EtA can be seen as a simple example application of the Composability Theorem 2.4: Using an encryption and authentication scheme according to Definitions 3.6 and 3.8, EtA is sound if the considered class of adversaries is closed under composition with the class of converters (protocols and simulators).

3.3.3 Public-key Encryption

A public-key encryption scheme is a protocol that produces a publicly known value, the *public key*, and a value that is only known to the receiver party Bob. This value is called the *private key*. The public key can then be used by the sender Alice to transform a message into a ciphertext which Bob can later decrypt to the original message using the private key.

Definition 3.11. *A* public-key encryption scheme *is a protocol $\pi = (\pi_1, \pi_2)$ with public-key space $\mathcal{K}$, message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$. π_1 and π_2 connect with their inner interfaces to a (potentially single-use) authenticated channel*

$\longleftarrow\!\!\bullet$ *from B to A with message space $\mathcal{K}$ and to an insecure channel $\longrightarrow$ from A to B with message space $\mathcal{M}' \supseteq \mathcal{C}$. The resulting resource is a channel from A to B with message space $\mathcal{M}$.*

A public-key encryption scheme family*, denoted shortly by $\pi(k)$, is a family of public-key encryption schemes parametrized by the security parameter k, $\{\pi(k)\}_{k\in\mathbb{N}}$.*

In terms of a channel transformation, a public-key encryption scheme is considered secure if it constructs a confidential channel from Alice to Bob using an insecure channel from Alice to Bob and a (potentially single-use) authenticated channel from Bob to Alice. The authenticated channel from receiver to sender is used to distribute the public key in a authenticated manner. This channel can thus be single-use or even limited to be only available for a short time.

Definition 3.12. *A public-key encryption scheme $\pi = (\pi_1, \pi_2)$ is* secure with error ε *and with respect to the distinguisher class $\mathcal{D}$ and converter set Σ if it securely constructs a confidential channel $\longrightarrow\!\!\bullet$ from A to B according to Definition 3.10 from $(\longleftarrow\!\!\bullet \parallel \longrightarrow)$ with error ε:*

$$\exists \sigma \in \Sigma : \Delta^{\mathcal{D}}(\pi_1^A \pi_2^B(\longrightarrow \parallel \longleftarrow\!\!\bullet), \sigma^E(\longrightarrow\!\!\bullet)) \leq \varepsilon \qquad \text{(security)}$$

and

$$\Delta^{\mathcal{D}}(\pi_1^A \pi_2^B \perp^E(\longrightarrow \parallel \longleftarrow\!\!\bullet), \perp^E(\longrightarrow\!\!\bullet)) \leq \varepsilon \qquad \text{(availability)}$$

A public-key encryption scheme family $\pi(k)$ is secure if for every k $\pi(k)$ is secure with error ε and with respect to the class of efficient distinguishers $\mathcal{D}^e$ and the set of efficient converters Σ^e, and the error ε is negligible in k.

Chapter 4

Game-based Security Notions

This chapter deals with game-based security notions and analyzes their formalization as a type of malleability in the model of constructive cryptography. For this, several security notions that are related to non-malleability, or that have interesting malleability characteristics, are formalized in a unified way. This chapter is organized as follows: Section 4.1 formalizes the notion of a game, the game winning probability and the advantage of an adversary in a game. The notion of a *bit-guessing game* is introduced since most of the existing games are stated in a indistinguishability style and thus in this form. Then, the winning of bit-guessing games is related to the concept of distinguishing.

Section 4.2 discusses the appropriate attack model that should be considered in the channel-based model according to constructive cryptography.

The further sections of this chapter focus the main results. The restricted types of malleability induced by the following game-based notions are stated and proven: Non-malleability and indistinguishability are studied in Section 4.3, integrity of plaintexts and ciphertexts, as well as existential unforgeability, in Section 4.4. Section 4.6 deals with the two notions of plaintext-uncertainty and chosen-plaintext forgery.

4.1 Games as Systems

In order to formalize the notion of a *game* with one player (e.g. the adversary), one should recall the definition of a $(\mathcal{X}, \mathcal{Y})$-system from Chapter 2. Such a system proceeds in rounds taking—in round i—input X_i and providing the output Y_i. A game is a $(\mathcal{X}, \mathcal{Y})$-system with an additional output in every round consisting of a bit indicating whether the game has been won. An important property of the game winning condition is its *monotonicity*.

Definition 4.1. *For a $(\mathcal{X}, \mathcal{Y} \times \{0,1\})$-system $\mathbf{S}$, the binary component A_i of the output (Y_i, A_i) is called a* monotone binary output (MBO) *if $A_i = 1$ implies $A_j = 1$ for $j \geq i$. Such a system $\mathbf{S}$ with MBO is called a* game.

Connecting a system (e.g. an adversary) to the game, the composition defines a random experiment. The *winning probability* is defined by the probability of the MBO bit to be 1 in this experiment.

Definition 4.2. *For a game $\mathbf{S}$ and for a system $\mathbf{W}$, we denote with $\Gamma_q^{\mathbf{W}}(\mathbf{S})$, the* game winning probability*, the probability that $\mathbf{W}$ wins the game within q queries:*

$$\Gamma_q^{\mathbf{W}}(\mathbf{S}) := \mathsf{P}^{\mathbf{WS}}(A_q = 1),$$

and with $\Gamma^{\mathbf{W}}(\mathbf{S})$ the probability that $\mathbf{W}$ wins the game eventually:

$$\Gamma^{\mathbf{W}}(\mathbf{S}) := \sup_{i \geq 1} \mathsf{P}^{\mathbf{WS}}(A_i = 1).$$

For a class $\mathcal{W}$ of systems, the winning probability of the best $\mathbf{W}$ in $\mathcal{W}$ is denoted as

$$\Gamma^{\mathcal{W}}(\mathbf{S}) := \sup_{\mathbf{W} \in \mathcal{W}} \Gamma^{\mathbf{W}}(\mathbf{S}).$$

4.1.1 Bit-guessing Games

In the following, only a special type of game is considered: The goal of the game can be seen as guessing a secret bit b correctly. The bit is assumed to be fixed at initialization and to remain constant throughout the execution of the game. Guessing the bit is represented by the first binary input with the identifier 'guess' causing the MBO to switch to 1 if the binary input matches the bit b. Moreover, no other input, i.e. also no second input of this form, has any influence on the MBO. Since the game is defined as a *random system*, no statement about an inner state can be made, which is why a formalization has to be stated differently. In other words, a game is a bit-guessing game if it is equivalent to a random combination of two other games where the secret bit is not random but fixed.

Definition 4.3. *Let $\mathbf{S}_b$ denote a game that is always won by any system that inputs the fixed bit b as its guess and that is never won by any system that inputs the fixed bit $1 - b$ as its guess, and in which any input other than a guess or a second guess cannot turn the MBO to 1. Formally, the MBO is initialized to 0, $A_0 := 0$. Let G_i^b denote the event that i-th input is the first guess input and of the form $X_i = [\text{'guess'}, b]$. Let further N_i denote the event that input i is not of the form $X_i = [\text{'guess'}, b']$, $b' \in \{0,1\}$ or there exists $j < i$ such that input j was of the form $X_i = [\text{'guess'}, b']$, $b' \in \{0,1\}$. We thus require that*

$$\mathsf{p}^{\mathbf{S}_b}_{A_i = 1 | X^i Y^i G_i^b} = 1, \ \mathsf{p}^{\mathbf{S}_b}_{A_i = 1 | X^i Y^i G_i^{1-b}} = 0,$$

and

$$\mathsf{p}^{\mathbf{S}_b}_{A_i=b|X^iY^iN_i} = \mathsf{p}^{\mathbf{S}_b}_{A_{i-1}=b|X^{i-1}Y^{i-1}}.$$

A game $\mathbf{S}$ *is called a* bit-guessing game *if there exist games* $\mathbf{S}_0$ *and* $\mathbf{S}_1$ *such that* $\mathbf{S} \equiv \mathbf{S}_B$ *where* B *is a binary random variable chosen at random according to some distribution. The games* $\mathbf{S}_0$ *and* $\mathbf{S}_1$ *are called* conditional games of $\mathbf{S}$.

A game $\mathbf{S}$ *is called a* uniform bit-guessing game *if* $\mathbf{S}$ *is a bit-guessing game where the random variable* B *is chosen uniformly at random (i.e.* 0 *and* 1 *each with probability* $\frac{1}{2}$*).*

Note that any system that always inputs the fixed bit b as its guess wins a uniform bit-guessing game with probability $\frac{1}{2}$.

Sometimes, it is easier to analyze the winning probability of a system in a bit-guessing game with respect to the two conditional games of the game. The following Lemma states the correlation of the respective winning probabilities.

Lemma 4.4. *Let* $\mathbf{S}$ *be a uniform bit-guessing game and* $\mathbf{S}_0$ *and* $\mathbf{S}_1$ *its two conditional games defined above, then for any system* $\mathbf{W}$

$$\Gamma^{\mathbf{W}}(\mathbf{S}) = \frac{1}{2} \cdot \Gamma^{\mathbf{W}}(\mathbf{S}_0) + \frac{1}{2} \cdot \Gamma^{\mathbf{W}}(\mathbf{S}_1)$$

Proof. We simply use Definitions 4.2 and 4.3 and get

$$\begin{aligned}
\Gamma^{\mathbf{W}}(\mathbf{S}) &= \sup_{i\geq 1} \mathsf{P}^{\mathbf{WS}}(A_i = 1) \\
&= \sup_{i\geq 1} \mathsf{P}^{\mathbf{WS}_B}(A_i = 1), B \in_R \{0,1\} \\
&= \frac{1}{2} \cdot \sup_{i\geq 1} \mathsf{P}^{\mathbf{WS}_0}(A_i = 1) + \frac{1}{2} \cdot \sup_{i\geq 1} \mathsf{P}^{\mathbf{WS}_1}(A_i = 1) \\
&= \frac{1}{2} \cdot \Gamma^{\mathbf{W}}(\mathbf{S}_0) + \frac{1}{2} \cdot \Gamma^{\mathbf{W}}(\mathbf{S}_1).
\end{aligned}$$

□

As a uniform bit-guessing game can be won by an adversary always outputting a fixed bit with probability $\frac{1}{2}$, the game winning probability itself seems not a good measure for the abilities of an adversary. Similar in spirit of the distinguishing advantage, a game-winning *advantage* is defined.

Definition 4.5. *For a uniform bit-guessing game* $\mathbf{S}$*, the* advantage *of a system* $\mathbf{W}$ *in winning the bit-guessing game* $\mathbf{S}$ *is defined as*

$$\Phi^{\mathbf{W}}(\mathbf{S}) := 2 \left| \Gamma^{\mathbf{W}}(\mathbf{S}) - \frac{1}{2} \right|.$$

Using Lemma 4.4, the expression can be rewritten as:

Corollary 4.6. *Let* $\mathbf{S}$ *be a uniform bit-guessing game and* $\mathbf{S}_0$ *and* $\mathbf{S}_1$ *its two conditional games defined above, then for any system* $\mathbf{W}$

$$\Phi^{\mathbf{W}}(\mathbf{S}) = \left| \Gamma^{\mathbf{W}}(\mathbf{S}_0) + \Gamma^{\mathbf{W}}(\mathbf{S}_1) - 1 \right|$$

Similarly to the statement in Lemma 2.14, the game winning probability of two conditionally equivalent games can be related as follows.

Lemma 4.7. *Let* $\mathbf{S}$ *be a uniform bit-guessing game on which additionally a MES* $\mathcal{A}$ *is defined and let* $\mathbf{T}$ *be a uniform bit-guessing game such that* $\mathbf{S}|\mathcal{A} \equiv \mathbf{T}$*, then for any system* $\mathbf{W}$*,*

$$\left|\Gamma^{\mathbf{W}}(\mathbf{S}) - \Gamma^{\mathbf{W}}(\mathbf{T})\right| \leq \mathsf{P}^{\mathbf{DS}}(\bar{\mathcal{A}}).$$

Proof. The absolute difference means that either $\Gamma^{\mathbf{W}}(\mathbf{S}) \leq \Gamma^{\mathbf{W}}(\mathbf{T}) + \mathsf{P}^{\mathbf{DS}}(\bar{\mathcal{A}})$ or that $\Gamma^{\mathbf{W}}(\mathbf{S}) \geq \Gamma^{\mathbf{W}}(\mathbf{T}) - \mathsf{P}^{\mathbf{DS}}(\bar{\mathcal{A}})$. We note that either $\mathcal{A}$ or $\bar{\mathcal{A}}$ in any random experiment. Thus,

$$\Gamma^{\mathbf{W}}(\mathbf{S}) = \Gamma^{\mathbf{W}}(\mathbf{S}|\mathcal{A}) \cdot \mathsf{P}^{\mathbf{DS}}(\mathcal{A}) + \Gamma^{\mathbf{W}}(\mathbf{S}|\bar{\mathcal{A}}) \cdot \mathsf{P}^{\mathbf{DS}}(\bar{\mathcal{A}}).$$

To show the first inequality, we give an upper bound of the probability and use the conditional equivalence

$$\begin{aligned}\Gamma^{\mathbf{W}}(\mathbf{S}) &\leq \Gamma^{\mathbf{W}}(\mathbf{S}|\mathcal{A}) \cdot 1 + 1 \cdot \mathsf{P}^{\mathbf{DS}}(\bar{\mathcal{A}}) \\ &= \Gamma^{\mathbf{W}}(\mathbf{T}) + \mathsf{P}^{\mathbf{DS}}(\bar{\mathcal{A}}).\end{aligned}$$

To show the second inequality, we bound the probability accordingly to get

$$\begin{aligned}\Gamma^{\mathbf{W}}(\mathbf{S}) &= \Gamma^{\mathbf{W}}(\mathbf{S}|\mathcal{A}) \cdot (1 - \mathsf{P}^{\mathbf{DS}}(\bar{\mathcal{A}})) + \Gamma^{\mathbf{W}}(\mathbf{S}|\bar{\mathcal{A}}) \cdot \mathsf{P}^{\mathbf{DS}}(\bar{\mathcal{A}}) \\ &\geq \Gamma^{\mathbf{W}}(\mathbf{S}|\mathcal{A}) - \Gamma^{\mathbf{W}}(\mathbf{S}|\mathcal{A}) \cdot \mathsf{P}^{\mathbf{DS}}(\bar{\mathcal{A}}) + 0 \\ &\geq \Gamma^{\mathbf{W}}(\mathbf{S}|\mathcal{A}) - 1 \cdot \mathsf{P}^{\mathbf{DS}}(\bar{\mathcal{A}}) + 0 \\ &= \Gamma^{\mathbf{W}}(\mathbf{T}) - \mathsf{P}^{\mathbf{DS}}(\bar{\mathcal{A}}).\end{aligned}$$

□

4.1.2 Game-based Security Notions

In the context of game-based security notions, one usually defines a security notion for an encryption scheme or other cryptographic protocols over a bit-guessing game. We state here for both secret-key and public-key encryption schemes $\pi = (\pi_1, \pi_2)$ according to Definitions 3.7 and 3.11 what it means to be secure with respect to some security notion that is defined over a particular game as follows:

For a secret-key or public-key encryption scheme $\pi = (\pi_1, \pi_2)$ according to Definitions 3.7 and 3.11 and a uniform bit-guessing game $\mathbf{S}(\pi)$ according to Definition 4.3 linked to the security notion *NOTION* and having queries of types $\{1, 2, \ldots, n\}$, we call the encryption scheme π $(\varepsilon, q_1, q_2, \ldots, q_n, t)$-*secure in the sense of NOTION*, if

$$\Phi^{\mathcal{A}_{(q_1,q_2,\ldots,q_n,t)}}(\mathbf{S}(\pi)) \leq \varepsilon$$

and $\mathcal{A}_{(q_1,q_2,\ldots,q_n,t)}$ is the class of all adversaries making at most q_i queries of type i and making at most t computation steps.

For a family of secret-key or public-key encryption schemes, parametrized by a security parameter k, we say that the encryption scheme family $\pi(k)$ is *NOTION secure* if for all k

$$\Phi^{\mathcal{A}_{(q_1,q21,\ldots,q_n,t)}}(\mathbf{S}(\pi(k))) \leq \varepsilon$$

holds if $q_1, q_2, \ldots, q_n$ and t are efficient in k and ε is negligible in k.

4.1.3 Relating Bit-guessing and Distinguishing

Recall the concept of a *distinguisher* from Chapter 2. A distinguisher interacts with either of two systems and outputs a bit W (that can be seen as a guess for which system it is interacting with). An adversary for a uniform bit-guessing game $\mathbf{S}$ is nothing else than an adversary interacting with either of $\mathbf{S}_0$ or $\mathbf{S}_1$, finally outputting a bit V as its guess. The two concepts seem to be identical with the exception of how the guess is presented. The following definition allows to compare bit-guessing game adversaries and distinguishers.

Definition 4.8. *For a system* $\mathbf{W}$ *interacting with a bit-guessing game* $\mathbf{S}$*, we denote by* $\mathbf{W}^\dagger$ *the distinguisher with same input/output behavior as* $\mathbf{W}$*, except that the guess-bit* V *is not input into the game but rather presented as the distinguishers decision bit* W*. We call* $\mathbf{W}^\dagger$ *the* distinguisher variant of $\mathbf{W}$.

For a distinguisher $\mathbf{D}$*, we analogously denote by* $\mathbf{D}^+$ *the bit-guessing game adversary system with the same input/output behavior as* $\mathbf{D}$*, except that the decision bit* W *is input to the game as guess-bit* V*. We call* $\mathbf{D}^+$ *the* bit-guessing game adversary variant of $\mathbf{D}$.

Note that the two transformations $^+$ and † define a duality. Therefore, we have the equivalence relations,

$$\left(\mathbf{W}^\dagger\right)^+ \equiv \mathbf{W} \tag{4.1}$$

$$\left(\mathbf{D}^+\right)^\dagger \equiv \mathbf{D}. \tag{4.2}$$

The two concepts of bit-guessing and distinguishing are indeed congruent, as the following Lemma shows.

Lemma 4.9 (Equivalence of distinguishing and bit-guessing). *If a system* $\mathbf{W}$ *has advantage* ε *in winning the uniform bit-guessing game* $\mathbf{S}$*, then the distinguisher* $\mathbf{W}^\dagger$ *has distinguishing advantage* ε *in distinguishing the two systems* $\mathbf{S}_0$ *and* $\mathbf{S}_1$ *according to Definition 4.3. Conversely, if a distinguisher* $\mathbf{D}$ *has advantage* ε *in distinguishing the two conditional games* $\mathbf{S}_0$ *and* $\mathbf{S}_1$ *of a uniform bit-guessing game* $\mathbf{S}$*, then* $\mathbf{D}^+$ *has advantage* ε *in winning the game* $\mathbf{S}$*.*

$$\Phi^{\mathbf{W}}(\mathbf{S}) = \Delta^{\mathbf{W}^\dagger}(\mathbf{S}_0, \mathbf{S}_1)$$
$$\Delta^{\mathbf{D}}(\mathbf{S}_0, \mathbf{S}_1) = \Phi^{\mathbf{D}^+}(\mathbf{S})$$

Proof. According to Corollary 4.6, we have

$$\begin{aligned}\Phi^{\mathbf{W}}(\mathbf{S}) &= \left| \Gamma^{\mathbf{W}}(\mathbf{S}_0) + \Gamma^{\mathbf{W}}(\mathbf{S}_1) - 1 \right|. \\ &= \left| \sup_{i\geq 1} \mathsf{P}^{\mathbf{W}\mathbf{S}_0}(A_i = 1) + \sup_{j\geq 1} \mathsf{P}^{\mathbf{W}\mathbf{S}_1}(A_j = 1) - 1 \right|.\end{aligned}$$

Clearly, the probability that $\mathbf{W}$ eventually wins game $\mathbf{S}_b$ (i.e. outputs as its first guess $V = b$) is the same as the probability that the distinguisher variant of $\mathbf{W}$, $\mathbf{W}^\dagger$ outputs the distinguishing bit $W = b$ interacting with $\mathbf{S}_b$. Thus,

$$\begin{aligned}\Phi^{\mathbf{W}}(\mathbf{S}) &= \left| \mathsf{P}^{\mathbf{W}^\dagger\mathbf{S}_0}(W = 0) + \mathsf{P}^{\mathbf{W}^\dagger\mathbf{S}_1}(W = 1) - 1 \right| \\ &= \left| (1 - \mathsf{P}^{\mathbf{W}^\dagger\mathbf{S}_0}(W = 1)) + \mathsf{P}^{\mathbf{W}^\dagger\mathbf{S}_1}(W = 1) - 1 \right| \\ &= \left| \mathsf{P}^{\mathbf{W}^\dagger\mathbf{S}_0}(W = 1) - \mathsf{P}^{\mathbf{W}^\dagger\mathbf{S}_1}(W = 1) \right| \\ &= \Delta^{\mathbf{W}^\dagger}(\mathbf{S_0}, \mathbf{S_1})\end{aligned}$$

Replacing $\mathbf{W}$ by $\mathbf{D}^+$ and applying (4.2) in the other direction follows directly. □

4.1.4 Notational Conventions

In the upcoming sections of this chapter, several uniform bit-guessing games according to Definition 4.3 are specified using the pseudo-code characterization from Chapter 2. To keep the definitions short, parts that are common to

all definitions are specified here and skipped later. All definitions share the monotonicity update step of the MBO at the beginning of every round and the outputting of the MBO at the end of every round.

initialize
generate the bit $b \in_R \{0,1\}$
initialize $A_0 \leftarrow 0$

on every **input** X_i **do**
$A_i \leftarrow A_{i-1}$
$\vdots$
on every **input** X_i **do**
output the MBO A_i

Since a uniform bit-guessing game is defined by the existence of two conditional games $\mathbf{S}_0$ and $\mathbf{S}_1$, a general characterization of those games that can be applied for every later definition is given here: If the game $\mathbf{S}$ is defined as above, then its conditional game $\mathbf{S}_d$ is defined in exactly the same way with the exception that the bit b is fixed to the value d. Note that this definition complies with the condition $\mathbf{S} \equiv \mathbf{S}_B, B \in_R \{0,1\}$.

4.2 Finding the Right Attack Model

Game-based security definitions are formalized as a game that an adversary cannot win with "substantial" probability. The capabilities of the adversary in the game are usually defined by one of the three traditional attack models: *chosen-plaintext attacks* (CPA), *chosen-ciphertext attacks* (CCA1) and *adaptive chosen-ciphertext attacks* (CCA2 or often just CCA).

Several works in the literature examine how different game-based notions relate to each other with respect to different attack models. For public-key encryption schemes, a good overview of the most common notions of indistinguishability and non-malleability is presented in [6]. In the context of private-key encryption schemes, a similar outline is provided in [7].

In the channel-based model of constructive cryptography, the capabilities of an adversary are defined implicitly by the model. The adversary (in the role of the distinguisher) gets access to all three interfaces of a channel. If an encryption scheme is applied to a multi-use insecure channel, an adversary can—at any time—produce multiple ciphertexts of chosen messages by inputting the messages at the sender interface, and he can also see the decryption of multiple chosen ciphertexts by inputting them at the adversary's

interface. An attack model like CPA is thus clearly too weak. A first idea could be that CCA should be considered, but a detail in the definition of the attack model renders it too strong.

Chosen-plaintext Attack Model In the chosen-plaintext attack model, the adversary only gets access to an encryption oracle (i.e. no decryption oracle). As pointed out in [6, Theorem 4], the lack of a decryption oracle renders schemes secure that potentially leak the secret key (or private key). It is clear that such a scheme does not comply with our understanding of confidentiality, which is why a stronger attack model must be considered.

Chosen-ciphertext Attack Model The chosen-ciphertext attack model is sufficient in the sense that it additionally gives access to a decryption oracle. The decryption oracle is modified so that decryption of a challenge ciphertext is refused to prevent trivial attacks. It does, however, only refuse the self challenge ciphertext and not a ciphertext that is "equivalent" to the challenge ciphertext (i.e. one that decrypts to the same message). This detail allows trivial attacks for schemes where efficient "re-randomization" of ciphertexts is possible. Therefore such a scheme is not secure under CCA.

In the channel-based view, the ultimate goal of secure communication is a secure channel. A secure channel is an abstraction of a communication channel where only sent messages and the forwarding and deleting of them are considered, but not ciphertexts. Being able to transform a given ciphertext into a new ciphertext decrypting to the same message is thus not of any concern with respect to a secure channel if this modification process can be detected by a simulator and "abstracted" as forwarding the message.

A simple scheme that allows such a behavior can be designed from any scheme that constructs a secure channel by prepending a random bit in the process of encryption, whereas the bit is just ignored during decryption. Clearly, the new scheme is not CCA-secure (neither for the notion of indistinguishability nor for non-malleability), but still constructs a secure channel: A new simulator can be constructed with the same behavior as the old one except that a random bit is prepended to any ciphertext that is output, and the first bit of any input ciphertext is just dropped.

Replayable Chosen-ciphertext Attack Model A relaxed variant of CCA, addressing exactly this unnecessary strictness of CCA, was introduced by Canetti, Krawczyk and Nielsen in [12]. Their attack model is called *replayable chosen-ciphertext attack* (RCCA) and being motivated that security with respect to RCCA is sufficient "for most practical purposes". As argued above, RCCA security is indeed suitable for secure communication.

In the RCCA attack model, both decryption of the challenge ciphertext and ciphertexts decrypting to a challenge message are refused (e.g. by a dummy output 'test').

In conclusion, one can say that neither CPA nor CCA are attack models that are appropriate if one wants to show the equivalence between a game-based security notion and a certain type of confidential channel. The RCCA variant given in [12] is the variant to use, especially for non-malleability and integrity notions.

Allowing Arbitrary Malleability in an Attack Model However, for other non-malleability related notions, RCCA can still be too strong. Imagine, for example, a notion that wants to capture the security properties of CBC. For a given ciphertext generated by CBC encryption, it is possible to modify parts of the ciphertext such that decryption of the modified ciphertext consists of message blocks that are the same as the initial message, and other blocks get randomized. Therefore, a notion capturing CBC must—in contrary to non-malleability—explicitly allow the transformation of a ciphertext into a different ciphertext decrypting to a "related" plaintext. Using the RCCA model for this notion leads to a trivial attack using the above described transformation of a challenge ciphertext since the transformed ciphertext is not refused by the decryption oracle.

For such notions, a more general attack model is needed. In Section 4.5, such a model is presented in order to introduce a game-based notion of "pure" confidentiality (without further restricting the malleability).

4.3 Non-malleability Notions

In the literature, there exist a large number of different definitions for non-malleability notions. It starts with the simulation-based definition in [17], and continues with a definition in [6] that is in the spirit of indistinguishability. Bellare and Sahai [8] introduce a third "pure" indistinguishability based definition and show the equivalence of all three definitions. In a later paper, they provide a corrected version with certain additional assumptions for the equivalence to hold [9]. In [32], all the existing definitions are recompiled and analyzed for all kinds of subtleties (such as restriction to the message space or restriction to the considered adversaries).

A common result of most studies is that the indistinguishability-based definition of non-malleability is equivalent to the indistinguishability-based definition of confidentiality under CCA. This equivalence can easily be transferred to the case of the RCCA attack model because the CCA oracles can simply be replaced by RCCA oracles and the proof is still valid.

Motivated by this, the indistinguishability-based definition of confidentiality under RCCA is preferred to the respective variant of non-malleability to analyze the corresponding type of malleability in the channel-based setting. It may sound counter-intuitive to use a confidentiality notion to make a statement about non-malleability, but due to the equivalence and the fact that the non-malleability definition is slightly more complex it seems reasonable to use the simpler definition.

Following the development of the notion of non-malleability, the public-key case is studied first. Three different types of indistinguishability definitions under RCCA are given and their asymptotic equivalence is shown. The first notion is a simple "left-or-right" definition, allowing only one challenge query. The second is the "real-or-random"-variant of it, whereas the third allows multiple "real-or-random" challenge queries.

Having formalized the games, it is shown that the common counterpart in terms of a channel is a confidential channel allowing forwarding, deleting, replaying and inserting "constant" messages (constant in the sense of independent of messages from the history).

4.3.1 Formalization of the Indistinguishability Games

The three above-mentioned games are formalized here as discrete systems using the previously introduced pseudo-code model. The games are denoted by $\mathbf{S}^{LR}_{RCCA}(\pi)$, $\mathbf{S}^{ROR}_{RCCA}(\pi)$ and $\mathbf{S}^{M-ROR}_{RCCA}(\pi)$. The parameter π corresponds to the encryption scheme that is used in the game and is seen as a pair of (potentially stateful) converters. On initialization of the converters, π_2 generates a secret and a public key and outputs the public key whereas the secret key is kept secret (and can be seen as being stored in the converter).

Definition 4.10 (LR-IND-RCCA game). *Let $\pi = (\pi_1, \pi_2)$ be a public-key encryption scheme according to Definition 3.11 and $\mathbf{S}^{LR}_{RCCA}(\pi)$ be a uniform bit-guessing game according to Definition 4.3, called* LR-IND-RCCA game*, with the following input/output behavior: $\mathbf{S}^{LR}_{RCCA}(\pi)$ has input variables $X_i \in [\{\text{'init'}\}] \dot\cup [\{\text{'LR'}\} \times \mathcal{M} \times \mathcal{M}] \dot\cup [\{\text{'decrypt'}\} \times \mathcal{C}] \dot\cup [\{\text{'guess'}\} \times \{0,1\}]$ as well as the output variables $Y_i \in [\{\text{'pubkey'}\} \times \mathcal{K}] \dot\cup [\{\text{'challenge'}\} \times \mathcal{C}] \dot\cup [\{\text{'decrypted'}\} \times (\mathcal{M} \dot\cup \{\perp, \text{'test'}\})] \dot\cup [\{\text{'guessed'}\}]$ and the MBO. $\mathbf{S}^{LR}_{RCCA}(\pi)$ works as follows:*

initialize
generate the bit $b \in_R \{0,1\}$
initialize π to get public key $\mathrm{K} \in_R \mathcal{K}$
initialize $S \leftarrow \emptyset$

on input $X_i = [\text{'init'}]$ **do**
output $Y_i = [\text{'pubkey'}, \mathrm{K}]$

on first input $X_i = [\text{'LR'}, m_0, m_1]$ **do**
if $|m_0| = |m_1|$ **then**
$c \leftarrow \pi_1(\mathrm{K}, m_b)$
$S \leftarrow S \cup \{m_0, m_1\}$
output $Y_i = [\text{'challenge'}, c]$

on input $X_i = [\text{'decrypt'}, c']$ **do**
if $\pi_2(c') \in S$ **then** $m' \leftarrow$ 'test' **else** $m' \leftarrow \pi_2(c')$
output $Y_i = [\text{'decrypted'}, m']$

on first input $X_i = [\text{'guess'}, d]$ **do**
$A_i \leftarrow A_{i-1} \vee (b = d)$
output $Y_i = [\text{'guessed'}]$

Inputs of type $X_i = [\text{'LR'}, m_0, m_1]$ *are called* LR-queries *and those of type* $X_i = [\text{'decrypt'}, c']$ decryption queries.

Definition 4.11 (LR-IND-RCCA security). *We say that an encryption scheme* π *according to Definition 3.11 is* (ε, q_d, t)-secure in the sense of LR-IND-RCCA *if*

$$\Phi^{\mathcal{A}_{(q_d,t)}}(\mathbf{S}^{LR}_{RCCA}(\pi)) \leq \varepsilon,$$

with q_d *denoting the number of decryption queries.*

We say that the encryption scheme family $\pi(k)$ *is* LR-IND-RCCA secure *if, for all* k, $\pi(k)$ *is* (ε, q_d, t)*-secure in the sense of LR-IND-RCCA and* q_d *and* t *are efficient in* k *and* ε *is negligible in* k.

The ROR variant of the game just replaces the LR-query by a ROR-query.

Definition 4.12 (ROR-IND-RCCA game). *Let* $\pi = (\pi_1, \pi_2)$ *be a public-key encryption scheme and* $\mathbf{S}^{ROR}_{RCCA}(\pi)$ *be a uniform bit-guessing game, called* ROR-IND-RCCA game. $\mathbf{S}^{ROR}_{RCCA}(\pi)$ *has the same input and output variables as* $\mathbf{S}^{LR}_{RCCA}(\pi)$ *except that in the input domain,* $[\{\text{'LR'}\} \times \mathcal{M} \times \mathcal{M}]$ *is replaced by* $[\{\text{'ROR'}\} \times \mathcal{M}]$. $\mathbf{S}^{ROR}_{RCCA}(\pi)$ *works exactly the same as* $\mathbf{S}^{LR}_{RCCA}(\pi)$ *except that the LR-query section is replaced by:*

on first input $X_i = [\text{'ROR'}, m_0]$ **do**
generate $m_1 \in_R \mathcal{M}$ with $|m_1| = |m_0|$
$c \leftarrow \pi_1(\mathrm{K}, m_b)$
$S \leftarrow S \cup \{m_0, m_1\}$

output $Y_i = [\text{'challenge'}, c]$

Accordingly, inputs of type $X_i = [\text{'ROR'}, m_0]$ *are called* ROR-queries.

Definition 4.13 (ROR-IND-RCCA security). *We say, an encryption scheme* π *according to Definition 3.11 is* (ε, q_d, t)-secure in the sense of ROR-IND-RCCA *if*

$$\Phi^{\mathcal{A}_{(q_d,t)}}(\mathbf{S}^{ROR}_{RCCA}(\pi)) \leq \varepsilon.$$

We say that the encryption scheme family $\pi(k)$ *is* ROR-IND-RCCA secure *if, for all* k, $\pi(k)$ *is* (ε, q_d, t)*-secure in the sense of ROR-IND-RCCA and* q_d *and* t *are efficient in* k *and* ε *is negligible in* k.

The third variant of the game allows multiple challenge queries of the form 'ROR'. Additionally, decryption of challenge messages has to be answered in a reasonable way, not just by a 'test'-output. Therefore, the game keeps in the random case a data structure to map the chosen random messages to the real challenge messages and outputs these challenge messages in place of a 'test'. This is an unusual definition, but it can be motivated by the equivalence that will be shown in the following section, and by the easier correspondence to the channel-based setting where forwarding messages from the history is allowed explicitly.

Definition 4.14 (M-ROR-IND-RCCA game). *Let* $\pi = (\pi_1, \pi_2)$ *be a public-key encryption scheme and* $\mathbf{S}^{M-ROR}_{RCCA}(\pi)$ *be a uniform bit-guessing game, called* M-ROR-RCCA game, *with the following input/output behavior:* $\mathbf{S}^{M-ROR}_{RCCA}(\pi)$ *has the same input variables as* $\mathbf{S}^{ROR}_{RCCA}(\pi)$, *as well as the same output variables except that* $[\{\text{'decrypted'}\} \times (\mathcal{M} \dot\cup \{\perp, \text{'test'}\})]$ *is replaced by* $[\{\text{'}decrypted\text{'}\} \times (\mathcal{M} \dot\cup \{\perp\})]$. $\mathbf{S}^{M-ROR}_{RCCA}(\pi)$ *works as follows:*

initialize
generate the bit $b \in_R \{0,1\}$
initialize π to get public key $\mathrm{K} \in_R \mathcal{K}$
initialize $S \leftarrow \emptyset$
initialize empty map $M : \mathcal{M} \to \mathcal{M}$

on input $X_i = [\text{'init'}]$ **do**
output $Y_i = [\text{'pubkey'}, \mathrm{K}]$

on input $X_i = [\text{'ROR'}, m_0]$ **do**
generate $m_1 \in_R \mathcal{M}$ with $|m_1| = |m_0|$
$c \leftarrow \pi_1(\mathrm{K}, m_b)$
$M[m_1] \leftarrow m_0$

output $Y_i = [\text{'challenge'}, c]$

on input $X_i = [\text{'decrypt'}, c']$ **do**
$m' \leftarrow \pi_2(c')$
if $b = 1$ and $M[m']$ is set **then**
$m' \leftarrow M[m']$
output $Y_i = [\text{'decrypted'}, m']$

on first input $X_i = [\text{'guess'}, d]$ **do**
$A_i \leftarrow A_{i-1} \vee (b = d)$
output $Y_i = [\text{'guessed'}]$

Definition 4.15 (M-ROR-IND-RCCA security). *We say, an encryption scheme π according to Definition 3.11 is* $(\varepsilon, q_r, q_d, t)$-secure in the sense of M-ROR-IND-RCCA *if*

$$\Phi^{\mathcal{A}_{(q_r,q_d,t)}}(\mathbf{S}^{M-ROR}_{RCCA}(\pi)) \leq \varepsilon,$$

with q_r being the number of ROR-queries and q_d denoting the number of decryption queries.

We say that the encryption scheme family $\pi(k)$ is M-ROR-IND-RCCA secure *if, for all k, $\pi(k)$ is $(\varepsilon, q_r, q_d, t)$-secure in the sense of M-ROR-IND-RCCA and q_r, q_d and t are efficient in k and ε is negligible in k.*

4.3.2 Equivalence of the Games

In what follows, the asymptotic equivalence of all three notions is shown under the assumption of a "non-efficient" size of the message space (i.e. that the probability to correctly guess a random message from the space is negligible).

Lemma 4.16 (LR-IND-RCCA $\Rightarrow$ ROR-IND-RCCA). *If a public-key encryption scheme family $\pi(k)$ is LR-IND-RCCA secure, then $\pi(k)$ is also ROR-IND-RCCA secure.*

Especially if a specific public-key encryption scheme π is (ε, q_d, t)-secure in the sense of LR-IND-RCCA, then it is also $(\varepsilon, q_d, t - t')$-secure in the sense of ROR-IND-RCCA where t' are the number of computation steps needed to generate a random message.

Proof. Note that the two games to consider are defined almost identically, with the exception that the LR-query body of the LR-IND-RCCA game is replaced by the ROR-query body in the ROR-IND-RCCA game.

We can thus construct a converter $\mathbf{C}$ that forwards all queries except the ROR-query. Getting a message m_0 in the ROR-query, $\mathbf{C}$ generates a second message m_1 of equal length uniformly at random and outputs m_0 and m_1 at the inner interface in a LR-query.

If the converter $\mathbf{C}$ is connected to the game $\mathbf{S}^{LR}_{RCCA}(\pi)$, a ROR-query m_0 provokes $\mathbf{C}$ to generate a random message m_1, $\mathbf{S}^{LR}_{RCCA}(\pi)$ encrypts the message m_b according to the bit b chosen at random and puts m_0 and m_1 into the set S. A ROR-query m_0 to the game $\mathbf{S}^{ROR}_{RCCA}(\pi)$ is treated exactly in the same way: A random message m_1 is generated, m_b is encrypted and both messages are put into S. All other queries, except an ROR-query, are in $\mathbf{S}^{LR}_{RCCA}(\pi)$ trivially answered in the same way as in the game $\mathbf{S}^{ROR}_{RCCA}(\pi)$ according to the definition.

We conclude that the two games are equivalent for any encryption scheme π,

$$\mathbf{S}^{ROR}_{RCCA}(\pi) \equiv \mathbf{C}\mathbf{S}^{LR}_{RCCA}(\pi). \tag{4.3}$$

Let π be a public-key encryption scheme according to Definition 3.11 being (ε, q_d, t)-secure in the sense of LR-IND-RCCA, namely that

$$\Phi^{\mathcal{A}_{(q_d,t)}}(\mathbf{S}^{LR}_{RCCA}(\pi)) \leq \varepsilon.$$

Let $\mathbf{A}$ further be an adversary attacking π in the sense of ROR-IND-RCCA, making q_d decryption queries to the game defined in Definition 4.12, making at most $t - t'$ computation steps and having advantage $\Phi^{\mathbf{A}}(\mathbf{S}^{ROR}_{RCCA}(\pi)) > \varepsilon$. Considering the converter $\mathbf{C}$ from above, we note that the composition $\mathbf{AC}$ makes at most q_d decryption queries since decryption queries are simply forwarded. The number of computation steps of $\mathbf{AC}$ is bounded by t since $\mathbf{A}$ makes at most $t - t'$ steps and the converter does increase the number of steps only in the case of an ROR-query by t', being the number of steps needed to generate a random message. As only one such query is treated by $\mathbf{C}$, the total number of computation steps is bounded by $t - t' + t' = t$. Using the Equivalence 4.3, we can express the advantage of $\mathbf{AC}$ in winning the LR-IND-RCCA game by

$$\begin{aligned}\Phi^{\mathbf{AC}}(\mathbf{S}^{LR}_{RCCA}(\pi)) &= \Phi^{\mathbf{A}}(\mathbf{C}\mathbf{S}^{LR}_{RCCA}(\pi)) \\ &= \Phi^{\mathbf{A}}(\mathbf{S}^{ROR}_{RCCA}(\pi)) > \varepsilon.\end{aligned}$$

As the existence of such an adversary $\mathbf{A}$ contradicts our assumption that π is (ε, q_d, t)-secure in the sense of LR-IND-RCCA, we conclude that no such adversary $\mathbf{A}$ can exist and thus π is $(\varepsilon, q_d, t - t')$-secure in the sense of ROR-IND-RCCA. As we proved this fact for a general encryption scheme π, it particularly holds for all k and corresponding encryption schemes $\pi(k)$ of a family. Assuming that the number of computation steps needed to generate a random message is efficient in k, the asymptotic statement that LR-IND-RCCA security implies ROR-RCCA security, follows directly. □

Lemma 4.17 (ROR-IND-RCCA ⇒ LR-IND-RCCA). *If a public-key encryption scheme family $\pi(k)$ using message space $\mathcal{M}(k)$ of size $|\mathcal{M}(k)|$ is ROR-IND-RCCA secure, then $\pi(k)$ is also LR-IND-RCCA secure if $|\mathcal{M}(k)|$ is not efficient in k.*

Especially if a public-key encryption scheme π with message space $\mathcal{M}$ of size $|\mathcal{M}|$ is (ε, q_d, t)-secure in the sense of ROR-IND-RCCA, then it is $\left(2\varepsilon + \frac{2q_d}{|\mathcal{M}|}, q_d, t - t'\right)$-secure in the sense of LR-IND-RCCA where t' captures some efficiently implementable computation steps.

Proof. Let π be an encryption scheme that is (ε, q_d, t)-secure in the sense of ROR-IND-RCCA, meaning that $\Phi^{\mathcal{A}_{(q_d,t)}}(\mathbf{S}^{ROR}_{RCCA}(\pi)) \leq \varepsilon$. Let $\mathbf{B}$ further be an adversary attacking π in the sense of LR-IND-RCCA, making at most q_d decryption queries to the game defined in Definition 4.10, making at most $t - t'$ computation steps and having advantage

$$\Phi^{\mathbf{B}}(\mathbf{S}^{LR}_{RCCA}(\pi)) > 2\varepsilon + \frac{2q_d}{|\mathcal{M}|}.$$

We can construct a converter $\mathbf{C}$ that we can apply to adversary $\mathbf{B}$ resulting in a system $\mathbf{BC}$ that makes q_d decryption queries and t computation steps and has advantage $\Phi^{\mathbf{BC}}(\mathbf{S}^{ROR}_{RCCA}(\pi)) > \varepsilon$ in winning the LR-IND-RCCA game.

Converter $\mathbf{C}$ works as follows:

initialize
  generate $d \in_R \{0, 1\}$
  initialize $m_{other} \leftarrow \perp$

on first input ['LR', m_0, m_1] at outer interface **do**
  output ['ROR', m_d] at inner interface
  get input ['challenge', c] at inner interface
  $m_{other} \leftarrow m_{1-d}$
  output ['challenge', c] at outer interface

on input ['decrypt', c'] at outer interface **do**
  output ['decrypt', c'] at inner interface
  get input ['decrypted', m'] at inner interface
  if $m_{other} \neq \perp$ and $m' = m_{other}$ **then** $m' \leftarrow$ 'test'
  output ['decrypted', m'] at outer interface

on first input ['guess', b] at outer interface **do**
  output ['guess', $b \oplus d$] at inner interface
  get input ['guessed'] at inner interface
  output ['guessed'] at outer interface

We see that **BC** makes exactly the same number of queries as **B**. The overhead in terms of computation steps induced by converter **C** is the number of steps needed to store one message, to assign a string to a variable and check simple conditions q_d times and once xor-ing two bits, which can be summarized in t'. For analyzing the advantage of **BC** in winning the ROR-IND-RCCA game, we let $\mathbf{S}^{ROR}_{RCCA-0}(\pi)$ and $\mathbf{S}^{ROR}_{RCCA-1}(\pi)$ be the two conditional games of the bit-guessing game $\mathbf{S}^{ROR}_{RCCA}(\pi)$ according to Definition 4.3. Thus, the first conditional game is the ROR-IND-RCCA game that always encrypts the real message as challenge. The second game is the one that always encrypts the random message accordingly.

We note that the game $\mathbf{CS}^{ROR}_{RCCA-0}(\pi)$ is equivalent to $\mathbf{S}^{LR}_{RCCA}(\pi)$, with the exception that the first game includes an additional random message m_r in the set S provoking a 'test' output in decryption for a queried ciphertext decrypting to m_r. This behavior cannot be observed in the second game where the message m_r is output. Let $\mathcal{B} = B_0, B_1, B_2$ denote the MES where B_i is the event that up to query i, no ciphertext decrypting to the randomly chosen message has been input. Thus, we have the conditional equivalence

$$\mathbf{CS}^{ROR}_{RCCA-0}(\pi)|\mathcal{B} \equiv \mathbf{S}^{LR}_{RCCA}(\pi). \tag{4.4}$$

Since every output **B** that gets from the game $\mathbf{CS}^{ROR}_{RCCA-0}(\pi)|\mathcal{B}$ is independent of m_r (as we are in the real case and the real message is encrypted as challenge), the probability that $\mathcal{B}$ no longer holds is, by using the union bound, at most $\frac{1}{|\mathcal{M}|}q_d$.

In the case where a random message is encrypted by the ROR-IND-RCCA game, the winning probability of **B** is $\frac{1}{2}$ since he has to guess a secret random bit d. The bit d is secret to him as all output he gets from the system $\mathbf{CS}^{ROR}_{RCCA-1}(\pi)$ is independent of the bit d:

- The LR-query is answered with the encryption of a random message and is thus independent of the bit d.
- Decryption queries are answered by its real plaintext content except for three cases where a 'test' answer is provided: The ROR-IND-RCCA game returns 'test' if the input ciphertext decrypts to the random message chosen in the LR-query or to m_d. Additionally, the converter **C** replaces the decryption answer by a 'test' answer if the input ciphertext decrypts to m_{1-d}. Since both encryptions of m_d and m_{1-d} are answered by 'test', the decryption query answers are also independent of d.
- The output to a guess query is trivially independent of d.

For our analysis of the advantage of **BC**, we use Corollary 4.6 and the fact that **B** has winning probability $\frac{1}{2}$ in the random case of the ROR-IND-RCCA

game to get

$$\begin{aligned}\Phi^{\mathbf{BC}}(\mathbf{S}^{ROR}_{RCCA}(\pi)) &= \left|\Gamma^{\mathbf{BC}}(\mathbf{S}^{ROR-0}_{RCCA}(\pi)) + \Gamma^{\mathbf{BC}}(\mathbf{S}^{ROR-1}_{RCCA}(\pi)) - 1\right| \\ &= \left|\Gamma^{\mathbf{B}}(\mathbf{CS}^{ROR-0}_{RCCA}(\pi)) + \frac{1}{2} - 1\right|.\end{aligned}$$

Since the advantage of $\mathbf{B}$ in the LR-IND-RCCA game is $\Phi^{\mathbf{B}}(\mathbf{S}^{LR}_{RCCA}(\pi)) > 2\varepsilon + \frac{2q_d}{|\mathcal{M}|}$, we distinguish two cases, namely $\Gamma^{\mathbf{B}}(\mathbf{S}^{LR}_{RCCA}(\pi)) > \frac{1}{2} + \varepsilon + \frac{q_d}{|\mathcal{M}|}$ or $\Gamma^{\mathbf{B}}(\mathbf{S}^{LR}_{RCCA}(\pi)) < \frac{1}{2} - \varepsilon - \frac{q_d}{|\mathcal{M}|}$. Using Lemma 4.7 with the conditional equivalence (4.4) in the first case, we get

$$\begin{aligned}\Gamma^{\mathbf{B}}(\mathbf{CS}^{ROR-0}_{RCCA}(\pi)) &\geq \Gamma^{\mathbf{B}}(\mathbf{S}^{LR}_{RCCA}(\pi)) - \frac{1}{|\mathcal{M}|}q_d \\ &> \frac{1}{2} + \varepsilon + \frac{q_d}{|\mathcal{M}|} - \frac{1}{|\mathcal{M}|}q_d = \frac{1}{2} + \varepsilon.\end{aligned}$$

Using the same Lemma and conditional equivalence yields for the second case

$$\begin{aligned}\Gamma^{\mathbf{B}}(\mathbf{CS}^{ROR-0}_{RCCA}(\pi)) &\leq \Gamma^{\mathbf{B}}(\mathbf{S}^{LR}_{RCCA}(\pi)) + \frac{1}{|\mathcal{M}|}q_d \\ &< \frac{1}{2} - \varepsilon - \frac{q_d}{|\mathcal{M}|} + \frac{1}{|\mathcal{M}|}q_d = \frac{1}{2} - \varepsilon.\end{aligned}$$

In either case, the advantage can thus be lower bound by

$$\begin{aligned}\Phi^{\mathbf{BC}}(\mathbf{S}^{ROR}_{RCCA}(\pi)) &= \left|\Gamma^{\mathbf{B}}(\mathbf{CS}^{ROR-0}_{RCCA}(\pi)) - \frac{1}{2}\right| \\ &> \frac{1}{2} - \frac{1}{2} + \varepsilon = \varepsilon.\end{aligned}$$

As this contradicts our assumption that π is (ε, q_d, t)-secure in the sense of ROR-IND-RCCA, such an adversary $\mathbf{B}$ cannot exist and we conclude that π is $\left(2\varepsilon + \frac{2q_d}{|\mathcal{M}|}, q_d, t - t'\right)$-secure in the sense of LR-IND-RCCA. The asymptotic statement follows directly under the assumption that the size of the message space is not efficient in the security parameter. □

The definitions of the two games allowing only a single challenge query are thus asymptotically equivalent. In the following statements, the multi-challenge version is also shown to be equivalent to these definitions.

Remark 4.18 (M-ROR-RCCA ⇒ ROR-IND-RCCA). *If a public-key encryption scheme family $\pi(k)$ is M-ROR-RCCA secure, then $\pi(k)$ is also ROR-RCCA secure.*

If a public-key encryption scheme π is (ε, q_r, q_d)-secure in the sense of M-ROR-IND-RCCA, then it is also (ε, q_d)-secure in the sense of ROR-IND-RCCA.

This implication is trivial since ROR-IND-RCCA merely is a special case of the M-ROR-RCCA game where only one challenge ciphertext is queried. In other words, any adversary to the ROR-IND-RCCA game is also an adversary to the M-ROR-RCCA game where only one ROR-query is allowed, using a simple converter that replaces the challenge message by a 'test' in decryption query responses.

Lemma 4.19 (ROR-IND-RCCA ⇒ M-ROR-IND-RCCA). *If a public-key encryption scheme family $\pi(k)$ is ROR-IND-RCCA secure, then $\pi(k)$ is also M-ROR-RCCA secure.*

Especially if a public-key encryption scheme π is (ε, q_d, t)-secure in the sense of ROR-IND-RCCA, then it is also $(\varepsilon \cdot q_r, q_r, q_d, t - t')$-secure in the sense of M-ROR-RCCA where t' captures some efficiently implementable computation steps.

Proof. Let π be an encryption scheme that is (ε, q_d, t)-secure in the sense of ROR-IND-RCCA, namely $\Phi^{\mathcal{A}_{q_d,t}}(\mathbf{S}^{ROR}_{RCCA}(\pi)) \leq \varepsilon$. Let $\mathbf{A}$ further be an adversary attacking π in the sense of M-ROR-IND-RCCA, making at most q_r ROR-queries and q_d decryption queries to the game defined in Definition 4.14, making at most $t - t'$ computation steps and having advantage $\Phi^{\mathbf{A}}(\mathbf{S}^{M-ROR}_{RCCA}(\pi)) > \varepsilon \cdot q_r$.

We show the implication using a hybrid argument and the equivalence of bit-guessing and distinguishing stated in Lemma 4.9. The intuition behind the argument is that the adversary $\mathbf{A}$, attacking the M-ROR-IND-RCCA security of π, can easily be transformed into a distinguisher for the two conditional games $\mathbf{S}^{M-ROR}_{RCCA-0}(\pi)$ and $\mathbf{S}^{M-ROR}_{RCCA-1}(\pi)$, the first being the game encrypting the real message given as argument in all q_r ROR-queries, the latter always encrypting a random message. To construct an adversary against the ROR-IND-RCCA security of π, we define a sequence of hybrid systems so that two subsequent systems only differ in one ROR-query as it is the case in the ROR-IND-RCCA game. For that we define the following sequence of converters $\mathbf{C}_i$, where $1 \leq i \leq q_r$:

initialize
  output ['init'] at inner interface
  get input ['pubkey', K] at inner interface
  initialize *challenge* ← 1.
  initialize empty map $M : \mathcal{M} \to \mathcal{M}$

on input ['ROR', m_0] at outer interface **do**
  if *challenge* $< i$ **then**
    generate $m_1 \in_R \mathcal{M}$ with $|m_1| = |m_0|$
    $c \leftarrow \pi(\mathrm{K}, m_1)$
    $M[m_1] \leftarrow m_0$

elseif *challenge* $= i$ **then**
output $['\text{ROR}', m_0]$ at inner interface
get input $['\text{challenge}', c]$ at inner interface
else
generate $m_1 \in_R \mathcal{M}$ with $|m_1| = |m_0|$
$c \leftarrow \pi(\text{K}, m_0)$
$M[m_1] \leftarrow m_0$
output $['\text{challenge}', c]$ at outer interface

on input $['\text{decrypt}', c']$ at outer interface **do**
output $['\text{decrypt}', c']$ at inner interface
get input $['\text{decrypted}', m']$ at inner interface
if $M[m']$ is set **then**
$m' \leftarrow M[m']$
output $['\text{decrypted}', m']$ at outer interface

on first input $['\text{guess}', b]$ at outer interface **do**
forward the query and the response

In conclusion, we note that in the case of ROR-queries, $\mathbf{C}_i$ simulates the M-ROR-IND-RCCA game by always encrypting a random message in the first $i-1$ queries, using the ROR-query in the i-th query and answering the remaining queries with the encryption of the real message given as argument. Encryption can be done by the converter using the public key. Decryption queries are also slightly adapted so that ciphertexts decrypting to one of the challenge messages are answered by the corresponding real challenge message, as it is done in the M-ROR-RCCA game. Guess queries are forwarded unchanged.

Considering the number of queries that the composed adversary system $\mathbf{AC}_i$ makes, we note that the number of ROR-queries is exactly 1 and that the number of decryption queries is q_d. The overhead in terms of computation steps is given by t', capturing a query to get the public-key, generation of random messages, encryption using the public-key and mapping messages onto integers in the order of q_r. In other words the system $\mathbf{AC}_i$ at most $t - t' + t' = t$ computation steps.

For a comparison of the particular hybrid systems, we use the two conditional games, $\mathbf{S}^{M-ROR}_{RCCA-0}(\pi)$ and $\mathbf{S}^{M-ROR}_{RCCA-1}(\pi)$ and further define $\mathbf{S}^{ROR}_{RCCA-0}(\pi)$ and $\mathbf{S}^{ROR}_{RCCA-1}(\pi)$ to be the two conditional games of the ROR-IND-RCCA game $\mathbf{S}^{ROR}_{RCCA}(\pi)$ according to Definition 4.3.

Therefore, we have for $1 \leq i < q_r$ the equivalences

$$\begin{aligned}\mathbf{C}_1\mathbf{S}^{ROR}_{RCCA-0}(\pi) &\equiv \mathbf{S}^{M-ROR}_{RCCA-0}(\pi)\\ \mathbf{C}_{q_r}\mathbf{S}^{ROR}_{RCCA-1}(\pi) &\equiv \mathbf{S}^{M-ROR}_{RCCA-1}(\pi)\\ \mathbf{C}_{i+1}\mathbf{S}^{ROR}_{RCCA-0}(\pi) &\equiv \mathbf{C}_i\mathbf{S}^{ROR}_{RCCA-1}(\pi).\end{aligned}$$

Using these equivalences, we can apply Theorem 2.15 with $F := \mathbf{S}^{M-ROR}_{RCCA-0}(\pi)$ and $G := \mathbf{S}^{M-ROR}_{RCCA-1}(\pi)$ to get for any distinguisher $\mathbf{D}$,

$$\Delta^{\mathbf{D}}\left(\mathbf{S}^{M-ROR}_{RCCA-0}(\pi), \mathbf{S}^{M-ROR}_{RCCA-1}(\pi)\right) = q_r \cdot \Delta^{\mathbf{DC}_I}(\mathbf{S}^{ROR}_{RCCA-0}(\pi), \mathbf{S}^{ROR}_{RCCA-1}(\pi)) \tag{4.5}$$

where I is chosen uniformly from $\{1, \ldots, q_r\}$. Let now be $\mathbf{A}^\dagger$ be the distinguisher variant of our adversary $\mathbf{A}$ according to Definition 4.8 so that we can apply Lemma 4.9 and get

$$\Phi^{\mathbf{A}}(\mathbf{S}^{M-ROR}_{RCCA}(\pi)) = \Delta^{\mathbf{A}^\dagger}\left(\mathbf{S}^{M-ROR}_{RCCA-0}(\pi), \mathbf{S}^{M-ROR}_{RCCA-1}(\pi)\right). \tag{4.6}$$

Combining (4.5) and (4.6), we get

$$\Phi^{\mathbf{A}}(\mathbf{S}^{M-ROR}_{RCCA}(\pi)) = q_r \cdot \Delta^{\mathbf{A}^\dagger\mathbf{C}_I}(\mathbf{S}^{ROR}_{RCCA-0}(\pi), \mathbf{S}^{ROR}_{RCCA-1}(\pi)), \tag{4.7}$$

and by applying Lemma 4.9 and using the fact that converter $\mathbf{C}_I$ just forwards the guess V and thus $(\mathbf{AC}_I)^\dagger \equiv \mathbf{A}^\dagger\mathbf{C}_I$, we have constructed adversary $\mathbf{AC}_I$ for the ROR-IND-RCCA game with the advantage

$$\begin{aligned}\Phi^{\mathbf{AC}_I}(\mathbf{S}^{ROR}_{RCCA}(\pi)) &= \Delta^{(\mathbf{AC}_I)^\dagger}(\mathbf{S}^{ROR}_{RCCA-0}(\pi), \mathbf{S}^{ROR}_{RCCA-1}(\pi))\\ &= \Delta^{\mathbf{A}^\dagger\mathbf{C}_I}(\mathbf{S}^{ROR}_{RCCA-0}(\pi), \mathbf{S}^{ROR}_{RCCA-1}(\pi))\\ &= \frac{\Phi^{\mathbf{A}}(\mathbf{S}^{M-ROR}_{RCCA}(\pi))}{q_r}\\ &> \frac{\varepsilon \cdot q_r}{q_r} = \varepsilon.\end{aligned}$$

The adversary $\mathbf{AC}_I$ makes, as we stated above for a general $\mathbf{C}_i$, at most q_d decryption queries and t computation steps, and has an advantage greater than ε. As this contradicts our assumption that π is (ε, q_d, t)-secure in the sense of ROR-IND-RCCA, we conclude that such an adversary $\mathbf{A}$ cannot exist and thus π is $(\varepsilon \cdot q_r, q_r, q_d, t - t')$-secure in the sense of M-ROR-IND-RCCA. The asymptotic implication follows directly noting that $\varepsilon \cdot q_r$ is still negligible if q_r is efficient in k. □

Secret-key versions of the Games All three game definitions can easily be adapted for secret-key encryption schemes. In contrast to public-key

schemes, where the knowledge of the public-key allows encryption of messages, encryption has to be allowed explicitly by the game. That is to say that the definitions of the games are modified by removing the initialization query that outputs the public-key and introducing encryption queries instead, where a message passed as argument is encrypted with the secret key.

For the concrete security statements, this change induces a new parameter q_e denoting the number of encryption queries made to the game. Let $\mathbf{S}_{RCCA}^{LR-secret}(\pi)$ denote the secret-key scheme variant of the LR-IND-RCCA game. LR-IND-RCCA security is therefore defined for the secret-key case as follows.

Definition 4.20 (LR-IND-RCCA security). *We say that a secret-key encryption scheme π according to Definition 3.7 is* $(\varepsilon, q_e, q_d, t)$-secure in the sense of LR-IND-RCCA *if*

$$\Phi^{\mathcal{A}_{(q_e,q_d,t)}}(\mathbf{S}_{RCCA}^{LR-secret}(\pi)) \leq \varepsilon,$$

with q_e denoting the number of encryption and q_d the number of decryption queries.

We say that the secret-key encryption scheme family $\pi(k)$ is LR-IND-RCCA secure *if, for all k, $\pi(k)$ is $(\varepsilon, q_e, q_d, t)$-secure in the sense of LR-IND-RCCA and q_e, q_d and t are efficient in k and ε is negligible in k.*

The asymptotic equivalence also holds for the secret-key versions of the three games. The precise statements and proofs are omitted.

4.3.3 Non-malleable Confidential Channel

Intuitively, it is clear that an encryption scheme that is non-malleable in the game-based sense allows to create ciphertexts of constant (or "fresh") messages by encrypting such a message using the public-key (or the encryption oracle in the secret-key case). To put another way, the corresponding malleability has to be permitted also in the analog confidential channel such a scheme constructs. Similarly, it is clear that allowing "some more" malleability (e.g. to create a ciphertext decrypting to a message that depends on the message history), would let an adversary easily win the game by applying this malleability on the challenge ciphertext and learning information about the challenge.

We thus define the non-malleable confidential channel to consist only of "constant" transformations.

Definition 4.21. *Let $F : \mathcal{M}^* \times \mathcal{M}^* \to \mathcal{M}$ be a transformation on the plaintext space. The transformation F is called* constant *if there exists an $m_c \in \mathcal{M}$ such that*

for all $M \in \mathcal{M}^*$ and all $M' \in \mathcal{M}^*$,

$$F(M, M') = m_c.$$

Definition 4.22. *Let the malleability $\mathcal{F}$ be defined as $\mathcal{F} = (\{F_\alpha\}_{\alpha\in\mathcal{A}}, \{\mathcal{A}_i\}_{i\in\mathbb{N}})$. And let $\longrightarrow\!\!\bullet$ be a confidential channel with malleability $\mathcal{F}$. Then, we call $\longrightarrow\!\!\bullet$ a* non-malleable confidential channel *denoted by $\xrightarrow{NM}\!\!\bullet$ if all $\alpha \in \mathcal{A}$ correspond to constant transformations according to Definition 4.21.*

A public-key encryption scheme is non-malleable *if it constructs a non-malleable confidential channel $\xrightarrow{NM}\!\!\bullet$ from the resource $(\longrightarrow \| \longleftarrow\!\!\bullet)$.*

A secret-key encryption scheme is non-malleable *if it constructs a non-malleable confidential channel $\xrightarrow{NM}\!\!\bullet$ from the resource $(\longrightarrow \| \bullet\!\!\longleftrightarrow\!\!\bullet)$.*

The following two theorems state that a public-key encryption scheme family is secure with respect to the three game notions if, and only if, the encryption scheme family constructs a non-malleable channel from an insecure channel and an (inverted) authenticated channel.

Theorem 4.23. *If a public-key encryption scheme family $\pi(k)$ is M-ROR-IND-RCCA secure, then $\pi(k)$ is non-malleable and constructs a non-malleable confidential channel $\xrightarrow{NM}\!\!\bullet$ from the resource $(\longrightarrow \| \longleftarrow\!\!\bullet)$.*

Especially if a public-key encryption scheme $\pi = (\pi_1, \pi_2)$ is $(\varepsilon, q_r, q_d, t)$-secure in the sense of M-ROR-IND-RCCA, then π constructs a non-malleable confidential channel $\xrightarrow{NM}\!\!\bullet$ with error ε from the resource $(\longrightarrow \| \longleftarrow\!\!\bullet)$, namely there exists a simulator σ in $\Sigma_{t'}$ such that $\Delta^{\mathcal{A}_{(q_r,q_d,t)}}(\pi_1^A \pi_2^B(\longrightarrow \| \bullet\!\!\longleftrightarrow\!\!\bullet), \sigma^E(\xrightarrow{NM}\!\!\bullet)) \leq \varepsilon$ where $\Sigma_{t'}$ is the class of all converters bounded to at most t' computation steps and $\mathcal{A}_{(q_r,q_d,t)}$ is the class of all distinguishers making at most q_r queries at the A-interface, at most q_d queries at the E-interface and at most t computation steps.

Proof. Let π be an encryption scheme that is $(\varepsilon, q_r, q_d, t)$-secure in the sense of M-ROR-IND-RCCA, $\Phi^{\mathcal{A}_{(q_r,q_d,t)}}(\mathbf{S}_{RCCA}^{M-ROR}(\pi)) \leq \varepsilon$. Let $\xrightarrow{NM}\!\!\bullet$ be the non-malleable confidential channel working as follows: On input m at the A-interface, it keeps track of the message history and outputs $|m|$ at the E-interface. On input ['forward', i], $\xrightarrow{NM}\!\!\bullet$ outputs the i-th message from the input history at the A-interface, on input ['replay', i], it outputs the i-th message from the output history at the B-interface that corresponds to outcomes of transformations. On input on input ['delete'] it outputs $\perp$ and on input ['modify', m'], it outputs m'. Clearly this channel is non-malleable according to the definition.

Let σ be the simulator working as follows:

initialize
K $\in_R \mathcal{K}$
$round \leftarrow 1$
initialize empty map $M : \mathcal{M} \rightarrow \mathbb{N}$

on input l at inner interface **do**
generate $m' \in_R \mathcal{M}$ with $|m'| = l$
$M[m'] \leftarrow round$
$c \leftarrow \pi_1(\mathrm{K}, m')$
$round \leftarrow round + 1$
output c at outer interface

on input c' at outer interface **do**
$m \leftarrow \pi_2(\mathrm{K}, c')$
if $M[m]$ is set **then**
output ['forward', $M[m]$] at the inner interface
elseif $m = \perp$ **then**
output ['delete'] at the inner interface
else
output ['modify', m] at inner interface

The quantity t' is defined as the number of computation steps the above defined simulator σ makes at most.

Let **D** be a distinguisher making at most q_r queries at the A-interface, at most q_d queries at the E-interface and making at most t computation steps. Distinguisher **D** can distinguish the two resources $\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ and $\sigma^E(\overset{NM}{\longrightarrow}\!\!\bullet)$ with advantage:

$$\Delta^{\mathbf{D}}\left(\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \sigma^E(\overset{NM}{\longrightarrow}\!\!\bullet)\right) > \varepsilon \tag{4.8}$$

We now show that one can construct a converter **C** so that the bit-guessing variant of the composition **DC** according to Definition 4.8 results in a M-ROR-IND-RCCA adversary $(\mathbf{DC})^+$ with advantage greater than ε.

Converter **C** works as follows:

on input m at the outer A-subinterface **do**
output ['ROR', m] at the inner interface
get input ['challenge', c] at inner interface
output c at the outer E-subinterface

on input c' at the outer E-subinterface **do**
output $['\text{decrypt}', c']$ at the inner interface
get input $['\text{decrypted}', m']$ **then**
output m' at the outer B-subinterface

Analyzing the system composition $(\mathbf{DC})^+$, we note that it makes at most q_r ROR- and q_d decryption queries. And since $\mathbf{C}$ only forwards queries and makes no additional computation steps, the composed system makes at most t computation steps.

Let $\mathbf{S}^{M-ROR}_{RCCA-0}(\pi)$ and $\mathbf{S}^{M-ROR}_{RCCA-1}(\pi)$ in the following be the two conditional games of the M-ROR-IND-RCCA game, the first being the M-ROR-IND-RCCA game always encrypting the real message in the ROR-queries, and the second being the analogous one always encrypting a random message.

Comparing $\mathbf{CS}^{M-ROR}_{RCCA-0}(\pi)$ and $\pi^A_1\pi^B_2(\longrightarrow \parallel \bullet\!\!\longmapsto\!\!\bullet)$, we see that the output to queries at the A-interface are distributed identically and thus, simulation of the channel is perfect at the A-interface: The encryption of the input message is returned. Also, the distribution of the output to queries at the E-interface is exactly the same as the decryption of the input ciphertext is returned in both systems. The two systems are thus equivalent,

$$\pi^A_1\pi^B_2(\longrightarrow \parallel \bullet\!\!\longmapsto\!\!\bullet) \equiv \mathbf{CS}^{M-ROR}_{RCCA-0}(\pi) \tag{4.9}$$

Comparing the other system $\mathbf{CS}^{M-ROR}_{RCCA-1}(\pi)$ with $\sigma^E(\overset{NM}{\longrightarrow}\!\!\bullet)$, we again see that the output to queries at the A-interface is distributed identically since a random message is encrypted and output. For the comparison of the output at the E-interface, we distinguish between the case where a ciphertext decrypting to one of the random messages (chosen internally) is input and the case where a ciphertext decrypting to a different message or to $\perp$ is input. In the first case, the game outputs the corresponding real message that the random message maps to. The channel behaves identically as the simulator keeps an analogous map, mapping random messages to rounds, and tells the channel to forward the real message from the corresponding round. In the second case, the two systems behave identically since the correct decryption of the input ciphertext is output. We conclude that the two systems are equivalent,

$$\sigma^E(\overset{NM}{\longrightarrow}\!\!\bullet) \equiv \mathbf{CS}^{M-ROR}_{RCCA-0}(\pi) \tag{4.10}$$

Using Lemma 4.9, (4.2) and the equivalences (4.9) and (4.10), the advantage

of $(\mathbf{DC})^+$ in the M-ROR-RCCA game is

$$\begin{aligned}\Phi^{(\mathbf{DC})^+}(\mathbf{S}^{M-ROR}_{RCCA}(\pi)) &= \Delta^{((\mathbf{DC})^+)^\dagger}(\mathbf{S}^{M-ROR}_{RCCA-0}(\pi), \mathbf{S}^{M-ROR}_{RCCA-1}(\pi)) \\ &= \Delta^{\mathbf{DC}}(\mathbf{S}^{M-ROR}_{RCCA-0}(\pi), \mathbf{S}^{M-ROR}_{RCCA-1}(\pi)) \\ &= \Delta^{\mathbf{D}}(\mathbf{CS}^{M-ROR}_{RCCA-0}(\pi), \mathbf{CS}^{M-ROR}_{RCCA-1}(\pi)) \\ &= \Delta^{\mathbf{D}}(\pi^A_1\pi^B_2(\longrightarrow \parallel \bullet\!\!\longrightarrow), \sigma^E(\overset{NM}{\longrightarrow}\!\!\bullet)) > \varepsilon.\end{aligned}$$

As this contradicts our assumption that π is $(\varepsilon, q_r, q_d, t)$-secure in the sense of M-ROR-IND-RCCA, we conclude that such a distinguisher $\mathbf{D}$ cannot exist and thus π does indeed construct a non-malleable confidential channel $\overset{NM}{\longrightarrow}\!\!\bullet$ with error ε from the resource $(\longrightarrow \parallel \longleftarrow\!\!\bullet)$. The asymptotic statement follows since both game adversary and distinguisher are from the same class of adversaries and are thus likewise in the class of efficient adversaries. The simulator is efficient since the quantity t' refers to an efficient number of encryptions, decryptions and other operations that are efficient in a reasonable model (otherwise the M-ROR-IND-RCCA game would itself not be efficient). □

Theorem 4.24. *If a public-key encryption scheme family $\pi(k)$ constructs a non-malleable confidential channel $\overset{NM}{\longrightarrow}\!\!\bullet$, then $\pi(k)$ is LR-IND-RCCA secure.*

Especially if the public-key encryption scheme π constructs a non-malleable confidential channel $\overset{NM}{\longrightarrow}\!\!\bullet$ with error ε from $(\longrightarrow \parallel \longleftarrow\!\!\bullet)$ with respect to the distinguisher class $\mathcal{A}_{(q_A,q_E,t)}$ of all distinguishers making at most q_A query at the A-interface, q_E queries at the E-interface and at most t computation steps, and with respect to the converter class $\Sigma_{t''}$ consisting of all converters making at most t'' computation steps, then π is $(2\varepsilon, q_E, t - t')$-secure in the sense of LR-IND-RCCA where t' captures some efficiently implementable computation steps.

Proof. Let π be an encryption scheme constructing a non-malleable confidential channel from $(\longrightarrow \parallel \bullet\!\!\longrightarrow)$ for the class $\mathcal{A}_{(q_A,q_E,t)}$ with error ε, formally

$$\exists\, \sigma \in \Sigma_{t''} : \Delta^{\mathcal{A}_{(q_A,q_E,t)}}(\pi^A_1\pi^B_2(\longrightarrow \parallel \bullet\!\!\longrightarrow), \sigma^E \overset{NM}{\longrightarrow}\!\!\bullet) \leq \varepsilon.$$

Let $\mathbf{A}$ further be a LR-IND-RCCA adversary with advantage $\Phi^{\mathbf{A}}(\mathbf{S}^{LR}_{RCCA}(\pi)) > 2\varepsilon$, making at most q_E decryption queries and at most $t - t'$ computation steps.

Towards a contradiction, we show that we can construct a converter $\mathbf{C}$ such that the combined system $\mathbf{AC}$ results in a distinguisher with advantage greater than ε distinguishing the above two channels. Converter $\mathbf{C}$ works as follows:

initialize
generate $b \in_R \{0,1\}$
initialize $challenge \leftarrow 0$
initialize $S \leftarrow \emptyset$

on first input $X_i = [\text{'LR'}, m_0, m_1]$ at outer interface **do**
if $|m_0| = |m_1|$ **then**
output m_b at inner A-subinterface
get input c at inner E-subinterface
$S \leftarrow S \cup \{m_0, m_1\}$
output $Y_i = [\text{'challenge'}, c]$ at outer interface

on input $X_i = [\text{'decrypt'}, c')]$ at outer interface **do**
output c' at inner E-subinterface
get input m' at inner B-subinterface
if $m' \in S$ **then** $m' \leftarrow$ 'test'
output $Y_i = [\text{'decrypted'}, m']$ at outer interface

on first input $X_i = [\text{'guess'}, d]$ at outer interface **do**
output $W \leftarrow (b = d)$ at inner interface
output $Y_i = [\text{'}guessed\text{'}]$ at outer interface

The composed system **AC** results in a distinguisher making 1 query at the A-interface and q_E queries at the E-interface. The overhead in terms of computation steps imposed by **C** is the number needed to generate one random bit, putting two messages in a set and checking q_E times if a message is contained in the set and finally checking the equality of two bits, captured in t' and thus leading to an upper bound of computation steps for **AC** of $t - t' + t' = t$. We conclude that $\mathbf{AC} \in \mathcal{A}_{(q_A, q_E, t)}$.

In the following we show that this distinguisher has advantage greater than ε, formally:

$$\forall \sigma' \in \Sigma_{t''} : \Delta^{\mathbf{AC}}(\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \sigma'^E \xrightarrow{NM} \bullet) > \varepsilon.$$

Comparing the two systems **C** $\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ and $\mathbf{S}^{LR}_{RCCA}(\pi)$, we see that for a LR-query both systems randomly encrypt one of the two argument messages based on a secret random bit. Additionally, both messages are put into a set S. For a decryption query c, both system output $\pi_2(K, c)$, except in the case where the decrypted message is contained in the set S where 'test' is returned. Since the sets in both systems are filled accordingly, the output to decryption queries is distributed identically.

Finally, we see that whenever $\mathbf{A}$ wins the game LR-IND-RCCA, that is, when he inputs the correct bit $d = b$, our constructed distinguisher $\mathbf{AC}$ will output the guess bit $W = 1$. We thus get

$$\mathsf{P}^{\mathbf{AC}\pi_1^A\pi_2^B(\longrightarrow\|\bullet\!\!=\!\!=\!\!\bullet)}(W = 1) = \Gamma^{\mathbf{A}}(\mathbf{S}^{LR}_{RCCA}(\pi)).$$

We further analyze the system $\mathbf{C}\,\sigma'^E(\xrightarrow{NM}\!\!\bullet)$ with an arbitrary simulator σ' by inspecting the dependency of the output on the bit b chosen by the converter at random during initialization.

1. $'LR'$-query: Message m_b is input into the channel, so the input depends on the bit b. But as the channel $\xrightarrow{NM}\!\!\bullet$ does per definition not leak any information about the message to the simulator, except length of the message, and because we have the condition that $|m_0| = |m_1|$ for the two messages in the LR-query, the input of the simulator is independent of b and the output of the simulator is independent of b.

2. $'decrypt'$-query: We distinguish the cases where a ciphertext is input, disposing the simulator to forward or replay the challenge message m_b, and cases where the input disposes the simulator to forward or replay another message independent from b or to do something else like deleting the message or inserting a new message. In the latter case, the output is trivially the same for both values of b and thus independent of b. In the first case, however, the channel outputs the message m_b to the converter. But since the converter replaces the message m_b by a 'test' message, this output is also the same for both values of b and hence does not depend on b.

We conclude that with any simulator σ' (constrained to any bound t'' on the number of computation steps), any adversary $\mathbf{A}'$ sees equally distributed output—for both values of b—interacting with the system $\mathbf{C}\,\sigma'^E(\xrightarrow{NM}\!\!\bullet)$, thus the adversary must guess a secret bit chosen uniformly at random and guesses correctly with probability $\frac{1}{2}$. In particular, this holds for adversary $\mathbf{A}$,

$$\forall\ \sigma' : \Gamma^{\mathbf{A}}(\mathbf{C}\,\sigma'^E(\xrightarrow{NM}\!\!\bullet)) = \frac{1}{2}.$$

And thus,

$$\forall\ \sigma' : \mathsf{P}^{\mathbf{AC}\sigma'^E(\xrightarrow{NM}\!\!\bullet)}(W = 1) = \frac{1}{2}.$$

When combining these results, we get a distinguishing advantage of $\mathbf{AC}$ of

$$\begin{aligned}
&\forall\, \sigma' : \Delta^{\mathbf{AC}}(\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!\longmapsto), \sigma'^E \xrightarrow{NM}\bullet) \\
&\qquad = \left| \mathsf{P}^{\mathbf{AC}\pi_1^A\pi_2^B(\longrightarrow\parallel\bullet\!\!\longmapsto)}(W=1) - \mathsf{P}^{\mathbf{AC}\sigma'^E(\xrightarrow{NM}\bullet)}(W=1) \right| \\
&\qquad = \left| \Gamma^{\mathbf{A}}(\mathbf{S}^{LR}_{RCCA}(\pi)) - \frac{1}{2} \right| \\
&\qquad = \frac{1}{2} \cdot \Phi^{\mathbf{A}}(\mathbf{S}^{LR}_{RCCA}(\pi)) \\
&\qquad > \frac{1}{2} \cdot 2\varepsilon = \varepsilon.
\end{aligned}$$

As this contradicts our assumption that π constructs a non-malleable confidential channel $\xrightarrow{NM}\bullet$ with error ε from $(\longrightarrow \parallel \bullet\!\!\longmapsto)$, no such adversary $\mathbf{A}$ can exist and π must be $(2\varepsilon, q_E, t - t')$-secure in the sense of LR-IND-RCCA. The asymptotic statement follows directly. □

Again, the statements can be adapted for the secret-key case. The arguments in the proofs of the two theorems can almost directly be transferred to the secret-key case.

4.4 Unforgeable Encryption

In contrast to public-key encryption schemes where a malleability that allows inserting constant messages is the best one can achieve, it makes sense to study even stronger notions for secret-key schemes since encryption intuitively should only be possible when the secret-key is known. One idea is to define a notion capturing the inability of an adversary to create a valid ciphertext. The notions of *integrity of plaintexts* [7, 19], *integrity of ciphertexts* [7], and *existential unforgeability* [22, 19] are notions in that spirit.

The concepts of integrity of encryption and unforgeability of a valid encryption cover essentially the same idea. As an adversary usually has access to an "encryption oracle", only "new" encryptions are considered as "non-integer encryptions" or forgeries. This can be defined both on the level of the plaintext space (new meaning decrypting to a new message) and on the level of the ciphertext space (new as an unseen ciphertext). As pointed out in Section 4.2, only the level of the plaintext space must be considered in the channel-based model, which is why the notion of integrity of plaintexts is the most promising for a comparison.

It must further be pointed out that it makes no sense to investigate integrity alone as an encryption scheme is only reasonable if it satisfies also its pri-

mary goal, namely confidentiality. To cover such a combined goal, a standard real-or-random game with multiple challenges is stated to guarantee confidentiality. In addition the game gives credit to ciphertext forgeries of the above mentioned form by outputting the bit to guess from the real-or-random game, which trivially allows the adversary to win the game.

4.4.1 Formalization of the Game

Definition 4.25 (Unforgeability game). *Let $\pi = (\pi_1, \pi_2)$ be a secret-key encryption scheme according to Definition 3.7 and $\mathbf{S}^{EF}_{RCCA}(\pi)$ be a bit-guessing game according to Definition 4.3, called* unforgeability game *with the following input/output behavior: $\mathbf{S}^{EF}_{RCCA}(\pi)$ has input variables $X_i \in [\{'encrypt'\} \times \mathcal{M}] \dot\cup [\{'decrypt'\} \times \mathcal{C}] \dot\cup [\{'guess'\} \times \{0,1\}]$ as well as the output variables $Y_i \in [\{'encrypted'\} \times \mathcal{C}] \dot\cup [\{'decrypted'\} \times (\mathcal{M} \dot\cup \{\perp\})] \dot\cup [\{'won'\} \times \mathcal{M} \times \{0,1\}] \dot\cup [\{'guessed'\}]$ and the MBO. $\mathbf{S}^{EF}_{RCCA}(\pi)$ works as follows:*

initialize
  generate key $\mathrm{K} \in_R \mathcal{K}$
  generate the bit $b \in_R \{0,1\}$
  initialize $S \leftarrow \emptyset$
  initialize empty map $M : \mathcal{M} \to \mathcal{M}$

on input $X_i = [\text{'encrypt'}, m]$ **do**
  if $b = 0$ **then**
    $S \leftarrow S \cup \{m\}$
    output $Y_i = [\text{'encrypted'}, \pi_1(\mathrm{K}, m)]$
  else
    generate $m' \in_R \mathcal{M}$ with $|m'| = |m|$
    $M[m'] = m$
    $S \leftarrow S \cup \{m'\}$
    output $Y_i = [\text{'encrypted'}, \pi_1(\mathrm{K}, m')]$

on input $X_i = [\text{'decrypt'}, c]$ **do**
  if $\pi_2(\mathrm{K}, c) = \perp$ **then**
    output $Y_i = [\text{'decrypted'}, \perp]$
  elseif $\pi_2(\mathrm{K}, c) \notin S$ **then**
    output $Y_i = [\text{'won'}, \pi_2(\mathrm{K}, c), b]$
  elseif $b = 1$ **then**
    output $Y_i = [\text{'decrypted'}, M[\pi_2(\mathrm{K}, c)]]$
  else
    output $Y_i = [\text{'decrypted'}, \pi_2(\mathrm{K}, c)]$

on first input $X_i = [\text{'guess'}, d]$ **do**

$A_i \leftarrow A_{i-1} \vee (b = d)$
output $Y_i = [\text{'guessed'}]$

Inputs of type $X_i = [\text{'encrypt'}, m]$ *are called* encryption queries*, and those of type* $X_i = [\text{'decrypt'}, c']$ decryption queries*.*

Definition 4.26 (EF-RCCA security). *We say that a secret-key encryption scheme* π *according to Definition 3.7 is* $(\varepsilon, q_e, q_d, t)$*-secure in the sense of EF-RCCA if*

$$\Phi^{\mathcal{A}_{(q_e,q_d,t)}}(\mathbf{S}^{EF}_{RCCA}(\pi)) \leq \varepsilon,$$

with q_e *being the number of encryption queries and* q_d *denoting the number of decryption queries.*

We say that the secret-key encryption scheme family $\pi(k)$ *is EF-RCCA secure if, for all* k*,* $\pi(k)$ *is* $(\varepsilon, q_e, q_d, t)$*-secure in the sense of EF-RCCA and* q_e, q_d *and* t *are efficient in* k *and* ε *is negligible in* k*.*

4.4.2 Equivalence Results

The following two theorems show that an encryption is EF-RCCA secure if, and only if, it constructs a confidential channel with no malleability, namely a secure channel $\bullet\!\!\longrightarrow\!\!\bullet$, from an insecure channel $\longrightarrow$ and a shared secret key $\bullet\!\!=\!\!\bullet$.

Theorem 4.27 (EF-RCCA $\Rightarrow$ Secure channel $\bullet\!\!\longrightarrow\!\!\bullet$). *If a secret-key encryption scheme family* $\pi(k)$ *is EF-RCCA secure, then* $\pi(k)$ *constructs a secure channel* $\bullet\!\!\longrightarrow\!\!\bullet$ *from the resource* $(\longrightarrow \| \bullet\!\!=\!\!\bullet)$ *.*

Especially if a secret-key encryption scheme $\pi = (\pi_1, \pi_2)$ *is* $(\varepsilon, q_e, q_d, t)$*-secure in the sense of EF-RCCA, then* π *constructs a secure channel* $\bullet\!\!\longrightarrow\!\!\bullet$ *with error* ε *from the resource* $(\longrightarrow \| \bullet\!\!=\!\!\bullet)$ *for the distinguisher class* $\mathcal{A}_{(q_e,q_d,t)}$ *and the converter class* $\Sigma_{t'}$ *where* $\mathcal{A}_{(q_e,q_d,t)}$ *is the class of all distinguishers making at most* q_e *queries at the A-interface and at most* q_d *queries at the E-interface.*

Proof. Let π be a secret-key encryption scheme that is $(\varepsilon, q_e, q_d, t)$-secure in the sense of EF-RCCA, namely that $\Phi^{\mathcal{A}_{(q_e,q_d,t)}}(\mathbf{S}^{EF}_{RCCA}(\pi)) \leq \varepsilon$. Let σ further be the simulator working as follows:

initialize
$K \in_R \mathcal{K}$
round $\leftarrow 1$
initialize empty map $M : \mathcal{M} \rightarrow \mathbb{N}$

on input $|m|$ at inner interface **do**
generate $m' \in_R \mathcal{M}$ with $|m'| = |m|$
$M[m'] \leftarrow round$
$c \leftarrow \pi_1(\mathrm{K}, m')$
$round \leftarrow round + 1$
output c at outer interface

on input c' at outer interface **do**
$m \leftarrow \pi_2(\mathrm{K}, c')$
if $M[m]$ is set **then**
output ['forward', $M[m]$] at the inner interface
else
output ['delete'] at the inner interface

The number t' is defined as the computation steps σ makes at most and thus $\sigma \in \Sigma_{t'}$. We note that σ only generates random messages, encrypts messages and executes some set operations in the order of the number of queries. Thus, t' is efficient with respect to any security parameter k in a reasonable computational (as we assumed it) if the number of queries is efficient in k.

Towards a contradiction, we assume $\pi_1^A \pi_2^B(\longrightarrow \| \bullet\!\!=\!\!\bullet)$ does not construct a secure channel, and let thus **D** be a distinguisher making at most q_e queries at the A-interface, q_d queries at the E-interface and making at most t computation steps that can distinguish the constructed channel and the ideal, secure channel with the above defined simulator σ with advantage greater than ε:

$$\Delta^{\mathbf{D}}\left(\pi_1^A \pi_2^B(\longrightarrow \| \bullet\!\!=\!\!\bullet), \sigma^E(\bullet\!\!\longrightarrow\!\!\bullet)\right) > \varepsilon$$

We show that one can construct a converter **C** so that the bit-guessing variant of the composition **DC** according to Definition 4.8 results in an unforgeability game adversary $(\mathbf{DC})^+$ with advantage greater than ε.

Converter **C** works as follows:

on input m at the outer A-subinterface **do**
output ['encrypt', m] at the inner interface
get input ['encrypted', c] at inner interface
output c at the outer E-subinterface

on input c' at the outer E-subinterface **do**
output ['decrypt', c'] at the inner interface

if get input ['won', m', b] **then**
  if $b = 0$ **then**
    output m' at the outer B-subinterface
  else
    output $\perp$ at the outer B-subinterface
else
  get input ['decrypted', m']
  output m' at the outer B-subinterface

We see that the composition $(\mathbf{DC})^+$ makes at most q_e encryption and q_d decryption queries. The overhead in terms of computation steps imposed by $\mathbf{C}$ can be ignored as $\mathbf{C}$ only forwards queries and thus the composition makes at most t computation steps.

In the following, we let $\mathbf{S}^{EF}_{RCCA-0}(\pi)$ and $\mathbf{S}^{EF}_{RCCA-1}(\pi)$ be the two conditioned systems of the EF-RCCA game according to Definition 4.3, the first being the EF-RCCA game always encrypting the real message in the encryption queries, and the second being the analogous one always encrypting a random message.

Comparing the two systems $\mathbf{CS}^{EF}_{RCCA-0}(\pi)$ and $\pi^A_1\pi^B_2(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$, we see that the distribution of the outputs is always the same. Inputs at the A-interface are answered by the encryption of the input and inputs at the E-interface are answered by the message to which the input ciphertext decrypts. Thus, the two systems are equivalent,

$$\mathbf{CS}^{EF}_{RCCA-0}(\pi) \equiv \pi^A_1\pi^B_2(\longrightarrow \parallel \bullet\!\!=\!\!\bullet). \tag{4.11}$$

Comparing the two systems $\mathbf{CS}^{EF}_{RCCA-1}(\pi)$ and $\sigma^E(\bullet\!\!\longrightarrow\!\!\bullet)$, we can state the same result, namely that the output distribution of the two systems is identical and that the systems are equivalent. Queries at the A-interface are answered in both cases with the encryption of a random equal-length message. For queries at the E-interface, we distinguish between the case where the input ciphertext corresponds to a random message used in a previous encryption or to a different message. In the first case, the game returns the respective random message and converter $\mathbf{C}$ replaces the random message by the corresponding real message. In the channel system, the simulator decrypts the random message and tells the channel to forward the corresponding message. In this case, the output behavior is thus the same in both systems. In the latter case, where the encryption of a different message is input, the game returns this message along with the bit 1, and the converter $\mathbf{C}$ replaces this answer by a $\perp$-symbol. In the channel system, the same effect can be observed as the simulator tells the channel to delete the

message resulting in a $\perp$-symbol at the B-interface. Thus,

$$\mathbf{CS}^{EF}_{RCCA-1}(\pi) \equiv \sigma^E(\bullet\longrightarrow\bullet). \tag{4.12}$$

Using Lemma 4.9 and the equivalence relations (4.11) and 4.12), we get an advantage for $(\mathbf{DC})^+$ of,

$$\begin{aligned}
\Phi^{(\mathbf{DC})^+}(\mathbf{S}^{EF}_{RCCA}(\pi)) &= \Delta^{((\mathbf{DC})^+)^\dagger}(\mathbf{S}^{EF}_{RCCA-0}(\pi), \mathbf{S}^{EF}_{RCCA-0}(\pi)) \\
&= \Delta^{\mathbf{DC}}(\mathbf{S}^{EF}_{RCCA-0}(\pi), \mathbf{S}^{EF}_{RCCA-0}(\pi)) \\
&= \Delta^{\mathbf{D}}(\mathbf{CS}^{EF}_{RCCA-0}(\pi), \mathbf{CS}^{EF}_{RCCA-0}(\pi)) \\
&= \Delta^{\mathbf{D}}(\pi_1^A\pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \sigma^E(\bullet\longrightarrow\bullet)) > \varepsilon.
\end{aligned}$$

As this contradicts our assumption that π is $(\varepsilon, q_e, q_d, t)$-secure in the sense of EF-RCCA, we conclude that such a distinguisher $\mathbf{D}$ cannot exist and thus π does indeed construct a secure channel $\bullet\longrightarrow\bullet$ with error ε from the resource $(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$. The asymptotic statement follows directly. □

Theorem 4.28 (Secure channel $\bullet\longrightarrow\bullet$ $\Rightarrow$ EF-RCCA). *If a secrect-key encryption scheme family $\pi(k)$ constructs a secure channel $\bullet\longrightarrow\bullet$ from $(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$, then $\pi(k)$ is EF-RCCA secure.*

Especially if the secret-key encryption scheme π constructs a secure channel $\bullet\longrightarrow\bullet$ from $(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ with error ε for the distinguisher class $\mathcal{A}_{(q_A,q_E,t)}$ of all distinguishers making at most q_A, q_E queries at the respective interface and at most t computation steps, and for the converter class $\Sigma_{t''}$, then π is $(\ 2\varepsilon, q_A, q_E, t - t')$-secure in the sense of EF-RCCA where t' captures some efficiently implementable computation steps.

Proof. Let π be an encryption scheme constructing a secure channel from $(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ with error ε, formally

$$\exists\ \sigma \in \Sigma_{t''} : \Delta^{\mathcal{A}_{(q_A,q_E,t)}}(\pi_1^A\pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \sigma^E\ \bullet\longrightarrow\bullet) \leq \varepsilon.$$

Let $\mathbf{A}$ further be an EF-RCCA adversary with advantage $\Phi^{\mathbf{A}}(\mathbf{S}^{EF}_{RCCA}(\pi)) > 2\varepsilon$, making at most q_A encryption, q_E decryption queries and at most $t - t'$ computation steps.

Towards a contradiction, we show that we can construct a converter $\mathbf{C}$ so that the combined system $\mathbf{AC}$ results in a distinguisher with advantage greater than ε distinguishing the above two channels.

Converter $\mathbf{C}$ works as follows:

initialize

generate $b \in_R \{0,1\}$
initialize $S \leftarrow \emptyset$
initialize empty map $M : \mathcal{M} \rightarrow \mathcal{M}$

on input $X_i = [\text{'encrypt'}, m]$ at outer interface **do**
if $b = 0$ **then**
$m' \leftarrow m$
else
generate $m' \in_R \mathcal{M}$ with $|m'| = |m|$
$M[m'] \leftarrow m$
$S \leftarrow S \cup \{m'\}$
output m' at the inner A-subinterface
get input c at inner E-subinterface
output $Y_i = [\text{'encrypted'}, c]$ at outer interface

on input $X_i = [\text{'decrypt'}, c']$ at outer interface **do**
output c' at inner E-subinterface
get input m' at inner B-subinterface
if $m' = \perp$ **then**
output $Y_i = [\text{'decrypted'}, \perp]$ at outer interface
elseif $m' \notin S$ **then**
output $Y_i = [\text{'won'}, m', b]$ at outer interface
elseif $b = 1$ **then**
output $Y_i = [\text{'decrypted'}, M[m']]$ at outer interface
else
output $Y_i = [\text{'decrypted'}, m']$ at outer interface

on first input $X_i = [\text{'guess'}, d]$ at outer interface **do**
if $b = d$ **then**
output $W = 1$ as distinguisher guess
else
output $W = 0$ as distinguisher guess
output $Y_i = [\text{'guessed'}]$ at outer interface

Analyzing the composed system **AC**, we see that it makes at most q_A queries at the A-interface, q_E queries at the E-interface. We further define $t' = t'(q_e, q_d)$ to be the number of computation steps needed by **C** consisting of the generation of random messages in the order of q_e and maintaining the map M and the set S in the order of $q_e + q_d$. Thus **AC** makes at most $t - t' + t' = t$

Comparing the game $\mathbf{S}^{EF}_{RCCA}(\pi)$ and the composed system $\mathbf{C}\pi_1^A\pi_2^B(\longrightarrow \| \bullet\!\!-\!\!\!-\!\!\bullet)$, we see that both encryption and decryption queries are handled in the

same way, which results in an identical output distribution: Either the input message is encrypted and inserted into the set S, or a random message is encrypted and inserted into the set. The choice of the variant is based on a secret bit b that is chosen uniformly at random in either system. If, in a decryption query, the ciphertext is valid and does not decrypt to a message in S, this is considered a forgery and credited by outputting the secret bit b in both systems. Invalid ciphertexts are answered trivially in the same way, and for a decryption query with a ciphertext decrypting to a message in the set S, the corresponding input message (stored in map M in the random case) is returned in both systems. We conclude that the two systems are equivalent with respect to encryption and decryption queries. Comparing the behavior on an input of a guess query with the bit d, we note that the game is won if $d = b$ and not won otherwise, and in the composed system, the distinguisher output $W = 1$ if $d = b$ and $W = 0$ otherwise. We conclude that winning the game corresponds exactly to a distinguisher output $W = 1$ in the composed system and thus,

$$\mathsf{P}^{\mathbf{AC}\pi_1^A\pi_2^B(\longrightarrow\|\bullet\Longrightarrow\bullet)}(W = 1) = \Gamma^{\mathbf{A}}(\mathbf{S}^{EF}_{RCCA}(\pi)). \tag{4.13}$$

We further analyze the system $\mathbf{C}\ \sigma'^E(\bullet\longrightarrow\bullet)$ with an arbitrary simulator σ' by inspecting the dependency of the output on the bit b chosen by the converter at random during initialization.

1. 'encrypt'-query: The real message m is input into the channel in the case of $b = 0$, a random message m' otherwise, so the input depends on the bit b. But as the channel $\bullet\longrightarrow\bullet$ does per definition not leak any information about the input message to the simulator, except length of the message, and as we have the condition that $|m| = |m'|$ upon generation of m', the input of the simulator is independent of b and the output of the simulator is independent of b.
2. 'decrypt'-query: A simulator only has the possibilities to tell the channel to replay a message from the history—in this case the corresponding real message is output by the converter as the decryption result, independent of the bit b— or the channel is told to delete the message where the output is a $\perp$-symbol trivially independent of b. Since all inputs to the simulator are independent of b, we come to the conclusion that the output to any decryption query is independent of the bit b.

We conclude that with any simulator σ' (constrained to any bound t'' on the number of computation steps), any adversary $\mathbf{A}'$ sees equally distributed output—for both values of b—interacting with the system $\mathbf{C}\ \sigma'^E(\bullet\longrightarrow\bullet)$, thus the adversary must guess a secret bit chosen uniformly at random and guesses correctly with probability $\frac{1}{2}$. In particular, this holds for adversary $\mathbf{A}$,

$$\forall\ \sigma' : \Gamma^{\mathbf{A}}(\mathbf{C}\,\sigma'^E(\bullet\longrightarrow\bullet)) = \frac{1}{2}. \tag{4.14}$$

Combining (4.13) and (4.14), the advantage of **AC** can be bounded by

$$\begin{aligned}
\forall\, \sigma' : \Delta^{\mathbf{AC}}(\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \sigma'^E \bullet\!\!\longrightarrow\!\!\bullet) \\
&= \left| \mathsf{P}^{\mathbf{AC}\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet=\bullet)}(W = 1) - \mathsf{P}^{\mathbf{AC}\sigma'^E \bullet\longrightarrow\bullet}(W = 1) \right| \\
&= \left| \Gamma^{\mathbf{A}}(\mathbf{S}^{EF}_{RCCA}(\pi)) - \frac{1}{2} \right| \\
&= \frac{1}{2} \cdot \Phi^{\mathbf{A}}(\mathbf{S}^{EF}_{RCCA}(\pi)) > \varepsilon.
\end{aligned}$$

Since **AC** has advantage greater than ε for any simulator σ', the existence of such an adversary **A** contradicts our assumption that π constructs a secure channel $\bullet\!\!\longrightarrow\!\!\bullet$ with error ε from $(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$. Therefore, no such adversary **A** can exist and π must be $(2\varepsilon, q_A, q_E, t - t')$-secure in the sense of EF-RCCA. The asymptotic statement follows directly. □

4.5 A Pure Confidentiality Notion

In Section 4.2, the need for a general attack model that also account for arbitrary malleability characteristics of encryption schemes was motivated: The traditional attack models do not account for a message-history dependent malleability and allow trivial attacks for schemes allowing this type of malleability. Thus a general model is needed both for notions that capture the security of such schemes as well as for a notion capturing "pure" confidentiality. The definition of a pure confidentiality notion is introduced here along with the formalization of the new general attack model.

Using the idea of real-or-random based games, oracle queries are either answered by a normal encryption and decryption in the "real" case, or in the "random" case by simulating the encryption and decryption (this simulation can and will strongly depend on the specific scheme).

4.5.1 Capturing Confidentiality in a Game

The intuition behind the formalization is as follows: Confidential communication does not leak any information about sent messages (except for their length) and allows an arbitrary malleability. Ciphertexts thus must be indistinguishable from simulated ciphertexts that only depend on the length of the message to encrypt. This is formalized by a pair of functions f_1 and f_2 where f_1 handles the generation of ciphertexts and f_2 handles the simulation of decryption. Since simulated encryption and decryption both can be probabilistic and share information (e.g a key), the functions f_1 and f_2

have a shared state s that potentially includes randomness. To formalize the property that the encryption scheme is confidential (i.e. leaks no information about the encrypted messages except for possibly their length), the function f_1 is restricted to produce outputs that may depend on the length of the input messages, but these outputs must be independent of the values of these messages.

Definition 4.29 (Confidentiality game). *Let $\pi = (\pi_1, \pi_2)$ be a secret-key encryption scheme according to Definition 3.7, f_1 be an efficiently computable function taking as input state information s and a message m_i, outputting state information s' and a ciphertext c_i. Let f_2 be another efficiently computable function taking as argument state information $\tilde{s}$ and a ciphertext c_i', outputting state information $\tilde{s}'$ and a message $m_i' \in (\mathcal{M} \dot{\cup} \{\perp\})$. Ciphertexts output by the function f_1 must only depend on length of previous input messages, on previous ciphertexts output by f_1, and on previous ciphertexts input into f_2. Let S_0 be a set of initial state information and $\mathbf{S}_{CONF}(\pi, f_1, f_2, S_0)$ be a bit-guessing game according to Definition 4.3, called* confidentiality game*, with the following input/output behavior: $\mathbf{S}_{CONF}(\pi, f_1, f_2, S_0)$ has input variables $X_i \in [\{'encrypt'\} \times \mathcal{M}] \dot{\cup} [\{'decrypt'\} \times \mathcal{C}] \dot{\cup} [\{'guess'\} \times \{0,1\}]$ as well as the output variables $Y_i \in [\{'encrypted'\} \times \mathcal{C}] \dot{\cup} [\{'decrypted'\} \times (\mathcal{M} \dot{\cup} \{\perp\})] \dot{\cup} [\{'guessed'\}]$ and the MBO. $\mathbf{S}_{CONF}(\pi, f_1, f_2, S_0)$ works as follows:*

initialize
  generate the bit $b \in_R \{0,1\}$
  generate key K $\in_R \mathcal{K}$
  initialize state information $s \in_R S_0$

on input $X_i = [\text{'encrypt'}, m_i]$ **do**
  if $b = 0$ **then**
    output $Y_i = [\text{'encrypted'}, \pi_1(\text{K}, m_i)]$
  else
    $(s, c_i) \leftarrow f_1(s, m_i)$
    output $Y_i = [\text{'encrypted'}, c_i]$

on input $X_i = [\text{'decrypt'}, c_i']$ **do**
  if $b = 0$ **then**
    output $Y_i = [\text{'decrypted'}, \pi_2(\text{K}, c_i')]$
  else
    $(s, m_i') \leftarrow f_2(s, c_i')$
    output $Y_i = [\text{'decrypted'}, m_i']$

on first input $X_i = [\text{'guess'}, d]$ **do**
  $A_i \leftarrow A_{i-1} \vee (b = d)$
  output $Y_i = [\text{'guessed'}]$

Inputs of type $X_i = [\text{'encrypt'}, m]$ *are called* encryption queries *and those of type* $X_i = [\text{'decrypt'}, c']$ decryption queries.

Definition 4.30 (Confidentiality). *We say a secret-key encryption scheme* π *according to Definition 3.7 is* $(\varepsilon, q_e, q_d, t)$*-confidential if there exist functions* f_1 *and* f_2 *and a set* S_0 *according to the definition of the Confidentiality game, such that*

$$\Phi^{\mathcal{A}_{(q_e,q_d,t)}}(\mathbf{S}_{CONF}(\pi, f_1, f_2, S_0)) \leq \varepsilon,$$

with q_e *being the number of encryption queries and* q_d *denoting the number of decryption queries.*

We say that the secret-key encryption scheme family $\pi(k)$ *is confidential if, for all* k*,* $\pi(k)$ *is* $(\varepsilon, q_e, q_d, t)$*-secure in the sense of Confidentiality and* q_e, q_d *and* t *are efficient in* k *and* ε *is negligible in* k*.*

4.5.2 Equivalence to the Construction of a Confidential Channel

To show that the introduced game indeed captures confidentiality, it is shown that a scheme is secure in the sense of Confidentiality if and only if it constructs a confidential channel from an insecure channel and a shared secret key.

Lemma 4.31 (Confidentiality $\Rightarrow$ Confidential channel $\longrightarrow\!\!\bullet$). *If a secret-key encryption scheme family* $\pi(k)$ *is confidential, then* $\pi(k)$ *constructs a confidential channel* $\longrightarrow\!\!\bullet$ *from the resource* $(\longrightarrow \| \bullet\!\!=\!\!\bullet)$*.*

Especially if a secret-key encryption scheme $\pi = (\pi_1, \pi_2)$ *is* $(\varepsilon, q_e, q_d, t)$*-confidential, then the encryption scheme* π *constructs a confidential channel* $\longrightarrow\!\!\bullet$ *from the resource* $(\longrightarrow \| \bullet\!\!=\!\!\bullet)$ *with error* ε*, namely there exists a simulator* σ *in* Σ_0 *such that* $\Delta^{\mathcal{A}_{(q_e,q_d,t)}}\left(\pi_1^A \pi_2^B(\longrightarrow \| \bullet\!\!=\!\!\bullet), \sigma^E(\longrightarrow\!\!\bullet)\right) \leq \varepsilon$ *where* $\mathcal{A}_{(q_e,q_d,t)}$ *is the class of all distinguishers making at most* q_e *queries at the A-interface and at most* q_d *queries at the E-interface.*

Proof. Let π be a secret-key encryption scheme that is $(\varepsilon, q_e, q_d, t)$-confidential, namely there exist efficiently computable functions f_1 and f_2 and a set S_0 according to the game definition, such that $\Phi^{\mathcal{A}_{(q_e,q_d,t)}}(\mathbf{S}_{CONF}(\pi, f_1, f_2, S_0)) \leq \varepsilon$.

Let $\longrightarrow\!\!\bullet$ be the following confidential channel based on the functions f_1, f_2 and the set S_0 from above: Transformations that are input at the E-interface are specified by ciphertexts $c_i' \in \mathcal{C}$ and evaluated using the function f_2 on the state s. Additionally on inputs m_i at the A-interface, the state s is updated by

using the function f_1 on m_i. The state of the channel and thus the set of eligible transformations after i queries, $\mathcal{A}_i$, is specified by the ciphertext history output by f_1 and the ciphertext history input to f_2. Thus, on input $X_i = m_i$ at the A-interface, $\mathcal{A}_i$ is specified implicitly by $\mathcal{A}_{i-1}$ (consisting of the previous ciphertext history) and explicitly by a ciphertext c_i (that is output in round i by f_1). Note that this definition of the sets of eligible transformations is compatible with the definition since the sets only depend on the length of the messages, the previous sets and the chosen transformations. We summarize the characterization of $\longrightarrow\!\!\bullet$ as follows:

initialize
initialize state information $s \in_R S_0$

on input $X_i = m_i$ at A-interface **do**
$(s, c_i) \leftarrow f_1(s, m_i)$
output $|m_i|$ and c_i as specification of $\mathcal{A}_i$ at E-interface

on input $X_i = [\text{'modify'}, c_i']$ at E-interface **do**
$(s, m_i') \leftarrow f_2(s, c_i')$
output $Y_i = m_i'$ at B-interface

We further define the simulator σ as follows:

on input $X_i = (l_i, c_i)$ at inner interface **do**
output $Y_i = c_i$ at outer interface

on input $X_i = c_i'$ at outer interface **do**
output $[\text{'modify'}, c_i']$ at inner interface

Since σ only forwards messages, it makes no computation steps and thus $\sigma \in \Sigma_0$. The definition of the simulator σ is sound with respect to the channel $\longrightarrow\!\!\bullet$ since it only outputs ciphertexts that are in the set of eligible transformations defined above.

Let $\mathbf{D}$ be a distinguisher making at most q_e queries at the A-interface, at most q_d queries at the E-interface, making at most t computation steps and having a distinguishing advantage greater than ε for distinguishing the two channels $\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!\longrightarrow\!\!\bullet)$ and $\sigma^E(\longrightarrow\!\!\bullet)$:

$$\Delta^{\mathbf{D}}\left(\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!\longrightarrow\!\!\bullet), \sigma^E(\longrightarrow\!\!\bullet)\right) > \varepsilon$$

Let adversary $\mathbf{D}^+$ be the bit-guessing game adversary variant of distinguisher $\mathbf{D}$ from Definition 4.8. We now construct a converter $\mathbf{C}$ that when composed with adversary $\mathbf{D}^+$ results in a Confidentiality adversary $\mathbf{D}^+\mathbf{C}$.

Converter $\mathbf{C}$ works as follows:

on input m at A-interface **do**
output $['\text{encrypt}', m]$ at the inner interface
get input $['\text{encryption}', c]$ at inner interface
output c at the outer E-interface

on input c' at E-interface **do**
output $['\text{decrypt}', c']$ at the inner interface
get input $['\text{decrypted}', m']$ at inner interface
output m' at the outer B-interface

on input $['\text{guess}', b]$ at the outer interface **do**
forward the query and the response

As the converter $\mathbf{C}$ essentially just forwards queries, $\mathbf{D}^+\mathbf{C}$ makes the same number of queries and computation steps as $\mathbf{D}$.

As a converter that only forwards guess queries such as $\mathbf{C}$ can be connected to both game winner and distinguisher without affecting the "rerouting" of the guess query, the following equivalence holds,

$$(\mathbf{D}^+\mathbf{C})^\dagger \equiv (\mathbf{D}^+)^\dagger\mathbf{C} \equiv \mathbf{D}\mathbf{C}. \tag{4.15}$$

Let $\mathbf{S}_{CONF-0}(\pi, f_1, f_2, S_0)$ and $\mathbf{S}_{CONF-1}(\pi, f_1, f_2, S_0)$ be the two conditional games of the Confidentiality game as per Definition 4.3, the first being the Confidentiality game encrypting always the real message, and the second being the analogous one applying the functions f_1 and f_2 for the respective queries.

Comparing $\mathbf{CS}_{CONF-0}(\pi, f_1, f_2, S_0)$ and $\pi_1^A\pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$, we see that the output on queries at the A-interface is distributed identically as the input message is encrypted using a random key K. The output on queries to the E-interface is distributed identically, since the input ciphertext is decrypted using the same key K as in encryption. Thus the two systems are equivalent:

$$\mathbf{CS}_{CONF-0}(\pi, f_1, f_2, S_0) \equiv \pi_1^A\pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet). \tag{4.16}$$

Comparing the system $\mathbf{CS}_{CONF-1}(\pi, f_1, f_2, S_0)$ with $\sigma^E(\longrightarrow\!\!\bullet)$, we again see that the output on queries at the A-interface as well as the output on queries

at the E-interface is distributed exactly the same in both systems: The game as well as the channel maintain a state s and the transformations on that state are according to the same distribution in both systems. This guarantees that the input to the functions f_1 and f_2 is distributed identically, what in turn implies that the output of the functions is distributed accordingly in both systems. A query at the A-interface is answered by the output of the function f_1 in both systems. For a query at the E-interface, we see that in both systems, the function f_2 is invoked and the result is the output of the system. We conclude that all the output is distributed identically and thus, the two systems are equivalent,

$$\mathbf{CS}_{CONF-1}(\pi, f_1, f_2, S_0) \equiv \sigma^E(\longrightarrow\!\bullet). \tag{4.17}$$

Combining Lemma 4.9 and the equivalences (4.15), (4.16), and (4.17), we get an advantage for $\mathbf{D}^+\mathbf{C}$ in winning the game $\mathbf{S}_{CONF}(\pi, f_1, f_2, S_0)$ of

$$\begin{aligned}
&\Phi^{\mathbf{D}^+\mathbf{C}}(\mathbf{S}_{CONF}(\pi, f_1, f_2, S_0)) \\
&\quad = \Delta^{(\mathbf{D}^+\mathbf{C})^\dagger}(\mathbf{S}_{CONF-0}(\pi, f_1, f_2, S_0), \mathbf{S}_{CONF-1}(\pi, f_1, f_2, S_0)) \\
&\quad = \Delta^{\mathbf{DC}}(\mathbf{S}_{CONF-0}(\pi, f_1, f_2, S_0), \mathbf{S}_{CONF-1}(\pi, f_1, f_2, S_0)) \\
&\quad = \Delta^{\mathbf{D}}(\mathbf{CS}_{CONF-0}(\pi, f_1, f_2, S_0), \mathbf{CS}_{CONF-1}(\pi, f_1, f_2, S_0)) \\
&\quad = \Delta^{\mathbf{D}}(\pi_1^A \pi_2^B(\longrightarrow \| \bullet\!\longrightarrow\!\bullet), \sigma^E(\longrightarrow\!\bullet)) > \varepsilon.
\end{aligned}$$

As this contradicts our assumption that π is $(\varepsilon, q_e, q_d, t)$-confidential, we conclude that such a distinguisher $\mathbf{D}$ cannot exist and thus π does indeed, with error ε, construct a confidential channel $\longrightarrow\!\bullet$ from the resource $(\longrightarrow \| \bullet\!\longrightarrow\!\bullet)$. The asymptotic statement follows directly. □

For the implication in the other direction, a new type of formalizing discrete systems has to be introduced that is equivalent to the random system formalization.

So far, we characterized systems as random systems described by conditional probability distributions based on the past inputs and outputs of the system. In this proof, we need, however, a different formalization of systems, introduced by Maurer in [25] as an equivalent characterization of systems, called *random automata*. This characterization allows to describe a system as if it would have an internal state: A random automaton has input space $\mathcal{X}$ and output space $\mathcal{Y}$, a space of state information S (consisting of both the state and the internal randomness), and is defined by a sequence of functions $f_1, f_2, \ldots$ called *transition functions*, $f_i : \mathcal{X} \times S \to \mathcal{Y} \times S$. A transition is written as $(Y_i, s_i) = f_i(X_i, s_{i-1})$ where s_i is the state information in round i, and s_0 the initial state information with the initialized internal randomness.

This characterization is used to specify the confidential channel and the simulator in the proof of the following lemma, showing that an encryption

scheme that constructs a confidential channel is secure in the sense of Confidentiality.

Lemma 4.32 (Confidential channel $\longrightarrow\!\!\bullet \Rightarrow$ Confidentiality). *If a secret-key encryption scheme family $\pi(k)$ constructs a confidential channel $\longrightarrow\!\!\bullet$ from the resource $(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$, then $\pi(k)$ is confidential.*

Especially if the secret-key encryption scheme π constructs a confidential channel $\longrightarrow\!\!\bullet$ from $(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ with error ε for the distinguisher class $\mathcal{A}_{(q_A,q_E,t)}$ of all distinguishers making at most q_A, q_E queries at the respective interface and at most t computation steps and for the converter class $\Sigma_{t''}$, then π is $(\varepsilon, q_A, q_E, t)$-confidential.

Proof. Let π be a secret-key encryption scheme constructing a confidential channel $\overset{*}{\longrightarrow}\!\!\bullet$ from $(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ for the classes $\mathcal{A}_{(q_A,q_E,t)}$ and $\Sigma_{t''}$ with error ε, formally

$$\exists\, \sigma' \in \Sigma_{t''} : \Delta^{\mathcal{A}_{(q_A,q_E,t)}}(\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \sigma'^E(\overset{*}{\longrightarrow}\!\!\bullet)) \leq \varepsilon.$$

We construct a channel $\longrightarrow\!\!\bullet$ by extending the malleability of $\overset{*}{\longrightarrow}\!\!\bullet$ with the transformations corresponding to forwarding, replaying and deleting messages. The resulting channel apparently still gives the same guarantees since forwarding, replaying and deleting is always possible in a confidential channel. By adapting the simulator σ' such that forward, replay and delete "commands" are replaced by modify "commands" containing the corresponding transformations, resulting in the simulator σ, we get an equivalent channel construction $\sigma^E(\longrightarrow\!\!\bullet)$:

$$\Delta^{\mathcal{A}_{(q_A,q_E,t)}}(\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \sigma^E(\longrightarrow\!\!\bullet)) \leq \varepsilon.$$

Note that the adapted simulator still makes the same number of computation steps.

Recall the introduction of a random automaton from above where a system is characterized by a set of transition functions operating on the input and on a state (including randomness). We characterize in the following both $\longrightarrow\!\!\bullet$ and σ as random automata.

To specify $\longrightarrow\!\!\bullet$ as random automaton, recall the definition of a confidential channel and note that on input of a message m_i at the A-interface, $|m_i|$ and the set of eligible transformations is output at the E-interface. We denote by g_1 the transition function that handles inputs at the A-interface and works according to the random system specification. Recall further that on input α_i at the E-interface, a message m_i' is output at the B-interface. The corresponding transition function is denoted by g_4. Let T_0 be the set of initial state

information (fixed initial state plus possible randomness), and thus $\longrightarrow\!\!\bullet$ is characterized as follows:

initialize
initialize state information $t \in_R T_0$

on input $X_i = m_i$ at the A-interface **do**
$(t, |m_i|, \mathcal{A}_i) \leftarrow g_1(t, m_i)$
output $|m_i|$ and $\mathcal{A}_i$ at the E-interface

on input $X_i = \alpha_i$ at the E-interface **do**
$(t, m_i') \leftarrow g_4(t, \alpha_i)$
output m_i' at the B-interface

Note that the output of g_1 referring to the set of eligible transformations is defined to only depend on the message lengths, the previous sets of eligible transformations and the chosen transformations.

The simulator σ can analogously be specified as a random automaton where S_0' is the set of initial state information, g_2 the efficiently computable transition function for inputs at the inner interface, and g_3 is the efficiently computable transition function responsible for inputs at the outer interface:

initialize
initialize state information $s' \in_R S_0'$

on input $X_i = (l_i, \mathcal{A}_i)$ at inner interface **do**
$(s', c_i) \leftarrow g_2(s', l_i, \mathcal{A}_i)$
output c_i at outer interface

on input $X_i = c_i'$ at outer interface **do**
$(s', \alpha_i) \leftarrow g_3(s', c_i')$
output ['modify', α_i] at inner interface

Note that any "good" simulator must have this behavior. A simulator that does not output a c_i at the outer interface on an input at the inner interface or that does not provide an output at the inner interface on an input at the outer interface can be transformed into a "good" simulator without decreasing the advantage of a distinguisher: Since any "real" channel $\pi_1^A \pi_2^B (\longrightarrow \| \bullet\!\!\longrightarrow)$ has the behavior that on an input at the A-interface, an output at the E-interface is produced, and similarly for an input at the E-interface, an output

at the B-interface is produced, a distinguisher can recognize the "misbehavior" of simulator described above in the system $\sigma^E(\longrightarrow\!\bullet)$ and thus possibly has a larger distinguishing advantage compared to the system with the "good" simulator.

The specification of the channel and the simulator as random automata above allowed a separation into functions for the respective inputs. Such a separation is needed to construct the functions f_1 and f_2 for the confidentiality game. For this, we define the set of initial state information S_0 as product set of the above two sets of initial state information $T_0 \times S_0'$. The functions f_1 and f_2 are constructed as follows:

function $f_1((t, s'), m)$:
$(t, |m|, \mathcal{A}_{el}) \leftarrow g_1(t, m)$
$(s', c) \leftarrow g_2(s', |m|, \mathcal{A}_{el})$
return $((t, s'), c)$

function $f_2((t, s'), c')$:
$(s', \alpha) \leftarrow g_3(s', c')$
$(t, m') \leftarrow g_2(t, \alpha)$
return $((t, s'), m')$

Note that any c output by f_1 does only depend on the message lengths (input $|m|$ of f_1), the previous output ciphertexts of f_1 and the ciphertexts previously input to f_2. Since the ciphertext c is produced by the function g_2 and since the only other external input that c can depend on is $\mathcal{A}_{el}$ (defined to depend only on the message lengths, the previous sets of eligible transformations and the chosen transformations), the function f_1 does indeed satisfy this property. Thus the functions f_1, f_2 and the set S_0 are compatible with the plaintext uncertainty game.

Let $\mathbf{A}$ be a Confidentiality adversary for the game (with parameters π, f_1, f_2 and S_0) with advantage $\Phi^{\mathbf{A}}(\mathbf{S}_{CONF}(\pi, f_1, f_2, S_0)) > \varepsilon$, making at most q_A encryption, q_E decryption queries and at most t computation steps.

Towards a contradiction, we show that we can use $\mathbf{A}$ to construct a distinguisher for $\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\longleftarrow\!\bullet)$ and $\sigma^E(\longrightarrow\!\bullet)$ with advantage greater than ε. For this, let $\mathbf{C}$ be the converter that just forwards encryption queries to the A-subinterface of the inner interface and decryption queries to the E-subinterface of the inner interface.

By a similar argument as in the proof of Lemma 4.31, we get the equivalence:

$$\mathbf{S}_{CONF-0}(\pi, f_1, f_2, S_0) \equiv \mathbf{C}\, \pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\longleftarrow\!\bullet).$$

For the comparison of the systems $\mathbf{S}_{CONF-1}(\pi, f_1, f_2, S_0)$ and $\mathbf{C}\, \sigma^E(\longrightarrow\!\bullet)$, we note that f_1 and f_2 are constructed exactly such that the two systems are equivalent: First of all, we note that the initial state informations are initialized accordingly in both systems. Encryption queries are in both systems

handled by applying g_1 on the state t and on the input message to get the message length and the set of eligible transformations. g_2 is further applied on the state s', the message length and the set of eligible transformations to get a ciphertext c that is output. Decryption queries are handled accordingly in both systems as f_2 is built exactly such that the game is equivalent to the converted channel construction for this kind of queries. We conclude that the two systems are equivalent,

$$\mathbf{S}_{CONF-1}(\pi, f_1, f_2, S_0) \equiv \mathbf{C}\,\sigma^E(\longrightarrow\!\bullet).$$

Using these two equivalences, Lemma 4.9 and (4.1), we get

$$\begin{aligned}
&\Delta^{\mathbf{A}^\dagger\mathbf{C}}\left(\pi_1^A\pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \sigma^E(\longrightarrow\!\bullet)\right)\\
&\quad= \Delta^{\mathbf{A}^\dagger}\left(\mathbf{C}\pi_1^A\pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \mathbf{C}\sigma^E(\longrightarrow\!\bullet)\right)\\
&\quad= \Delta^{\mathbf{A}^\dagger}\left(\mathbf{S}_{CONF-0}(\pi, f_1, f_2, S_0), \mathbf{S}_{CONF-1}(\pi, f_1, f_2, S_0)\right)\\
&\quad= \Phi^{(\mathbf{A}^\dagger)^+}(\mathbf{S}_{CONF}(\pi, f_1, f_2, S_0))\\
&\quad= \Phi^{\mathbf{A}}(\mathbf{S}_{CONF}(\pi, f_1, f_2, S_0)) > \varepsilon.
\end{aligned}$$

As this contradicts our assumption that π constructs a confidential channel $\longrightarrow\!\bullet$ with error ε from $(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$, we conclude that such an adversary $\mathbf{A}$ cannot exist and π must be $(\varepsilon, q_A, q_E, t)$-confidential. The asymptotic statement follows directly. □

4.6 Plaintext Uncertainty

The notion of plaintext-uncertainty is introduced in [19] along with several other notions capturing the integrity properties of encryption schemes. Plaintext-uncertainty aims at capturing an adversary's inability to create a ciphertext forgery to which he "knows" the result of decryption. Knowing the result of decryption refers to guessing the plaintext (as whole and not only a part of it, e.g. the last bit). In the formalization given in [19], several encryption queries can be submitted to an oracle before one single challenge ciphertext is produced. The game is won if the adversary guesses the decryption of the challenge ciphertext with "substantial" probability.

4.6.1 Formalization of the Game

Analogously to Section 4.4, the game is extended by a part covering the confidentiality of the encryption scheme to capture the essential property

of an encryption scheme. The added "real-or-random" oracle can only be queried once, similarly to the restriction to the number of challenge queries.

The formalization of the game is almost identical to the one provided for Confidentiality in the previous section. The difference is that in the plaintext-uncertainty (PU) version of the game, only one query per type ('encrypt' and 'decrypt') are allowed. Additionally, the adversary's "knowledge" of the decryption outcome is formalized by a plaintext that is queried along with the challenge ciphertext. A correct guess of the outcome is credited by outputting the secret bit b of the confidentiality game.

Definition 4.33 (Plaintext uncertainty game). *Let $\pi = (\pi_1, \pi_2)$ be a secret-key encryption scheme, let the functions f_1 and f_2 and the set S_0 be defined according to Definition 4.29 of the confidentiality game. And let $\mathbf{S}_{PU}(\pi, f_1, f_2, S_0)$ be a bit-guessing game, called* plaintext uncertainty game*, with the following input/output behavior: $\mathbf{S}_{PU}(\pi, f_1, f_2, S_0)$ has input variables $X_i \in [\{'encrypt'\} \times \mathcal{M}] \dot\cup [\{'decrypt'\} \times \mathcal{C} \times (\mathcal{M} \dot\cup \{\perp\})] \dot\cup [\{'guess'\} \times \{0,1\}]$ as well as the output variables $Y_i \in [\{'encrypted'\} \times \mathcal{C}] \dot\cup [\{'decrypted'\} \times (\mathcal{M} \dot\cup \{\perp\})] \dot\cup [\{'won'\} \times \{0,1\}] \dot\cup [\{'guessed'\}]$ and the MBO. $\mathbf{S}_{PU}(\pi, f_1, f_2, S_0)$ works as follows:*

initialize
  generate the bit $b \in_R \{0,1\}$
  generate key $\mathrm{K} \in_R \mathcal{K}$
  initialize $m_e \leftarrow \perp$
  initialize state information $s \in_R S_0$

on first input $X_i = ['\text{encrypt}', m]$ **do**
  $m_e \leftarrow m$
  if $b = 0$ **then**
    output $Y_i = ['\text{encrypted}', \pi_1(\mathrm{K}, m)]$
  else
    $(s, c) \leftarrow f_1(s, m)$
    output $Y_i = ['\text{encrypted}', c]$

on first input $X_i = ['\text{decrypt}', c, m_{guess}]$ **do**
  if $b = 0$ **then**
    $m' \leftarrow \pi_2(\mathrm{K}, c')$
  else
    $(s, m') \leftarrow f_2(s, c')$
  if $m' = \perp$ **then**
    output $Y_i = ['\text{decrypted}', \perp]$
  elseif $m' \neq m_e$ and $m' = m_{guess}$ **then**
    output $Y_i = ['\text{won}', b]$
  else
    output $Y_i = ['\text{decrypted}', m']$

on first input $X_i = [\text{'guess'}, d]$ **do**
$A_i \leftarrow A_{i-1} \vee (b = d)$
output $Y_i = [\text{'guessed'}]$

Definition 4.34 (PU-Confidentiality). *We say a secret-key encryption scheme π according to Definition 3.7 is (ε, t)-secure in the sense of PU-Confidentiality if there exist functions f_1 and f_2 and a set S_0 according to the definition of the Plaintext uncertainty game, such that*

$$\Phi^{\mathcal{A}_t}(\mathbf{S}_{PU}(\pi, f_1, f_2, S_0)) \leq \varepsilon.$$

We say that the secret-key encryption scheme family $\pi(k)$ is PU-confidential if, for all k, $\pi(k)$ is (ε, t)-secure in the sense of PU-Confidentiality and t is efficient in k and ε is negligible in k.

4.6.2 Capturing the Malleability

The analog of a game that allows only one encryption and decryption query in the constructive channel-based model is a single-use channel. Thus the translation of the notion of PU-Confidentiality into a channel-based analog must result in a single-use confidential channel with a certain type of malleability. Note that the malleability in a single-use confidential channel is specified by transformations $F : (\mathcal{M} \cup \{\perp\}) \rightarrow (\mathcal{M} \cup \{\perp\})$ since the "history" for the first input at the E-interface can only consist of no message at all (represented by the symbol $\perp$) or the message that was input in the preceding query at the A-interface. The reason why the $\perp$-symbol is included in the range of the transformation is to allow deleting transformations as part of the malleability, simplifying the following proofs.

Impossibility of Information-theoretical Definition A first idea to capture the malleability of a scheme that is secure according to Definition 4.34 is to allow transformations that have at least some min-entropy. This ensures that the probability of every single result of such a transformation is very unlikely, and thus no adversary can guess the outcome of the transformation and win the game defined above with substantial probability. The problem however with such an information-theoretical definition of the transformation-characteristic is that a transformation might not have *information-theoretical min-entropy*, but the outcome of the transformation might still be *computationally unpredictable* and the game cannot be won.

The following counter-example shows that one indeed has to define the allowed set of transformations in a computational sense by some notion of

unpredictability: Let $\pi = (\pi_1, \pi_2)$ be an encryption scheme that is secure in the sense of PU-Confidentiality according to Definition 4.34 and let f be an efficiently computable one-way permutation. We construct the encryption scheme $\pi' = (\pi_1', \pi_2')$ from the π as follows: We expand the original key K by an independent second key K_2 resulting in a key $K\|K_2$. The encryption converter π_1' uses the original encryption converter π_1, prepends the bit 0 and appends $f(K_2)$. Decryption works as in the original scheme if the prepended bit is 0. In the case of a prepended 1, π_2' outputs the second key K_2.

The new scheme π' is still secure in the sense of PU-Confidentiality, otherwise one could break the original scheme or the one-way permutation f. Considering the type of channel π' constructs, note that the malleability of a corresponding confidential channel must allow the output of the key K_2 on an input of the form $1\|c$ at the E-interface, what refers to a transformation without any min-entropy (neither information-theoretical or computational HILL-type according to [4]) since the output can be checked by the already known $f(K_2)$.

Computational Definition The second idea, which is more promising but less elegant, is to define transformations of the channel to be unpredictable in a computational sense. To formalize this, a game, called *prediction game*, is defined.

Definition 4.35. *Let $\longrightarrow\!\bullet$ be a single-use confidential channel, and let $\mathbf{S}_{pred}(\longrightarrow\!\bullet)$ be a game according to Definition 4.1 called* prediction game. *$\mathbf{S}_{pred}(\longrightarrow\!\bullet)$ works as follows:*

initialize
  initialize $\mathcal{A} \leftarrow \mathcal{A}_0$
on first input m at A-interface **do**
  input m at A-interface of $\longrightarrow\!\bullet$ to get $(l, \mathcal{A}_1)$
  $\mathcal{A} \leftarrow \mathcal{A}_1$
  output $\mathcal{A}_1$ at E-interface

on first input $(\alpha, m_{guess}), \alpha \in \mathcal{A}$ at E-interface **do**
  input α at E-interface of $\longrightarrow\!\bullet$ to get m'
  if $m_{guess} = m' \wedge m' \neq \perp \wedge m' \neq m$ **then**
    $A_i \leftarrow 1$

A prediction adversary can submit an (optional) query to the A-interface and must produce a description α from the set of eligible transformations of the channel and a guess message m_{guess} to win the game. The prediction

adversary is successful if his guess is not trivial ("forwarding" or "deleting") and if it matches the outcome of the transformation F_α.

A single-use confidential channel is called *δ-uncertain against t-predictors* if every prediction adversary bounded to at most t computation steps has at most a game winning probability of δ for the prediction game.

Definition 4.36. *A single-use confidential channel $\longrightarrow\!\!\bullet$ is called* δ-uncertain against t-predictors *if*

$$\Gamma^{\mathcal{A}_t}(\mathbf{S}_{pred}(\longrightarrow\!\!\bullet)) \leq \delta.$$

An encryption scheme is δ-uncertain against t-predictors *if it transforms a single-use, insecure channel $\longrightarrow$ into a single-use, δ-uncertain confidential channel against t-predictors. A single-use, δ-uncertain confidential channel against t-predictors is called* single-use, plaintext-uncertain confidential channel $\xrightarrow{PU}\!\!\bullet$ with respect to the security parameter k, *if δ is negligible in k and t is efficient in k.*

In the following, a confidential channel $\xrightarrow{f_1,f_2,S_0}\!\!\bullet$ is defined based on functions f_1, f_2 and the set S_0 according to the PU-game definition that is later used to show the equivalence of the game-based the security definition and the construction of a plaintext-uncertain confidential channel.

Definition 4.37. *Let the functions f_1, f_2 and the set S_0 be defined according to the plaintext uncertainty game definition, and let $S \times C$ be the domain of the function f_2. Let $\xrightarrow{f_1,f_2,S_0}\!\!\bullet$ be the following single-use confidential channel based on the functions f_1, f_2 and the set S_0 from above where transformations are defined as follows: A transformation that is input at the E-interface is specified by a value c' from C and evaluated using the function f_2 on the state s. Additionally on inputs m at the A-interface, the state s is updated by using the function f_1 on m. The state of the channel and thus the set of eligible transformations $\mathcal{A}_1$ after such a query at the A-interface is specified by the value output by f_1. The initial set of eligible transformations is implicitly given by the set C. We summarize the characterization of $\xrightarrow{f_1,f_2,S_0}\!\!\bullet$ as follows:*

initialize
initialize state information $s \in_R S_0$

on first input m at A-interface **do**
$(s, c) \leftarrow f_1(s, m)$
output $|m|$ and c as specification of $\mathcal{A}_1$ at E-interface

on input ['modify', c'] at E-interface **do**
$(s, m') \leftarrow f_2(s, c')$
output $Y_i = m'$ at B-interface

4.6.3 Equivalence Results

Using the definition from the previous section and the same proof technique as in Section 4.5.2, the following two theorems show that an encryption scheme family is PU-confidential if, and only if, it constructs a single-use, plaintext-uncertain confidential channel from an insecure channel and a key.

Theorem 4.38. *If a secret-key encryption scheme family $\pi(k)$ is PU-confidential, then $\pi(k)$ constructs a single-use, plaintext-uncertain confidential channel $\overset{PU}{\longrightarrow\!\bullet}$ from the resource $(\longrightarrow \parallel \bullet\!\Longrightarrow\!\bullet)$.*

Especially if an encryption scheme $\pi = (\pi_1, \pi_2)$ is (ε, t)-secure in the sense of PU-Confidentiality, then π constructs a single-use, 2ε-uncertain confidential channel $\longrightarrow\!\bullet$ against t-predictors with error ε from the resource $(\longrightarrow \parallel \bullet\!\Longrightarrow\!\bullet)$, namely there exists a simulator $\sigma \in \Sigma_0$ such that $\Delta^{\mathcal{A}_t}\left(\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\Longrightarrow\!\bullet), \sigma^E(\longrightarrow\!\bullet)\right) \leq \varepsilon$ where $\longrightarrow\!\bullet$ is a single-use, 2ε-uncertain confidential channel against t-predictors.

Proof. Let π be an encryption scheme that is (ε, t)-secure in the sense of PU-Confidentiality, i.e. there exist efficiently computable functions f_1 and f_2 and a set S_0 according to the game definition, such that $\Phi^{\mathcal{A}_t}(\mathbf{S}_{\mathbf{PU}}(\pi, f_1, f_2, S_0)) \leq \varepsilon$.

Let us further consider the confidential channel $\longrightarrow\!\bullet$ according to Definition 4.37 that is parametrized by the functions f_1, f_2 and the set S_0 from the plaintext uncertainty game, claiming that $\longrightarrow\!\bullet$ is 2ε-uncertain against t-predictors.

We prove the claim by contradiction, assuming that $\longrightarrow\!\bullet$ is not a 2ε-uncertain confidential channel against t-predictors: Let $\mathbf{P}$ be a prediction adversary making at most t computation steps with game winning probability

$$\Gamma^{\mathbf{P}}(\mathbf{S}_{pred}(\longrightarrow\!\bullet)) > 2\varepsilon.$$

We show that one can construct a converter $\mathbf{C}_1$, such that the composition $\mathbf{PC}_1$ results in a game adversary that wins the plaintext uncertainty game with advantage greater than ε. Converter $\mathbf{C}_1$ works as follows:

on first input m at A-interface **do**
    output ['encrypt', m] at the inner interface
    get input ['encryption', c] at inner interface
    output c at the outer E-interface

on first input (c', m_{guess}) at E-interface **do**
    output ['decrypt', c', m_{guess}] at the inner interface
    if get input ['won', 1] **then**
        output ['guess', 1] at the inner interface

else
$b' \in_R \{0,1\}$
output $['\text{guess}', b']$ at the inner interface

We note that $\mathbf{PC}_1$ is bound to make at most t computation steps since the operations of $\mathbf{C}_1$ (forwarding of messages and flipping a bit) can be ignored.

Analyzing $\mathbf{PC}_1$ with respect to the two conditional games $\mathbf{S}_{PU-0}(\pi, f_1, f_2, S_0)$ and $\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)$, we note that $\mathbf{PC}_1$ does not have any advantage interacting with $\mathbf{S}_{PU-0}(\pi, f_1, f_2, S_0)$ since every possible output of the game results in a guess bit that is chosen uniformly at random. We conclude that $\mathbf{PC}_1$ has winning probability $\frac{1}{2}$,

$$\Gamma^{\mathbf{PC}_1}(\mathbf{S}_{PU-0}(\pi, f_1, f_2, S_0)) = \frac{1}{2}.$$

For the winning probability of $\mathbf{PC}_1$ interacting with $\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)$, let $\mathcal{W}$ denote the event that $\mathbf{PC}_1$ sees an input of the form $['\text{won}', 1]$ and thus wins the game with probability 1. We can thus write the winning probability as

$$\begin{aligned}
&\Gamma^{\mathbf{PC}_1}(\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)) \\
&\quad = \mathsf{P}^{\mathbf{PC}_1\mathbf{S}_{PU-1}(\pi,f_1,f_2,S_0)}(\mathcal{W}) \cdot 1 + (1 - \mathsf{P}^{\mathbf{PC}_1\mathbf{S}_{PU-1}(\pi,f_1,f_2,S_0)}(\mathcal{W})) \cdot \frac{1}{2} \\
&\quad = \frac{1}{2} \cdot \mathsf{P}^{\mathbf{PC}_1\mathbf{S}_{PU-1}(\pi,f_1,f_2,S_0)}(\mathcal{W}) + \frac{1}{2}.
\end{aligned}$$

Using similar arguments as in the proof of Lemma 4.31, the prediction game and the plaintext uncertainty game behave identically on an input at the A-interface and on an encryption query respectively, and the "decryption part" is handled accordingly. We conclude that the winning condition in the prediction game corresponds to the case where $\mathcal{W}$ occurs in the plaintext uncertainty game. Due to the assumption made about $\mathbf{P}$, the adversary $\mathbf{PC}_1$ predicts the decryption result of the plaintext uncertainty game (i.e. the event $\mathcal{W}$ occurs) with probability greater than 2ε. Thus we get

$$\Gamma^{\mathbf{PC}_1}(\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)) > \varepsilon + \frac{1}{2}.$$

Using Corollary 4.6, the advantage of $\mathbf{PC}_1$ is thus

$$\begin{aligned}
&\Phi^{\mathbf{PC}_1}(\mathbf{S}_{PU}(\pi, f_1, f_2, S_0)) \\
&\quad = \left| \Gamma^{\mathbf{PC}_1}(\mathbf{S}_{PU-0}(\pi, f_1, f_2, S_0)) + \Gamma^{\mathbf{PC}_1}(\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)) - 1 \right| \\
&\quad = \left| \Gamma^{\mathbf{PC}_1}(\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)) - \frac{1}{2} \right|,
\end{aligned}$$

and since $\Gamma^{\mathbf{PC}_1}(\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)) > \frac{1}{2}$, we get

$$\Phi^{\mathbf{PC}_1}(\mathbf{S}_{PU}(\pi, f_1, f_2, S_0)) = \Gamma^{\mathbf{PC}_1}(\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)) - \frac{1}{2} > \varepsilon.$$

As this contradicts our assumption that π is (ε, t)-secure in the sense of PU-Confidentiality, such a predictor $\mathbf{P}$ cannot exist and thus the channel $\longrightarrow\!\bullet$ is indeed 2ε-uncertain against t-predictors.

Now we define an appropriate simulator σ as follows:

on first input l and c at inner interface **do**
    output c at outer interface

on first input c' at outer interface **do**
    output $['\text{modify}', c']$ at inner interface

We note that σ only forwards queries and is in Σ_0.

Let $\mathbf{D}$ be a distinguisher making at most t computation steps and having a distinguishing advantage greater than ε for distinguishing the two channels $\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ and $\sigma^E(\longrightarrow\!\bullet)$:

$$\Delta^{\mathbf{D}}\left(\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \sigma^E(\longrightarrow\!\bullet)\right) > \varepsilon \tag{4.18}$$

Let adversary $\mathbf{D}^+$ be the bit-guessing game adversary variant of distinguisher $\mathbf{D}$ from Definition 4.8. We now show that one can construct a simple converter $\mathbf{C}_2$ that when composed with adversary $\mathbf{D}^+$ results in a Plaintext-uncertainty-game adversary $\mathbf{D}^+\mathbf{C}_2$.

Converter $\mathbf{C}_2$ works as follows:

on first input m at A-interface **do**
    output $['\text{encrypt}', m]$ at the inner interface
    get input $['\text{encryption}', c]$ at inner interface
    output c at the outer E-interface

on first input c' at E-interface **do**
    output $['\text{decrypt}', c', \perp]$ at the inner interface
    get input $['\text{decrypted}', m']$ at inner interface
    output m' at the outer B-interface

on first input $['\text{guess}', b]$ at the outer interface **do**
    forward the query and the response

As the converter $\mathbf{C}_2$ essentially just forwards queries, $\mathbf{D}^+\mathbf{C}_2$ trivially makes the same number of queries and computation steps as $\mathbf{D}$, namely at most t computation steps.

And as the converter $\mathbf{C}_2$ always inputs a $\perp$-symbol as its guess, the game will never provide any output of the form $['won', b]$. The analysis falls down to the case of breaking confidentiality, and by using similar arguments as in the proof of Lemma 4.31, we get the two equivalences

$$\mathbf{C}_2\mathbf{S}_{PU-0}(\pi, f_1, f_2, S_0) \equiv \pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet),$$

and

$$\mathbf{C}_2\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0) \equiv \sigma^E(\longrightarrow\!\!\bullet).$$

Similarly, the advantage of $\mathbf{D}^+\mathbf{C}_2$ is therefore

$$\begin{aligned}
&\Phi^{\mathbf{D}^+\mathbf{C}_2}(\mathbf{S}_{PU}(\pi, f_1, f_2, S_0)) \\
&\quad = \Delta^{(\mathbf{D}^+\mathbf{C}_2)^\dagger}(\mathbf{S}_{PU-0}(\pi, f_1, f_2, S_0), \mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)) \\
&\quad = \Delta^{\mathbf{D}\mathbf{C}_2}(\mathbf{S}_{PU-0}(\pi, f_1, f_2, S_0), \mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)) \\
&\quad = \Delta^{\mathbf{D}}(\mathbf{C}_2\mathbf{S}_{PU-0}(\pi, f_1, f_2, S_0), \mathbf{C}_2\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)) \\
&\quad = \Delta^{\mathbf{D}}\left(\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \sigma^E(\longrightarrow\!\!\bullet)\right) > \varepsilon.
\end{aligned}$$

As this contradicts our assumption that π is (ε, t)-secure in the sense of PU-Confidentiality, we conclude that such a distinguisher $\mathbf{D}$ cannot exist and thus π does indeed, with error ε, construct a 2ε-uncertain confidential channel $\longrightarrow\!\!\bullet$ from the resource $(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$. The asymptotic statement follows directly. □

Theorem 4.39. *If a secret-key encryption scheme family $\pi(k)$ constructs a plaintext-uncertain confidential channel $\overset{PU}{\longrightarrow}\!\!\bullet$, then $\pi(k)$ is PU-confidential.*

Especially if the secret-key encryption scheme π constructs a δ-uncertain confidential channel $\longrightarrow\!\!\bullet$ against t_1-predictors with error ε from $(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ for the distinguisher class $\mathcal{A}_{t_2}$ and the converter class $\Sigma_{t_1-t_2}$, then π is $(\varepsilon + 2\delta, t_2)$-secure in the sense of PU-Confidentiality.

Proof. Let π be an encryption scheme constructing with error ε a δ-uncertain confidential channel $\longrightarrow\!\!\bullet$ against t_1-predictors from $(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$, formally

$$\exists\, \sigma \in \Sigma_{t_1-t_2} : \Delta^{\mathcal{A}_{t_2}}(\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!=\!\!\bullet), \sigma^E(\longrightarrow\!\!\bullet)) \leq \varepsilon.$$

Using the same formalization of a channel and the same arguments as in the proof of Lemma 4.32, the channel $\longrightarrow\!\!\bullet$ can be described as a random

automaton with initial state space T_0, transition function g_1 for an input at the A-interface and transition function g_4 for an input at the E-interface. The simulator σ can be described analogously by S_0' and transition functions g_2 and g_3 for the respective input at the inner and outer interface.

Let f_1, f_2 and S_0 further be defined in the same way as in the proof of Lemma 4.32: $S_0 := T \times S_0$, f_1 and f_2 work as follows:

function $f_1((t, s'), m)$:
$(t, |m|, \mathcal{A}_{el}) \leftarrow g_1(t, m)$
$(s', c) \leftarrow g_2(s', |m|, \mathcal{A}_{el})$
return $((t, s'), c)$

function $f_2((t, s'), c')$:
$(s', \alpha) \leftarrow g_3(s', c')$
$(t, m') \leftarrow g_2(t, \alpha)$
return $((t, s'), m')$

Note that the functions f_1, f_2 and the set S_0 are compatible with the plaintext uncertainty game.

Let $\mathbf{A}$ be an adversary for the plaintext-uncertainty game with advantage $\Phi^{\mathbf{A}}(\mathbf{S}_{PU}(\pi, f_1, f_2, S_0)) > \varepsilon + 2\delta$, making at most t_2 computation steps.

Towards a contradiction we show that we can construct a converter $\mathbf{C}$ such that the combined system $\mathbf{A}^{\dagger}\mathbf{C}$ results in a distinguisher with advantage greater than ε distinguishing the above two channels.

Converter $\mathbf{C}$ works as follows:

on first input ['encrypt', m] at the outer interface **do**
output m at the inner A-subinterface
get input c at the inner E-subinterface
output ['encryption', c] at the outer interface

on first input ['decrypt', c', m_{guess}] at outer interface **do**
output c' at the inner E-subinterface
get input m' at inner B-subinterface
if $m' \neq \perp$ and $m' = m_{guess}$ **then**
output ['won', 0] at the outer interface
else
output ['decrypted', m'] at the outer interface

Since $\mathbf{C}$ essentially just forwards queries, $\mathbf{A}^{\dagger}\mathbf{C}$ makes at most t_2 computation steps.

Comparing the two systems $\mathbf{S}_{PU-0}(\pi, f_1, f_2, S_0)$ and $\mathbf{C}\,\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!-\!\!\!-\!\!\bullet)$, we can argue similarly as in the proof of Lemma 4.31 that the two systems

are equivalent if no decryption outcome is guessed correctly. In the case of a correct guess, both systems output ['won', 0] and thus are equivalent also in this case. Thus,

$$\mathbf{S}_{PU-0}(\pi, f_1, f_2, S_0) \equiv \mathbf{C}\,\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!\longleftarrow\!\!\bullet). \tag{4.19}$$

For the comparison of the systems $\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)$ and $\mathbf{C}\,\sigma^E(\longrightarrow\!\!\bullet)$, we define the MES $\mathcal{B} = B_0, B_1, \ldots$ where the condition B_i is satisfied if no output $Y_j, j \leq i$ was of the form ['won', b], $b \in \{0, 1\}$. Clearly, the two systems in consideration behave exactly the same way as long as this condition is satisfied: Encrypt queries are simulated accordingly using the functions g_1 and g_2, and decryption queries are always answered by a 'decrypted' output containing the simulated decryption using the functions g_3 and g_4. We state this fact by the following conditional equivalence,

$$\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)|\mathcal{B} \equiv (\mathbf{C}\,\sigma^E(\longrightarrow\!\!\bullet))|\mathcal{B}. \tag{4.20}$$

We use Lemma 4.9 and the triangle inequality to get

$$\begin{aligned}\Phi^{\mathbf{A}}(\mathbf{S}_{PU}(\pi, f_1, f_2, S_0) &= \Delta^{\mathbf{A}^\dagger}(\mathbf{S}_{PU-0}(\pi, f_1, f_2, S_0), \mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)) \\ &\leq \Delta^{\mathbf{A}^\dagger}(\mathbf{S}_{PU-0}(\pi, f_1, f_2, S_0), \mathbf{C}\,\sigma^E(\longrightarrow\!\!\bullet)) \\ &\quad + \Delta^{\mathbf{A}^\dagger}(\mathbf{C}\,\sigma^E(\longrightarrow\!\!\bullet), \mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)).\end{aligned}$$

Using the equivalence (4.19), we get

$$\begin{aligned}&\Delta^{\mathbf{A}^\dagger}(\mathbf{S}_{PU-0}(\pi, f_1, f_2, S_0), \mathbf{C}\,\sigma^E(\longrightarrow\!\!\bullet)) \\ &\quad = \Delta^{\mathbf{A}^\dagger}(\mathbf{C}\,\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!\longleftarrow\!\!\bullet), \mathbf{C}\,\sigma^E(\longrightarrow\!\!\bullet)) \\ &\quad = \Delta^{\mathbf{A}^\dagger\mathbf{C}}(\pi_1^A \pi_2^B(\longrightarrow \parallel \bullet\!\!\longleftarrow\!\!\bullet), \sigma^E(\longrightarrow\!\!\bullet)).\end{aligned}$$

Applying further Lemma 2.14 and equivalence (4.20), the distinguishing advantage of $\mathbf{A}^\dagger$ distinguishing $\mathbf{C}\,\sigma^E(\longrightarrow\!\!\bullet)$ and $\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)$ is bounded by

$$\begin{aligned}&\Delta^{\mathbf{A}^\dagger}(\mathbf{C}\,\sigma^E(\longrightarrow\!\!\bullet), \mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)) \\ &\quad \leq \Delta^{\mathbf{A}^\dagger}(\mathbf{C}\,\sigma^E(\longrightarrow\!\!\bullet), \mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)|\mathcal{B}) + \mathsf{P}^{\mathbf{A}^\dagger\mathbf{C}\,\sigma^E(\longrightarrow\bullet)}(\bar{\mathcal{B}}) \\ &\quad \leq \Delta^{\mathbf{A}^\dagger}(\mathbf{C}\,\sigma^E(\longrightarrow\!\!\bullet)|\mathcal{B}, \mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)|\mathcal{B}) \\ &\qquad\quad + \mathsf{P}^{\mathbf{A}^\dagger\,\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)}(\bar{\mathcal{B}}) + \mathsf{P}^{\mathbf{A}^\dagger\mathbf{C}\,\sigma^E(\longrightarrow\bullet)}(\bar{\mathcal{B}}) \\ &\quad = \mathsf{P}^{\mathbf{A}^\dagger\,\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)}(\bar{\mathcal{B}}) + \mathsf{P}^{\mathbf{A}^\dagger\mathbf{C}\,\sigma^E(\longrightarrow\bullet)}(\bar{\mathcal{B}}).\end{aligned}$$

Note that in the system $\mathbf{C}\,\sigma^E(\longrightarrow\!\!\bullet)$, an adversary sees ['won', 0] in the first round where the condition is no longer satisfied, whereas in the system

$\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)$, the output ['won', 1] can be observed in this case, but since $\mathbf{C}\sigma^E(\longrightarrow\!\bullet)$ and $\mathbf{S}_{PU-1}(\pi, f_1, f_2, S_0)$ are equivalent as long as $\mathcal{B}$ holds and since the MES "switches" in both systems on the same condition, namely when an adversary inputs a correct guess of the decryption outcome, the probabilities that $\bar{\mathcal{B}}$ eventually holds is the same in both systems.

Using the fact that $\longrightarrow\!\bullet$ is a δ-uncertain confidential channel against t_1-predictors, the probability $\mathsf{P}^{\mathbf{A}^\dagger\mathbf{C}\sigma^E(\longrightarrow\bullet)}(\bar{\mathcal{B}})$ can be bounded by δ, otherwise $\mathbf{A}^\dagger\mathbf{C}_2(\sigma)$ would be a t_1-predictor that wins the prediction game with probability greater than δ and thus breaks the δ-uncertain property of the channel: The converter $\mathbf{C}_2$ forwards an encryption query to the A-interface and inputs the response at the inner interface of the simulator σ. The output of the simulator is forwarded back to the adversary. On a decryption output, the ciphertext is input at the outer interface of σ to get α, consecutively the α and the message of the decryption query is output at the E-interface. Note that $\mathbf{A}^\dagger\mathbf{C}_2(\sigma)$ is indeed a t_1 predictor since $\mathbf{A}$ makes at most t_2 computation steps and $\mathbf{C}_2(\sigma)$ is bound to $t_1 - t_2$ computation steps. And since the output to an encryption query is distributed identically from the viewpoint of $\mathbf{A}^\dagger$ in the game setting as well as in the channel setting, any decryption query output is distributed accordingly. And since the condition $\bar{\mathcal{B}}$ is equivalent to the game winning condition, the probability can indeed be bound by the game winning probability that is assumed for any t_1-predictor:

$$\mathsf{P}^{\mathbf{A}^\dagger\mathbf{C}\sigma^E(\longrightarrow\bullet)}(\bar{\mathcal{B}}) \leq \delta.$$

We conclude that

$$\Phi^{\mathbf{A}}(\mathbf{S}_{PU}(\pi, f_1, f_2, S_0) \leq \Delta^{\mathbf{A}^\dagger\mathbf{C}}(\pi_1^A\pi_2^B(\longrightarrow \| \bullet\!\!=\!\!\bullet), \sigma^E(\longrightarrow\!\bullet)) + 2\delta,$$

and thus by rearranging the inequality, we get for the distinguishing advantage of $\mathbf{A}^\dagger\mathbf{C}$,

$$\begin{aligned}\Delta^{\mathbf{A}^\dagger\mathbf{C}}(\pi_1^A\pi_2^B(\longrightarrow \| \bullet\!\!=\!\!\bullet), \sigma^E(\longrightarrow\!\bullet)) &\geq \Phi^{\mathbf{A}}(\mathbf{S}_{PU}(\pi, f_1, f_2, S_0) - 2\delta \\ &> \varepsilon + 2\delta - 2\delta = \varepsilon.\end{aligned}$$

As this contradicts our assumption that π constructs a δ-uncertain confidential channel $\longrightarrow\!\bullet$ against t_1-predictors with error ε from $(\longrightarrow \| \bullet\!\!=\!\!\bullet)$, no such adversary $\mathbf{A}$ can exist and π must be $(\varepsilon + 2\delta, t_2)$-secure in the sense of PU-Confidentiality. The asymptotic statement follows directly. □

4.6.4 Chosen Plaintext Forgery

A notion closely related to plaintext uncertainty (PU) is *chosen plaintext forgery*. To notion is also introduced in [19] and captures the inability of

an adversary to create a ciphertext forgery decrypting to a message that is chosen and fixed before the game starts. Tough it may seem that the two notions are very similar, they are strictly separated as the following counter-example shows: Consider an encryption scheme that is secure in the sense of chosen plaintext forgery (CPF). From this scheme, one can construct a scheme that is still secure in the sense of CPF but is clearly not PU-secure. The encryption of the new scheme takes the generates the ciphertext according to the base scheme and appends a message chosen at random and its encryption. Clearly an adversary can use this information in the PU game to submit a correct guess since he "knows" the outcome of decryption. In any CPF, where the message to which a forgery must be presented is chosen at the beginning, the probability that exactly the fixed message is "revealed" in encryption is negligible if the message space is large enough.

Since the notion of chosen plaintext forgery is thus strictly weaker than plaintext uncertainty, the formalization is skipped and only described informally.

Similarly to the malleability definition of plaintext uncertainty, a game has to be defined based on a confidential channel. That game is parametrized by a message m and the game is won if m is not input into the channel but a transformation is input resulting in the message m. A channel is defined to be chosen-plaintext unforgeable, if for any message the game cannot be won with substantial probability.

Chapter 5

Encryption with Redundancy

In the authenticate-then-encrypt paradigm (AtE) an authentication scheme is applied in a first step to authenticate a message, and consecutively an encryption scheme is used to encrypt the authenticated message. An interesting question in this context is, if even weaker types of authentication mechanisms suffice to provide security for the composed scheme. An and Bellare [2] study the authenticate-then-encrypt paradigm towards these question in a more general way: They do not restrict their focus on authentication mechanisms that employ a shared secret key, but also consider simple "redundancy functions" that are publicly known and can be computed by any party. The motivation to study even such weaker authentication mechanisms is that a sufficient strong encryption scheme might protect the added redundancy such that the combined scheme provides the needed authenticity guarantees.

In the following we denote by AtE the paradigm where the used authentication scheme uses a secret parameter for authentication and verification (i.e. a shared secret key). The paradigm to use a keyless authentication mechanism (e.g. a public redundancy code) is denoted by encryption-with-redundancy (EwR).

In [2, Theorem 4.2], the traditional game-based notions of IND-CPA, NM-CPA and IND-CCA are examined if they provide some form of protection for the underlying authentication mechanism. It is shown that these notions do not provide enough protection for any EwR composition to be sound and the result is taken as "powerful indication that the intuition that privacy helps provide integrity via encryption-with-redundancy is wrong" [2, Page 4].

In contrast to the statement above, I propose a restricted type of malleability rendering the EwR paradigm secure with only minimal requirements for the redundancy function. To show that the EwR paradigm has in fact a right to

exist and is interesting to be considered, it is shown that a practical scheme called *bidirectional IGE* conforms with the given type of malleability and a construction in the sense of EwR is in fact possible.

5.1 Definition of the Malleability

In a first step, a general definition of "adding redundancy to a message" is given in the form of the *redundancy protocol*. A useful constraint that can be made to such a protocol is that intuitively it adds the same amount of redundancy to every message. This property is called *equal-length-preserving* and formalized by requiring that two equal length messages retain equal length even after the redundancy is added by the protocol.

Definition 5.1. *A* redundancy protocol $\rho = (\rho_1, \rho_2)$ with domain space $\mathcal{M}$ and range space $\mathcal{M}'$ *is a protocol consisting of an injective converter* ρ_1 *with input space* $\mathcal{M}$ *and output space* $\mathcal{M}'$ *and its inverse converter* ρ_2*:*

$$\forall m \in \mathcal{M} : \rho_2(\rho_1(m)) = m.$$

The converter ρ_2 *is defined to output the symbol* $\perp \notin \mathcal{M}$ *on inputs not in* $\mathcal{M}'$*. The protocol* ρ *transforms a channel with message space* $\mathcal{M}'' \supseteq \mathcal{M}'$ *into a channel with message space* $\mathcal{M}$*.*

A redundancy protocol $\rho = (\rho_1, \rho_2)$ *is called* equal-length-preserving*, if for any messages* $m_1, m_2 \in \mathcal{M}$ *with* $|m_1| = |m_2|$*, it holds that* $|\rho_1(m_1)| = |\rho_1(m_2)|$*.*

A redundancy protocol family*, denoted shortly by* $\rho(k)$*, is a family of redundancy protocols parametrized by the security parameter* k*,* $\{\rho(k)\}_{k\in\mathbb{N}}$*.*

The authenticity property of such a protocol is very weak and comes from the fact that only messages from the range space and not from the whole message space are considered as valid. A confidential channel intuitively protects this authenticity property if it does not allow an adversary to apply a transformation resulting in a valid "redundant message", namely a message in $\mathcal{M}'$. A confidential channel gives sufficient protection if transformations resulting in $\mathcal{M}'$ are not possible. The following proposition of a type of malleability, called λ*-subset-unlikely*, is slightly weaker and allows only transformations whose outcome is in $\mathcal{M}'$ with at most probability λ.

Definition 5.2. *A transformation* $F_\alpha : \mathcal{M}''^* \times \mathcal{M}''^* \to \mathcal{M}''$ *is called* λ-subset-unlikely for the subset $\mathcal{M}' \subseteq \mathcal{M}''$*, if for any message history* $M_1 \in \mathcal{M}'^*, M_2 \in \mathcal{M}'^*$*, the probability that the result of the transformation is in the set* $\mathcal{M}'$ *is bounded by* λ*,*

$$\mathsf{P}(F_\alpha(M_1, M_2) \in \mathcal{M}') \leq \lambda.$$

The corresponding confidential channel is alike called λ-subset-unlikely and only allows transformations that are of the above defined form. If in an asymptotic model, λ is negligible, the channel is called *subset-blurring*.

Definition 5.3. *A confidential channel $\longrightarrow\!\bullet$ with message space $\mathcal{M}''$ and malleability $\mathcal{F} = (\{F_\alpha\}_{\alpha\in\mathcal{A}}, \{\mathcal{A}_q\}_{q\in\mathbb{N}})$ is called* λ-subset-unlikely for the subset *$\mathcal{M}' \subseteq \mathcal{M}''$, if all $\alpha \in \mathcal{A}$ correspond to λ-subset-unlikely transformations for the subset $\mathcal{M}' \subseteq \mathcal{M}''$.*

A λ-subset-unlikely confidential channel $\longrightarrow\!\bullet$ for the subset $\mathcal{M}' \subseteq \mathcal{M}''$ is called subset-blurring with respect to a security parameter k for the subset *$\mathcal{M}' \subseteq \mathcal{M}''$, denoted by $\xrightarrow{\$\$}\!\bullet$, if λ is negligible in k.*

5.2 Soundness of Encryption-with-Redundancy

In the following, it is shown that a subset-blurring confidential channel and an equal-length-preserving redundancy protocol are sufficient for the EwR composition paradigm to be sound, i.e. that the resulting composition constructs a secure channel.

Theorem 5.4. *Any equal-length-preserving redundancy protocol family $\rho(k)$ with domain space $\mathcal{M}$ and range space $\mathcal{M}'$ according to Definition 5.1 constructs a secure channel $\bullet\!\longrightarrow\!\bullet$ with message space $\mathcal{M}$ from a subset-blurring confidential channel $\xrightarrow{\$\$}\!\bullet$ for the subset $\mathcal{M}' \in \mathcal{M}''$.*

Especially, any equal-length-preserving redundancy protocol ρ with domain space $\mathcal{M}$ and range space $\mathcal{M}'$ constructs a secure channel $\bullet\!\longrightarrow\!\bullet$ with message space $\mathcal{M}$ from a λ-subset-unlikely confidential channel $\longrightarrow\!\bullet$ for the subset $\mathcal{M}' \subseteq \mathcal{M}''$ with error $q_E \cdot \lambda$ for the distinguisher class $\mathcal{A}_{q_A,q_E,t}$ and converter class $\Sigma_{t'}$.

Proof. Let $\longrightarrow\!\bullet$ be a λ-subset-unlikely confidential channel for the subset $\mathcal{M}' \subseteq \mathcal{M}''$ with malleability $\mathcal{F} = (\{F_\alpha\}_{\alpha\in\mathcal{A}}, \{\mathcal{A}_q\}_{q\in\mathbb{N}})$. Note that any transformation $F_\alpha, \alpha \in \mathcal{A}$ is efficiently implementable and λ-subset-unlikely for the subset $\mathcal{M}' \subseteq \mathcal{M}''$.

Let $\rho = (\rho_1, \rho_2)$ be an equal-length-preserving redundancy protocol. And let σ be the following simulator: σ simulates the constructed channel $\mathbf{R}_s := \rho_1^A \rho_2^B(\longrightarrow\!\bullet)$ and works as follows:

on input $X_i = l_i$ at inner interface **do**
  choose $m_i' \in \mathcal{M}$ with $|m_i'| = l_i$
  input m_i' at A-interface of $\mathbf{R}_s$ to get l_i' and $\mathcal{A}_i$
  output l_i' and $\mathcal{A}_i$ at outer interface
on input $X_i = [\text{'forward'}, j]$ at outer interface **do**

output ['forward', j] at inner interface
on input $X_i = $ ['delete'] at outer interface **do**
output ['delete'] at inner interface
on input $X_i = $ ['modify', α] at outer interface **do**
input ['modify', α] at E-interface of $\mathbf{R}_s$
output ['delete'] at inner interface
on input $X_i = $ ['replay', j] at outer interface **do**
output ['delete', j] at inner interface

We define t' to be the number of computation steps that σ makes at most, capturing the simulation of the channel $\mathbf{R}_s$, forwarding queries, and the generation of a message. We argue that t' is efficient in any reasonable model since $\mathbf{R}_s$ must be efficient in such a model.

We now compare the two channels $\mathbf{R} := \rho_1^A \rho_2^B(\longrightarrow\bullet)$ and $\mathbf{S} := \sigma^E(\bullet\longrightarrow\bullet)$: Note that on an input m_i at the A-interface, the output of the two systems at the E-interface is distributed identically if the previous input/output was distributed accordingly: $\mathbf{R}$ outputs $l_i' := |\rho_1(m_i)|$ along with the set of eligible transformations. In $\mathbf{S}$, σ simulates $\mathbf{R}_s$ on an input of the message m_i' that has the same length as m_i. Since ρ is equal-length-preserving, the simulation outputs exactly l_i'. And since the set of eligible transformations depends only on the message lengths, the previous sets and the chosen transformations, all known to the simulated channel $\mathbf{R}_s$, the set of eligible transformation $\mathcal{A}_i$ that is output by the simulation has the same distribution as the output of the real channel.

For the analysis of the output at the B-interface of the two systems, let $\mathcal{B} = B_1, B_2, \ldots$ denote the MES with B_i being the event that for any input of the form ['modify', α] at the E-interface in all previous rounds j, $j \leq i$, the output at the B-interface was $\perp$. Now we claim that the following conditional equivalence holds:

$$\mathbf{R}|\mathcal{B} \equiv \mathbf{S}. \tag{5.1}$$

The output of the systems on forwarding and deleting inputs at the E-interface are distributed identically if all previous inputs/outputs were. Replaying inputs are answered in both systems by a $\perp$-symbol: In the first system, every previously chosen transformation resulted in $\perp$ due to the condition $\mathcal{B}$, and thus only $\perp$ can be replayed. In the second system, the simulator always deletes the message resulting similarly in $\perp$. For a modify query, $\mathbf{R}|\mathcal{B}$ always outputs $\perp$ at the B-interface due to the condition $\mathcal{B}$. In $\mathbf{S}$, exactly the same behavior can be observed as the simulator transforms any modify query into a delete query to the channel, resulting also in a $\perp$ symbol at the B-interface. We conclude that every query preserves the equality of the output distribution of the two systems and thus the above equivalence holds.

Applying Lemma 2.14 with (5.1), we get for any distinguisher $\mathbf{D} \in \mathcal{A}_{q_A,q_E,t}$,

$$\Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{S}) \leq \mathsf{P}^{\mathbf{DR}}(\bar{\mathcal{B}}).$$

Consecutively, we want to show that the probability that eventually $\bar{\mathcal{B}}$ holds in **DR** is bounded by $q_E \cdot \lambda$. Note that for any message $m' \in \mathcal{M}''$, ρ_2 outputs $\perp$ except if $m' \in \mathcal{M}'$. And since $\longrightarrow\!\bullet$ is defined such that any transformation is λ-subset-unlikely, the probability of any transformation to output a message in $\mathcal{M}'$ is at most λ. As $\mathbf{D}$ makes at most q_E queries at the E-interface and thus at most q_E transformations are evaluated, the probability that eventually a transformation results in a message in $\mathcal{M}'$ and ρ_2 presents an output different from $\perp$ is, by using the union bound, at most $q_E \cdot \lambda$. We get, as stated in the lemma,

$$\forall \mathbf{D} \in \mathcal{A}_{q_A,q_E,t} : \Delta^{\mathbf{D}}(\rho_1^A \rho_2^B(\longrightarrow\!\bullet), \sigma^E(\bullet\!\longrightarrow\!\bullet)) \leq q_E \cdot \lambda. \tag{5.2}$$

The asymptotic statement for a redundancy protocol family $\rho(k)$ follows since adversaries are required to be efficient in k and since a subset-blurring confidential channel $\xrightarrow{\$\$}\!\bullet$ is λ-subset-unlikely with λ negligible in k. Thus, also q_E must be efficient in k and $q_E \cdot \lambda$ must be negligible in k. □

5.3 Example Scheme

To give the introduced definition of the subset-blurring confidential channel and the EwR paradigm a justification for their existence, it is shown that there exists a practical scheme that meets the definition. As illustration example, a scheme analyzed in [19] is taken. The scheme is called *bidirectional infinite garble extension mode* (BIGE) and is based on the original infinite garble extension mode introduced by Campbell [10].

BIGE is based on shared uniform random permutations (URP's), $\mathbf{P} : \{0,1\}^n \to \{0,1\}^n$ and operates in two sequential block chaining sequences where the block length is n bits. The variant of BIGE specified in [19] uses as resources a channel with message space $\{0,1\}^{nl}$ for $l \in \mathbb{N}$, and three shared URP's $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$. Encryption works basically as follows: The base IGE encryption is used on the message blocks resulting in an intermediate ciphertext. IGE encryption takes iteratively every message block, uses a random permutation on the bit-wise xor of this block and the previous ciphertext block resulting in an intermediate block that is subsequently bit-wisely xored with the previous message block (as in CBC mode). In a second phase, the process of IGE encryption is repeated on the intermediate ciphertext, but in reverse block order.

Formally, encryption of an nl-bit string works as follows:

bigeenc$(m = (m_1, \ldots, m_l))$:
generate $m_0 \in_R \{0,1\}^n$
$z_0 \leftarrow \mathbf{P}_1(m_0)$
for $i = 1, \ldots, l$ **do**
$z_i \leftarrow \mathbf{P}_2(m_i \oplus z_{i-1}) \oplus m_{i-1}$
$c_0 \leftarrow \mathbf{P}_1(z_l)$
for $i = 1, \ldots, l$ **do**
$c_i \leftarrow \mathbf{P}_3(z_{l-i} \oplus c_{i-1}) \oplus z_{l-i+1}$
return $c = (c_0, c_1, \ldots, c_l)$

Conversely, BIGE decryption of an nl-bit string works as follows:

bigedec$(c = (c_1, \ldots, c_m))$:
$z_l \leftarrow \mathbf{P}_1^{-1}(c_0)$
for $i = 1, \ldots, l$ **do**
$z_{l-i} \leftarrow \mathbf{P}_3^{-1}(c_i \oplus z_{l-i+1}) \oplus c_{i-1}$
$m_0 \leftarrow \mathbf{P}_1^{-1}(z_0)$
for $i = 1, \ldots, l$ **do**
$m_i \leftarrow \mathbf{P}_2^{-1}(z_i \oplus m_{i-1}) \oplus z_{i-1}$
return $m = (m_0, m_1, \ldots, m_l)$

Note that $\mathbf{P}_1$ is used to permute the random IV values, the two other URP's are used for the first and the second IGE phase respectively. Since chaining is symmetric in encryption and decryption, a modification to a ciphertext propagates (with overwhelming probability) to the end of the intermediate ciphertext. This results in a modification of the first intermediate ciphertext block (as the IGE phases work in the opposite directions) and thus the "error" propagates through the whole message in the second phase. Since random permutations are used, such an error corresponds to the randomization of the message blocks.

For the analysis of the scheme, recall the corresponding arguments of a similar analysis of the scheme in [19].

Remark 5.5. *Consider the situation after an encryption of a message m^e with nl' bits resulting in ciphertext c^e. A ciphertext $c = (c_1, \ldots, c_l)$ that is distinct from c^e is submitted to decryption. Analogously to the proof of Lemma 6 in [19], the following sets of blocks are defined: S^e is the set of blocks that were previously output by $\mathbf{P}_2$ during encryption of the message. The set S^d denotes the set of blocks that are input to $\mathbf{P}_2^{-1}$ in the current decryption process. Formally, the sets are defined as follows:*

$$S^e := \{z_k^e \oplus m_{k-1}^e, 1 \leq k \leq l'\}$$
$$S^d := \{z_s \oplus m_{s-1}, 1 \leq s \leq l\}.$$

Let us now define the event $\mathcal{B}$ meaning that S^d is collision-free (i.e. that $|S^d| = l$) and $S^e \cap S^d = \emptyset$. According to the proof of Lemma 6 in [19], the probability that the event $\bar{\mathcal{B}}$ occurs can be bounded by:

$$\mathsf{P}(\bar{\mathcal{B}}) \leq \frac{nl'(nl'-n)}{n^2 2^n} + \frac{nl'(nl'+n)}{n^2 2^{n+1}} + \frac{3(l+1)nl'}{n2^n} + \frac{3(l+1)^2}{2^{n+1}}.$$

For the following analysis of the channel construction that is achieved by BIGE, the scenario is simplified to the case of a single-use channel with a message space that contains bit strings of the fixed length $nl, l \in \mathbb{N}$. The analysis should, however, be transferable to the more general case using similar arguments, but a more carful study of the collision probabilities.

Using the above bound, the following lemma shows that BIGE constructs a single-use subset-blurring confidential channel from a single-use insecure channel and three URP's.

Lemma 5.6. *Let l' be defined as $l' := l + 1$, and let f be a function $f : \{0,1\}^{nl-1} \to \{0,1\}^n$ that maps $n(l-1)$-bit strings to n-bit strings. Then, the encryption protocol (*bigeenc, bigedec*) with block size n constructs a single-use $\frac{1}{2^n}$-subset-unlikely confidential channel for the subset $\{(m,t)|m \in \{0,1\}^{n(l-1)} \wedge f(m) = t\} \subseteq \{0,1\}^{nl}$, from a single-use insecure channel $\longrightarrow$ with message space $\{0,1\}^{nl}$, and three URP's $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$, for the distinguisher class $\mathcal{A}_t$ and for the converter class $\Sigma_{t'}$. The protocol constructs the channel with error $\frac{16l^2+1}{2^n} + 2 \cdot \delta_{nu}$ where δ_{nu} is defined as follows:*

$$\delta_{nu} := \frac{nl'(nl'-n)}{n^2 2^n} + \frac{nl'(nl'+n)}{n^2 2^{n+1}} + \frac{3(l'+1)nl'}{n2^n} + \frac{3(l'+1)^2}{2^{n+1}}.$$

Proof. Let $\longrightarrow\!\!\bullet$ be a single-use confidential channel with the malleability $\mathcal{F} = \{F_l\}$ defined as follows: The transformation F_l outputs, for any input arguments, a uniform random bit string of the length nl. Note that $\longrightarrow\!\!\bullet$ is indeed $\frac{1}{2^n}$-subset-unlikely for the subset $\mathcal{M}' := \{(m,t)|m \in \{0,1\}^{n(l-1)} \wedge f(m) = t\} \subseteq \{0,1\}^{nl}$ since the probability P_{subset} that a uniform random bit string of length l is in the subset $\mathcal{M}'$ is

$$\mathsf{P}_{subset} = \frac{|\mathcal{M}'|}{|\{0,1\}^{nl}|} = \frac{2^{n(l-1)}}{2^{nl}} = \frac{1}{2^n}.$$

Let σ be the simulator working as follows:

on first input $len = n \cdot l$ at inner interface **do**
generate $c \in_R \{0,1\}^{len}$
output c at outer interface

on first input $c' \in \{0,1\}^{nl}$ at outer interface **do**
  if $c' = c$ **then**
    output $['\text{forward}', 1]$ at inner interface
  else
    output $['\text{modify}', l]$ at inner interface

The quantity t' is defined as the number of computation steps σ makes at most, capturing the generation of a random message and forwarding of messages. Clearly, t' is an efficient quantity in any reasonable model.

Note that leaking the length and specifying the transformation explicitly is not necessary, but highlights how an argument could work for the more general case where arbitrary values of l are considered.

For the following analysis, we use the notation $\mathbf{R} := \mathsf{bigeenc}^A\mathsf{bigedec}^B(\longrightarrow \|\mathbf{P}_1\|\mathbf{P}_2\|\mathbf{P}_3)$ and $\mathbf{S} := \sigma^E(\longrightarrow\!\!\bullet)$ and define additionally a variant of the real channel where the BIGE converters use some slightly modified URP's for encryption and decryption: A system $\bar{\mathbf{P}}$ is constructed from the URP $\mathbf{P}$ such that the system, and also its inverse, outputs, on an input that has never been input (or output in the case of the inverse) before, a uniform random bit string (i.e. potentially one that was already output, resulting in a collision), and on "known" inputs (or outputs for the inverse), it outputs the same bit string as before. We call such a system in the following *modified URP*. Based on this definition, we specify a modified real channel $\bar{\mathbf{R}} := \mathsf{bigeenc}^A\mathsf{bigedec}^B(\longrightarrow \|\bar{\mathbf{P}}_1\|\bar{\mathbf{P}}_2\|\bar{\mathbf{P}}_3)$.

We now define the following MES $\mathcal{B} = B_1, B_2, \ldots$ that allows a simpler comparison of systems. B_i is defined as the event $\mathbf{B}$ from Remark 5.5 if the input in round i is a an input at the E-interface consisting of a ciphertext distinct from a (potentially) previously output ciphertext, and as B_{i-1} otherwise. Since we study a single-use channel where at most one encryption and one decryption query is processed, this event is well defined.

Note that a query at the E-interface is answered by the system $\bar{\mathbf{R}}$ conditioned on $\mathcal{B}$ by a uniform random bit string if the ciphertext is distinct from a previously seen ciphertext (since all inputs to the modified URP's are distinct from any previous inputs and thus a random bit string is output) or by the message that was encrypted previously if an already seen ciphertext is input. Thus, the output on an input at the E-interface is distributed identically to the system $\mathbf{S}$ where in the first case, a transformation is chosen that outputs a uniform random bit string of the appropriate length, and in the second case, the corresponding message is forwarded.

A second MES $\mathcal{C} = C_1, C_2, \ldots$ is defined for the comparison of queries at the

A-interface. The event C_i is defined with respect to the two following sets:[1]

$$T^d := \{z^d_{l-k} \oplus c^d_{k-1}, 1 \leq k \leq l\} \cup \{z^d_l\}$$
$$T^e := \{z_{l-s} \oplus c_{s-1}, 1 \leq s \leq l\} \cup \{z_l\}.$$

More precisely, C_i holds if in round i a message that is different from a previous decrypted message (including the hidden IV block) is input at the A-interface, T^e is collision-free and $T^e \cap T^d = \emptyset$. And C_i is defined as C_{i-1} otherwise (i.e. no input at the A-interface in round i or an input that is not "new"). In addition, the MES $\mathcal{D}$ is defined as consisting of events D_i that hold if, up to round i, no "IV-value" (i.e. blocks m_0) collide.

Thus, an input at the A-interface to the system $\bar{\mathbf{R}}$ conditioned on $\mathcal{C}$ and $\mathcal{D}$ results in a uniform random bit string since the IV is distinct from a previous IV and thus the message to encrypt as whole is distinct from a previous seen message, and since the input to the modified URP's are all distinct from previous inputs and thus uniform random values are output. This behavior is identical to the one of the ideal system $\mathbf{S}$, where an input at the A-interface always results in a uniform random bit string of the appropriate length.

We conclude that the output to both inputs at the A- and E-interface are distributed identically in the two systems and they are thus equivalent,

$$\bar{\mathbf{R}}|\mathcal{B} \wedge \mathcal{C} \wedge \mathcal{D} \equiv \mathbf{S}.$$

For the comparison of the modified real system and the real system, the MES $\mathcal{E}$ is defined as sequence of events E_i where E_i holds if, up to round i, all the outputs of the modified URP's do not collide. Clearly the modified URP's, conditioned on $\mathcal{E}$, behave identically to the original ones, and thus

$$\mathbf{R} \equiv \bar{\mathbf{R}}|\mathcal{E}.$$

Using the triangle inequality and Lemma 2.14, we get for any distinguisher $\mathbf{D} \in \mathcal{A}_t$

$$\begin{aligned}\Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{S}) &\leq \Delta^{\mathbf{D}}(\mathbf{R}, \bar{\mathbf{R}}) + \Delta^{\mathbf{D}}(\bar{\mathbf{R}}, \mathbf{S}) \\ &\leq \mathsf{P}^{\mathbf{D}\bar{\mathbf{R}}}(\bar{\mathcal{E}}) + \mathsf{P}^{\mathbf{D}\bar{\mathbf{R}}}(\bar{\mathcal{B}} \vee \bar{\mathcal{C}} \vee \bar{\mathcal{D}}) \\ &\leq \mathsf{P}^{\mathbf{D}\bar{\mathbf{R}}}(\bar{\mathcal{E}}) + \mathsf{P}^{\mathbf{D}\bar{\mathbf{R}}}(\bar{\mathcal{B}}) + \mathsf{P}^{\mathbf{D}\bar{\mathbf{R}}}(\bar{\mathcal{C}}) + \mathsf{P}^{\mathbf{D}\bar{\mathbf{R}}}(\bar{\mathcal{D}}).\end{aligned}$$

For the probability of a collision in the modified real system, we note that after at most one encryption and one decryption, there were at most $4 \cdot l$ invocations of a modified URP's, and thus

$$\mathsf{P}^{\mathbf{D}\bar{\mathbf{R}}}(\bar{\mathcal{E}}) \leq \frac{1}{2} \cdot (4l)^2 \cdot \frac{1}{2^n} = \frac{8l^2}{2^n}.$$

[1] Note that the sets contain one more element than the sets defined in the remark. This is due to the fact, that in encryption c_0 is output whereas m_0 is not output in decryption.

The probabilities for $\bar{\mathcal{B}}$ and $\bar{\mathcal{C}}$ to occur are bounded using Remark 5.5 (i.e. the analysis used in the proof of [19, Lemma 6]) and the fact the encryption and decryption of BIGE is symmetric (with the subtlety that the sets the MES $\mathcal{C}$ is defined on are one element larger):

$$\mathsf{P}^{\mathbf{D}\bar{\mathbf{R}}}(\bar{\mathcal{B}}) \leq \frac{nl(nl-n)}{n^2 2^n} + \frac{nl(nl+n)}{n^2 2^{n+1}} + \frac{3(l+1)nl}{n2^n} + \frac{3(l+1)^2}{2^{n+1}} \leq \delta_{nu}$$
$$\mathsf{P}^{\mathbf{D}\bar{\mathbf{R}}}(\bar{\mathcal{C}}) \leq \frac{nl'(nl'-n)}{n^2 2^n} + \frac{nl'(nl'+n)}{n^2 2^{n+1}} + \frac{3(l'+1)nl'}{n2^n} + \frac{3(l'+1)^2}{2^{n+1}} \leq \delta_{nu}.$$

Finally, the probability that $\bar{\mathcal{B}}$ occurs can be bounded by the probability that the IV chosen uniformly at random during the encryption happens to be the same as the corresponding "IV" resulting from a decryption:

$$\mathsf{P}^{\mathbf{D}\bar{\mathbf{R}}}(\bar{\mathcal{D}}) \leq \frac{1}{2^n}.$$

Combining these bounds, we get, as stated in the lemma,

$$\Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{S}) \leq \frac{16l^2+1}{2^n} + 2 \cdot \delta_{nu}.$$ □

Chapter 6

Discussion and Conclusion

6.1 Discussion

The primary goal of an encryption scheme is confidentiality of the encrypted communication. Beyond that, encryption schemes intuitively vary substantially in terms of the integrity properties they provide. The guarantees concerning the integrity of encryption are captured in the malleability characteristics of an encryption scheme and of the communication channel that is constructed by such a scheme. In this thesis, the general model of malleability, introduced in [29] and stemming from the paradigm of constructive cryptography, is considered. The general model of malleability imposes a natural scale in terms of security: The less malleable a scheme, the more secure it is.

A major part of formalizations addressing the security guarantees of encryption schemes were done using a game-based approach. This resulted in a large variety of different notions and a large number of definitions for a particular notion. In an attempt to clarify the meaning of certain notions, they are represented and examined in the general model of malleability of constructive cryptography.

6.1.1 Indistinguishability and Non-malleability

The first finding is that a set of definitions related to the notion of *indistinguishability of encryption* (e.g. IND-CCA [6, 23, 7], IND-RCCA [12]) and the definitions of *non-malleability* [17, 6, 19, 23, 7] capture essentially the same security guarantee for communication: Communication is confidential and allows modifications only in the form of inserting "constant, unrelated" mes-

sages. This statement, given in Section 4.3, is valid for both public-key and secret-key schemes.

Public-key encryption In the case of public-key encryption, the result is as expected and justifies the existence of the notions: Since the type of malleability described by these notions is the strongest that is achievable for schemes where any party can always encrypt a constant message using the public key, this type of malleability can be seen as the ultimate security goal of a public-key encryption scheme. However, it has to be pointed out that traditional definitions of the notions often use the CCA attack model that is too strong for this context. As argued in Section 4.2, the RCCA model that provides the framework for the definition in this thesis is the appropriate attack model.

Secret-key encryption For the secret-key case, it is, however, unclear how a malleability that allows inserting constant messages can be useful. It is an odd property for a scheme where, intuitively, encryption (i.e. creation of valid ciphertexts) should only be possible with the secret key to be able to create ciphertexts of constant messages, especially if the message the ciphertext decrypts to is known or can be chosen.

6.1.2 Integrity of encryption

As in the case of secret-key encryption, the meaning and usefulness of the notion of non-malleability is not clear because the notion induces a malleability giving an adversary the ability to insert constant messages into communication. Several notions addressing the elimination of such an ability and capturing the integrity of encryption are thus studied and translated into the general constructive model of malleability. In Section 4.4, I present the finding that the goal of notions such as *integrity of plaintexts*, *integrity of ciphertexts*, and *existential unforgeability* translates into communication that allows no malleability at all, which refers to the construction of a secure channel. Schemes that are secure with respect to such notions cover both confidentiality and authenticity in one single step and are thus called *authenticated encryption schemes*.

6.1.3 History-dependent Malleability

Most of the existing game-based notions only cover schemes that allow no "real" malleability, namely a malleability that allows transformations depending on the message history. The problem is that traditional attack

models, such as CCA or RCCA, exclude such transformations and the definition of a general attack model allowing such transformations is not trivial. I propose such a general model for games in the "real-or-random"-spirit, based on the simulation of both encryption and decryption in the random case. This simulation is not given explicitly as it may strongly depend on the encryption scheme in question.

Using the general attack model, I also propose a game-based definition of "pure" confidentiality. This is the first such game-based definition capturing the confidentiality of the corresponding encrypted communication without any restriction to the malleability of communication. An open question is whether the simulation of encryption can be given in a more game-like spirit (e.g. by encryption of a random message).

Plaintext-uncertainty and Chosen-plaintext forgery Two notions that allow history-dependent malleability and seem to represent an interesting type of malleability are *plaintext-uncertainty* and *chosen-plaintext forgery*. Since the original definitions were only given in weak attack models that are not considered here (e.g. CPA), the notions are adapted into the newly introduced general attack model. The first idea to formalize the malleability properties of plaintext uncertainty by defining the min-entropy characteristics of the allowed transformations is shown to be insufficient. Therefore, the definition given in Section 4.6 states that the malleability is restricted to transformations for which it is computationally hard to predict their outcome. Chosen-plaintext forgery captures a very similar type where the transformations should be unpredictable for any message that was fixed in advance (prior to key generation).

6.2 Conclusion

The constructive model of malleability used in this thesis allows a very natural and meaningful view on the properties of encryption schemes. It has been shown how several traditional game-based security notions for encryption schemes translate to this model. Additionally, the notion of "pure" confidentiality is introduced allowing arbitrary malleability. A selection of such restricted types of malleability that seem to be interesting for practical purposes are presented in Table 6.1. To complete the picture, the type of malleability given in [29] that is sufficient for a sound AtE composition is added as well.

The overview includes the game-based notions that are equivalent to the respective type of malleability. It further recalls the channel construction and

the set of allowed transformations that corresponds to the type of malleability. Moreover, it names the paradigm for which the malleability is useful.

Game notions	*Channel construction*	*Allowed F_α*	*Paradigm*
INT-PTXT, INT-CTXT, EF-RCCA	$(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ to $\bullet\!\!\longrightarrow\!\!\bullet$	none	Authenticated encryption schemes
?	$(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ to $\overset{\$\$}{\longrightarrow}\bullet$	subset-unlikely	EwR
IND-RCCA NM-RCCA	$(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ to $\overset{NM}{\longrightarrow}\bullet$	constant	(AtE)
?	$(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ to $\overset{AtE}{\longrightarrow}\bullet$	forwarding, deleting and reconstructible	AtE
Confidentiality	$(\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ to $\longrightarrow\!\!\bullet$	all	(EtA)
IND-CPA	$(\bullet\!\!\longrightarrow \parallel \bullet\!\!=\!\!\bullet)$ to $\bullet\!\!\longrightarrow\!\!\bullet$	undefined	EtA

Table 6.1: Overview of the different game-based notions and types of malleability introduced in this thesis with the corresponding composition paradigm in which they can be used, ordered by increasing security requirements to the corresponding encryption schemes.

The types of malleability are ordered according to the level of security requirements made to a corresponding encryption scheme. For most composition paradigms (except for AtE), the intuition that the more secure an encryption scheme must be in a composition, the less security requirements are made to the corresponding authentication scheme of the composition, holds true. Needless to say, there are four composition paradigms that have been shown to be sound when an encryption scheme with the corresponding malleability properties is applied. The least security requirements are made to an encryption scheme used in EtA. Such a scheme must not even satisfy the pure definition of confidentiality since decryption potentially can leak the secret key.

A higher level of security is required from a scheme to be securely applied in an AtE composition. The malleability of the scheme is restricted to be AtE-compatible according to [29]. The channel constructed by such a scheme is denoted here by the symbol $\overset{AtE}{\longrightarrow}\bullet$. Non-malleability defines an even higher level of security, but it is unclear how this traditional notion and the cor-

responding type of malleability can be used in the context of secret-key encryption schemes.

The EwR composition paradigm moves almost all the security requirements from the authentication scheme to the encryption scheme. The possibility to use a keyless redundancy protocol comes at the price of a very restricted type of malleability for the encryption scheme: It must construct a subset-blurring confidential channel with the corresponding type of malleability that I introduced in this thesis.

Finally, it is also possible to use no authentication scheme at all, that is, to require that the encryption scheme does not allow any malleability at all and thus constructs a secure channel.

I conclude that Table 6.1 includes a short summary and an overview of the results of my thesis. The table also presents four different practical variants of constructing a secure channel from an insecure channel. The composition variants differ in the level of security that is required from the encryption scheme. Using the channel-based approach of constructive cryptography with the general model of malleability, the respective security requirements can be modeled in a natural way and can thus be easily compared.

Acknowledgements

I warmly thank Professor Ueli Maurer for the opportunity to work with the Cryptography Research Group of the Institute for Theoretical Computer Science at ETH Zurich.

I would like to show my gratitude to my supervisor Björn Tackmann who helped me develop a better understanding of the subject and learned me how ideas and thoughts can be made precise and be couched. This work would not have been possible without his helping comments, encouragement and guidance throughout the whole process of writing this thesis.

It is a pleasure to show my appreciation to Annette Kindschi for proofreading and revising the English of my thesis.

Finally, I would like to thank Kenny Paterson for referring to further related literature, and all of those supporting me in any way during the development of this work.

Andreas Rüedlinger

Bibliography

[1] J. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Advances in Cryptology—EUROCRYPT 2002*, pages 83–107. Springer Berlin / Heidelberg, 2002.

[2] Jee An and Mihir Bellare. Does encryption with redundancy provide authenticity? In *Advances in Cryptology — EUROCRYPT 2001*, pages 512–528. Springer Berlin / Heidelberg, 2001.

[3] Michael Backes, Birgit Pfitzmann, and Michael Waidner. The reactive simulatability (rsim) framework for asynchronous systems. *Information and Computation*, 205(12):1685 – 1720, 2007.

[4] Boaz Barak, Ronen Shaltiel, and Avi Wigderson. Computational analogues of entropy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 200–215. Springer Berlin / Heidelberg, 2003.

[5] Mihir Bellare, Anand Desai, Eron Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. *Proceedings of the 38th Symposium on Foundations of Computer Science*, pages 394–403, 1997.

[6] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology — CRYPTO '98*, pages 26–45. Springer Berlin / Heidelberg, 1998.

[7] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology*, 21:469–491, 2008.

[8] Mihir Bellare and Amit Sahai. Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In *Advances in Cryptology — CRYPTO' 99*, pages 519–536. Springer Berlin / Heidelberg, 1999.

[9] Mihir Bellare and Amit Sahai. Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. http://eprint.iacr.org/2006/228, 2006. This is a reviewed version of [8].

[10] Carl Campbell. Design and specification of cryptographic capabilities. *Communications Society Magazine, IEEE*, 16(6):15–19, 1978.

[11] Ran Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, October 2001.

[12] Ran Canetti, Hugo Krawczyk, and Jesper Nielsen. Relaxing chosen-ciphertext security. In *Advances in Cryptology - CRYPTO 2003*, pages 565–582. Springer Berlin / Heidelberg, 2003.

[13] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Advances in Cryptology — CRYPTO '98*, volume 1462, pages 13–25. Springer Berlin / Heidelberg, 1998.

[14] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. *ACM Trans. Inf. Syst. Secur.*, 3:161–185, 2000.

[15] T. Dierks and Eric Rescorla. RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2, August 2008.

[16] D. Dolev and A. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198 – 208, 1983.

[17] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. In *Proceedings of the 23rd annual ACM symposium on Theory of computing*, pages 542–552. ACM, 1991.

[18] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000. This is an extended version of [17].

[19] Virgil D. Gligor, Pompiliu Donescu, and Jonathan Katz. On message integrity in symmetric encryption. Manuscript, February 2002.

[20] Oded Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, 6:21–53, 1993.

[21] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270 – 299, 1984.

[22] Jonathan Katz and Moti Yung. Unforgeable encryption and chosen ciphertext secure modes of operation. In *Fast Software Encryption*, pages 25–36. Springer Berlin / Heidelberg, 2001.

[23] Jonathan Katz and Moti Yung. Characterization of security notions for probabilistic private-key encryption. *Journal of Cryptology*, 19:67–95, 2006.

[24] Hugo Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is ssl?). In *Advances in Cryptology — CRYPTO 2001*, pages 310–331, 2001.

[25] Ueli Maurer. Indistinguishability of random systems. In *Advances in Cryptology — EUROCRYPT 2002*, pages 110–132, May 2002.

[26] Ueli Maurer. Constructive cryptography – a primer. In *Financial Cryptography and Data Security*, pages 1–1. Springer Berlin / Heidelberg, 2010.

[27] Ueli Maurer. Cryptography. Lecture Notes, Cryptography, ETH Zurich, 2010.

[28] Ueli Maurer and Renato Renner. Abstract cryptography. In *Innovations in Computer Science - ICS 2011*, pages 1–21, 2011.

[29] Ueli Maurer and Björn Tackmann. On the soundness of authenticate-then-encrypt: formalizing the malleability of symmetric encryption. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 505–515, 2010.

[30] Silvio Micali, Charles Rackoff, and Bob Sloan. The notion of security for probabilistic cryptosystems (extended abstract). In Andrew Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, volume 263, pages 381–392. Springer Berlin / Heidelberg, 1987.

[31] Chanathip Namprempre. Secure channels based on authenticated encryption schemes: A simple characterization. In Yuliang Zheng, editor, *Advances in Cryptology — ASIACRYPT 2002*, volume 2501, pages 111–118. Springer Berlin / Heidelberg, 2002.

[32] Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Relations among notions of non-malleability for encryption. In *Advances in Cryptology – ASIACRYPT 2007*, pages 519–535. Springer Berlin / Heidelberg, 2007.

[33] B. Pfitzmann and M. Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, pages 184 – 200, 2001.

[34] Claude E. Shannon. Communication theory of secrecy systems. *Bell Syst. Tech J.*, 28(4):656–715, October 1949.

[35] Victor Shoup. A proposal for an ISO standard for public key encryption. *IACR E-Print Archive*, 112, 2001.