Confidentiality and Integrity: A Constructive Perspective

Ueli Maurer, Andreas Rüedlinger, and Björn Tackmann

Department of Computer Science, ETH Zürich, Switzerland
{maurer,bjoernt}@inf.ethz.ch, andreas.rueedlinger@gmail.com

Abstract. Traditional security definitions in the context of secure communication specify properties of cryptographic schemes. For symmetric encryption schemes, these properties are intended to capture the protection of the confidentiality or the integrity of the encrypted messages. A vast variety of such definitions has emerged in the literature and, despite the efforts of previous work, the relations and interplay of many of these notions (which are a priori not composable) are unexplored. Also, the exact guarantees implied by the properties are hard to understand.
In constructive cryptography, notions such as confidentiality and integrity appear as attributes of channels, i.e., the communication itself. This makes the guarantees achieved by cryptographic schemes explicit, and leads to security definitions that are composable.
In this work, we follow the approach of constructive cryptography, questioning the justification for the existing (game-based) security definitions. In particular, we compare these definitions with related constructive notions and find that some are too weak, such as INT-PTXT, or artificially strong, such as INT-CTXT. Others appear unsuitable for symmetric encryption, such as IND-CCA.
Keywords: confidentiality, integrity, constructive cryptography.

1 Introduction

Symmetric encryption protects the confidentiality of messages transmitted between two parties that share a secret key. Intuitively, this means that the encrypted message (the ciphertext) transmitted from the sender A to the receiver B does not leak information about the contents of the message (other than, for example, its length). In contrast, encryption generally does not protect integrity: If the ciphertext is modified during transmission, the message obtained by decrypting might differ from the original message.

For some applications of encryption schemes, bare confidentiality is not sufficient. In his analysis of the Authenticate-then-Encrypt (AtE) transformation, Krawczyk [18] constructs an encryption scheme that guarantees confidentiality, but if one uses it to encrypt authenticated plaintexts, the combined scheme does not guarantee both confidentiality and integrity. The vulnerabilities can either be seen as a breach of confidentiality [18] or as a breach of integrity, see Sect. 4.4. Natural candidates, such as the cipher block chaining mode (CBC)

or stream ciphers, are not vulnerable; they provide weak but sufficient integrity guarantees [25].

In this paper, we use the approach of constructive cryptography [21, 22] for a systematic treatment of security notions for symmetric encryption schemes. This approach leads to security definitions that capture the exact conditions that the schemes have to satisfy to achieve certain guarantees for the message transmission. In particular, these definitions are composable, which is instrumental for the soundness of a modular protocol design. We then show how different types of confidentiality and integrity are captured and compare these notions with several security definitions from the literature. This shows that some of the previous definitions are either too weak or artificially strict (which is in general undesired as it may lead to disregarding efficient schemes that are indeed sufficient).

1.1 Game-based Security Definitions

Most widely-used security definitions for cryptographic schemes in the context of secure communication are game-based. The main concept of these definitions is an interaction of two (hypothetical) entities: The challenger and the attacker. During this interaction, the attacker issues certain "oracle queries" to the challenger; these queries model the use of the scheme in applications. The game also specifies a goal for the attacker, which often corresponds to forging a message or distinguishing encryptions of different messages. The infeasibility of achieving this goal is supposed to capture the guarantees required from the scheme.

Unfortunately, the oracle queries and winning conditions of games encode the use and guarantees only implicitly, and the exact guarantees are often hard to understand. In particular, such security definitions are generally not composable, and subtle details often have a significant impact on the resulting guarantees: Examples where slight slackness in the oracle queries rendered the guarantees of games too weak are discussed in Sect. 4.

1.2 Constructive Cryptography

The foundational idea of constructive cryptography [21, 22] is to specify both the setup assumptions and the guarantees of protocols explicitly as resources, and to consider a protocol as a transformation of such resources. Here, a *resource* is a shared functionality accessed by several parties (similar to the ideal functionalities in frameworks such as [2, 8]). *Real resources* are assumed functionalities needed for executing protocols (such as a network) and *ideal resources* describe the guaranteed functionalities the parties want to achieve. The way a party accesses a resource is described by the *interface* provided by the resource to this party; the resource provides one interface per party.

A *converter* systems formalizes the actions that a party performs locally, for example when it uses a cryptographic scheme. A converter has two interfaces: The *inner* interface is attached to an interface of the resource, and the *outer* interface is used by the party instead of the original interface of the resource. In particular, the composition of the resource and the converter is again a resource

with one interface for each party, which is depicted in Fig 1 for the case of symmetric encryption.

A *protocol* is a tuple (in our context just a pair) of converters, there is one such system for each (honest) party. The goal of a protocol is to *construct* a specified ideal resource from available real resources, where the meaning of "construct" is made precise in Sect. 2.3. The constructed ideal resources can again serve as real resources for other protocols.

1.3 Secure Communication

The resources considered in this work are communication functionalities with different types of security guarantees, and the goal of a cryptographic protocol is to construct a functionality with stronger guarantees from one (or more) with weaker guarantees. As the setting for communication security is described by two (honest) entities that communicate in a potentially hostile environment, we consider resources with three interfaces: One interface labeled A for the sender,[1] one labeled B for the receiver, and a third one that is labeled E and captures potential adversarial access. A resource of this type is called a *channel* (from A to B), and its security properties are described by the capabilities provided at the E-interface. The basic types of channels are (informally) described in the following table, using the notation of [24].

- $\longrightarrow$ An *insecure channel* leaks the complete messages at the E-interface, and allows at the E-interface to delete, change, or inject messages.
- $\bullet\!\!\longrightarrow$ An *authenticated channel* leaks the complete messages. The E-interface only allows to forward or to delete messages.
- $\longrightarrow\!\!\bullet$ A *confidential channel* only leaks the length of the messages, but allows to delete, change, or inject messages.
- $\bullet\!\!\longrightarrow\!\!\bullet$ A *secure channel* only leaks the length of the messages and only allows to forward or to delete messages.

The intuitive interpretation of the symbol "$\bullet$" is that the capabilities at the marked (sender's or receiver's) side of the channel are provided exclusively to that party. Consequently, if one side is not marked, the adversary might also be able to send or receive messages. A *shared secret key* is a system $\bullet\!\!=\!\!\bullet$ that outputs the same random value at the A- and B-interfaces, and does not interact at the adversarial interface. This system models the key that is required by (symmetric) schemes; it could be generated in a key agreement protocol.

Security mechanisms such as encryption or MAC schemes are protocols that transform one type of channel (and possibly a shared secret key) into a "more secure" type of channel. In Fig. 1, the protocol $(\mathsf{enc}, \mathsf{dec})$ uses as resources a channel $\longrightarrow$ and a key $\bullet\!\!=\!\!\bullet$. The converter enc is attached with its inner interface to the A-interfaces of $\longrightarrow$ and $\bullet\!\!=\!\!\bullet$ (dec is attached to the B-interfaces), and the outer interfaces of enc and dec are the interfaces of the constructed (dashed) system, which is again a channel. For more examples, we refer to [22, 25].

[1] Bidirectional communication also involves the analogous setting with opposite roles.

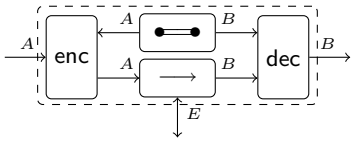

Fig. 1. Encryption protocol (enc, dec) applied to the channel $\longrightarrow$ and the key $\bullet\!\!=\!\!\bullet$.

1.4 Related Work

The major part of research on (symmetric) encryption schemes has been pursued in game-based security models. The nowadays "standard" confidentiality notions IND-CPA and IND-CCA are derived from [14] and have been translated to the setting of symmetric encryption schemes by [3]. Further variants of these notions are introduced and compared in [16]. Several types of integrity guarantees have been considered: Notions of non-malleability have been translated in [5] from the respective public-key notions [12]. Further standard notions are INT-CTXT and INT-PTXT (integrity of ciphertext and integrity of plaintext, respectively) introduced and analyzed in [5], their relation is further examined in [28]. Also, various types of unforgeability notions appear in the literature [13, 17, 18].

The security requirements for schemes used to protect communication over insecure networks is often specified as a combination of properties for confidentiality and integrity, where the standard combination is IND-CPA and INT-CTXT [5, 7, 17]; combinations with weaker types of integrity properties appear in [5, 9, 13, 17, 27]. A single game-based notion for authenticated encryption appeared in [31, 34]. A different approach is taken in the definition of [9]: While confidentiality is similar to IND-CPA, authenticity is simulation-based; equivalent fully game-based notions appear in [27]. Fully simulation-based definitions of secure communication have been provided in [29] for Reactive Simulatability and in [10] in the UC framework.

1.5 Outline

We analyze confidentiality and integrity notions for (symmetric) encryption schemes using the paradigm of constructive cryptography. Sect. 2 introduces the notation and the general model, and Sect. 3 shows how different types of confidentiality and integrity guarantees are captured. In Sect. 4, we compare various existing game-based security definitions to the notions in our model.

2 Preliminaries

We use the concept of abstract systems [22, 23] to formulate our results. At the highest level of abstraction, a system is an object with interfaces via which it interacts with its environment and with other systems. Every two systems can be

composed by connecting one interface of each system, and the composed object is again a system. Also, every two different systems are mutually independent.

2.1 Notation

We consider two distinct types of systems, resources and converters, and we describe topologies of these systems using the notation from [23]. Resources, with three interfaces labeled by A, B, and E, are denoted either by special symbols or by upper case boldface letters. Converters, with one *inner* and one *outer* interface, are denoted either by small Greek letters or by special identifiers such as enc or dec; the set of all converters is denoted as Σ.

The composition of a resource $\mathbf{R}$ and a converter ϕ is written as $\phi^I\mathbf{R}$, where the label $I \in \{A, B, E\}$ means that the inner interface of ϕ is attached to the I-interface of the resource $\mathbf{R}$. Note that the composed system is again a resource that exposes the outer interface of ϕ as the I-interface together with the other interfaces of $\mathbf{R}$. A protocol is a pair of converters, one for each honest party, and applying the protocol (ϕ_1, ϕ_2) to the resource $\mathbf{R}$ is defined as $\phi_1^A\phi_2^B\mathbf{R}$—attaching the converters to the A- and B-interfaces of the resource.

If two resources $\mathbf{R}$ and $\mathbf{S}$ are used in parallel, this is denoted as $\mathbf{R}\|\mathbf{S}$ and is again a resource with the same set of interfaces; each of these interfaces A, B, or E of $\mathbf{R}\|\mathbf{S}$ allows to access the corresponding interfaces of both sub-systems $\mathbf{R}$ and $\mathbf{S}$. The sequential composition of converters is denoted by $\psi\circ\phi$, and is defined by $(\psi \circ \phi)^I\mathbf{R} = \psi^I(\phi^I\mathbf{R})$ for all resources $\mathbf{R}$. The parallel composition $\psi\|\phi$ of converters is defined by $(\psi\|\phi)^I(\mathbf{R}\|\mathbf{S}) = (\psi^I\mathbf{R})\|(\phi^I\mathbf{S})$ for all $\mathbf{R}$ and $\mathbf{S}$. The term id refers to the "identity converter" that forwards all inputs and outputs.

In general, for bit-strings $x = x_1 \cdots x_n \in \{0,1\}^n$ and $l \leq n$, we denote by $x|_l$ the sub-string $x|_l = x_1 \cdots x_l$. We extend the operation "$\oplus$" to bit-strings by defining, for $x = x_1 \cdots x_n$ and $x' = x'_1 \cdots x'_n$, the ith bit of $x \oplus x'$ to be $x_i \oplus x'_i$.

2.2 Discrete Systems

In the analysis of protocols, we model all systems as (probabilistic) *discrete systems* that communicate by passing messages, where the term "discrete" refers to the value spaces of the messages as well as the time. The behavior of discrete systems is formalized by random systems [20], i.e., conditional distributions of the outputs of the system (as random variables) given all previous inputs and outputs. Each input or output is associated to a specific interface.

Discrete systems are an instance of the abstract systems concept described above. The composition of two discrete systems (such as connecting a resource and a converter via interfaces) is a discrete system whose behavior is defined via an interaction of the two sub-systems: A message that is input to the system is processed by the sub-system corresponding to the (external) interface where the message was input, and, if the sub-system provides output at the (internal) connected interface, this value is processed by the other sub-system. Once one of the two sub-systems outputs a message at an external interface, this becomes the output of the composed system. The parallel composition of two resources is

defined asynchronously: Each input at an interface A, B, or E explicitly specifies one of the sub-systems, and this sub-system is invoked with the input.

A *distinguisher* $\mathbf{D}$ is a system that connects to all interfaces A, B, and E of a resource $\mathbf{U}$ and outputs (at a separate interface) a single bit, here called W. The complete interaction of $\mathbf{D}$ and $\mathbf{U}$ defines a random experiment, and the probability that the bit W is 1 is written as $\mathsf{P}^{\mathbf{DU}}(W = 1)$. The *distinguishing advantage* of $\mathbf{D}$ for $\mathbf{U}$ and $\mathbf{V}$ measures how much the output of $\mathbf{D}$ differs when it is connected to either $\mathbf{U}$ or $\mathbf{V}$. Intuitively, if no (efficient) distinguisher differentiates between the two systems, they can be used interchangeably in any environment (as otherwise the environment serves as a distinguisher).

Definition 1 (Distinguishing advantage). *The* distinguishing advantage of a distinguisher $\mathbf{D}$ for the systems $\mathbf{U}$ and $\mathbf{V}$ *is defined as*

$$\Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V}) := \left|\mathsf{P}^{\mathbf{DU}}(W = 1) - \mathsf{P}^{\mathbf{DV}}(W = 1)\right|,$$

where W is the special output of $\mathbf{D}$. The advantage for a set $\mathcal{D}$ of distinguishers is defined as $\Delta^{\mathcal{D}}(\mathbf{U}, \mathbf{V}) := \sup_{\mathbf{D}\in\mathcal{D}} \Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V})$.

2.3 The Simulation-Based Security Definition

The paradigm of constructive cryptography is derived from [23] and follows the ideal world/real world approach similar to [8, 29]: The "real world" describes the protocol execution with two honest parties and an adversary, and is defined by the composition of the two converters of the protocol (π_1, π_2) with the real resource $\mathbf{R}$. In the "ideal world", the ideal resource $\mathbf{S}$ specifying the security goals is composed with a simulator σ connected to the E-interface. The purpose of σ is to adapt the E-interface of $\mathbf{S}$ such that it resembles the corresponding interface of $\pi_1^A\pi_2^B\mathbf{R}$. (As the adversary can emulate the behavior of σ, using $\sigma^E\mathbf{S}$ instead of $\mathbf{S}$ can only restrict the adversary's power, so using $\sigma^E\mathbf{S}$ and hence $\pi_1^A\pi_2^B\mathbf{R}$ instead of $\mathbf{S}$ is safe.)

To exclude trivial protocols, we require that if no adversary is present, the protocol must implement the specified functionality. In the definition, we use the special converter "$\perp$" that, when attached to a certain interface of a system, blocks this interface for the distinguisher.

Definition 2 (Secure construction). *The protocol π* constructs $\mathbf{S}$ from the resource $\mathbf{R}$ within ε and with respect to the set $\mathcal{D}$ of distinguishers *if*

$$\exists\, \sigma \in \Sigma:\ \Delta^{\mathcal{D}}(\pi_1^A\pi_2^B\mathbf{R}, \sigma^E\mathbf{S}) \leq \varepsilon \qquad \text{and} \qquad \Delta^{\mathcal{D}}(\pi_1^A\pi_2^B\perp^E\mathbf{R}, \perp^E\mathbf{S}) \leq \varepsilon.$$

An important property of Definition 2 is its composability. Intuitively, if a resource $\mathbf{S}$ is used in the construction of a larger system, then the composability implies that $\mathbf{S}$ can be replaced by a construction $\pi_1^A\pi_2^B\mathbf{R}$ without affecting the security of the composed system. Theorem 1, taken from [25], shows that security and availability are preserved under sequential and parallel composition.

Theorem 1 (Composition for the 3-party setting). *Let* $\mathbf{R}$*,* $\mathbf{S}$*,* $\mathbf{T}$*, and* $\mathbf{U}$ *be resources, and let* $\pi = (\pi_1, \pi_2)$ *and* $\psi = (\psi_1, \psi_2)$ *be protocols such that* π *constructs* $\mathbf{S}$ *from the resource* $\mathbf{R}$ *within* ε_1 *and* ψ *constructs* $\mathbf{T}$ *from* $\mathbf{S}$ *within* ε_2*. If the considered class of distinguishers is closed under composition with converters, that is* $\mathcal{D} \circ \Sigma \subseteq \mathcal{D}$*, then* $(\psi_1 \circ \pi_1, \psi_2 \circ \pi_2)$ *constructs* $\mathbf{T}$ *from* $\mathbf{R}$ *within* $\varepsilon_1 + \varepsilon_2$*,* $(\pi_1 \| \mathsf{id}, \pi_2 \| \mathsf{id})$ *constructs* $\mathbf{S}\|\mathbf{U}$ *from* $\mathbf{R}\|\mathbf{U}$ *within* ε_1 *and* $(\mathsf{id}\|\pi_1, \mathsf{id}\|\pi_2)$ *constructs* $\mathbf{U}\|\mathbf{S}$ *from* $\mathbf{U}\|\mathbf{R}$ *within* ε_1*.*

In asymptotic statements, a system $\mathbf{S}$ implicitly refers to a family of systems $\{\mathbf{S}_k\}_{k\in\mathbb{N}}$, and the distinguishing advantage is a real-valued function in the parameter k: For each k, one considers the distinguishing advantage where, for all involved systems, one takes the element described by this k. Efficiency notions for sets of systems and a negligibility notion for the distinguishing advantage can be chosen such that they are closed under composition. Examples are the sets of systems with a polynomial bound on the number of queries and/or the run-time, together with the standard notion of negligibility.

2.4 Resources and Protocols as Discrete Systems

This section details the resources and protocols considered in the setting of secure communication.

Channels. Let $\mathcal{M}$ be a discrete set, we usually consider $\mathcal{M} := \{0,1\}^*$. A *channel with message space* $\mathcal{M}$ is a resource that takes at the A-interface inputs from the set $\mathcal{M}$ and provides at the B-interface outputs from $\bar{\mathcal{M}} := \mathcal{M} \cup \{\boxslash\}$, where the element $\boxslash$ is interpreted as indicating a transmission error. A *single-use* channel allows for exactly one input at the A-interface and one output at the B-interface, a *multiple-use* channel allows for several (arbitrarily interleaved) such interactions. The possible interactions at the E-interface describe the security properties of the channel. For the insecure channel $\longrightarrow$, every input $m \in \mathcal{M}$ at the A-interface provokes the output m at the E-interface, and every input $m' \in \mathcal{M}$ at the E-interface leads to the output m' at the B-interface. The E-interfaces of the "more secure" types of channels are detailed in Sect. 3.

Keys. Let $\mathcal{K}$ be a discrete set, usually $\mathcal{K} := \{0,1\}^k$ for some $k \in \mathbb{N}$. A key with key space $\mathcal{K}$ is a resource that draws a key $\kappa \in \mathcal{K}$ uniformly at random and outputs it to both A and B. The E-interface does not provide any output.

Encryption Protocols. An *encryption protocol* with key space $\mathcal{K}$, message space $\mathcal{M}$, and ciphertext space $\mathcal{C}$ is a pair $(\mathsf{enc}, \mathsf{dec})$ of converters. These converters connect with their inner interfaces to a shared secret key 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 $\mathcal{M}$.

As an example, we describe the one-time pad encryption for bit-strings with length at most n. The key space in this setting is $\mathcal{K} = \{0,1\}^n$, and the message space of the assumed channel is (in general at least) the set of strings of length at most n bits, $\mathcal{M}' = \mathcal{C} = \bigcup_{l\leq n}\{0,1\}^l$.

Example 1 (The one-time pad). The encryption converter $\mathsf{otp\text{-}enc}$ (generically called enc in Fig. 1) obtains as input the n-bit key κ at the inner interface and a message m with $|m| \leq n$ at the outer interface. The message transmitted via the channel is $c = m \oplus \kappa|_{|m|}$. The decryption converter $\mathsf{otp\text{-}dec}$ obtains the key κ and the ciphertext c' at its inner interface. It computes $m' = c' \oplus \kappa|_{|c'|}$ and outputs the message m' at the outer interface.

Fig. 1 shows the setting in which the encryption and decryption converters are attached to the resources, the channel $\longrightarrow$ and the key $\bullet\!\!=\!\!\bullet$, with their inner interfaces. Both the A-interface and the B-interface of the combined (dashed) system are of the same type as for the original channel: The A-interface allows to input messages from $\mathcal{M} = \mathcal{C}$, and the B-interface outputs messages from the same set. Hence, the complete system is again a channel with message space $\mathcal{M}$ (but differs at the E-interface).

The scheme extends to multiple, say t, messages as follows. Consider a key with key space $\{0,1\}^{tn}$, and encrypt/decrypt the ith message with the bits $(i-1)n+1$ through $(i-1)n+|m_i|$. ♦

2.5 Formalizing Games

In game-based definitions, we formalize both the adversary and the game (or challenger) as systems, which are connected via their interfaces as described in Sect. 2.2. The game allows the adversary to make certain "oracle queries" via this interface. Whether or not the game is won is signaled by a special (monotone) output bit of $\mathbf{G}$ (this can be considered as an additional interface) that is initially 0 but switches to 1 as soon as the winning condition is fulfilled. For a game $\mathbf{G}$ and an adversary $\mathbf{A}$, we define the *game-winning probability* after q steps as

$$\Gamma_q^{\mathbf{A}}(\mathbf{G}) := \mathsf{P}^{\mathbf{AG}}(W_q = 1).$$

For an adversary that halts after (at most) q steps, we write $\Gamma^{\mathbf{A}}(\mathbf{G}) := \Gamma_q^{\mathbf{A}}(\mathbf{G})$. As winning the game with a certain probability might be trivial (such as when the goal is to guess a secret bit), one usually considers the *advantage* of $\mathbf{A}$, that is, the (absolute) difference between $\mathbf{A}$'s probability of winning $\mathbf{G}$ and the probability for "trivial" strategies.

If a security property of a scheme is defined by the adversary's inability to win a game $\mathbf{G}$, then we say that the scheme is ε-secure with respect to that property and a class[2] $\mathcal{D}$ of adversaries if the advantage for $\mathcal{D}$ in winning $\mathbf{G}$ is bounded by ε.

3 Notions of Confidentiality and Integrity

The security of communication channels corresponds to restrictions on the capabilities provided at the E-interface, which can be characterized according to two aspects: the amount of information leaked about transmitted messages, and

[2] We will often use the same class $\mathcal{D}$ for both adversaries and distinguishers.

the potential influence on messages delivered to the receiver. Consequently, a confidentiality guarantee bounds the amount of information that is leaked, and an integrity guarantee restricts the adversarial influence on delivered messages.

3.1 Confidentiality

A channel is perfectly confidential if no information about the transmitted plaintext message is leaked at the E-interface. We also consider weaker types of confidentiality where the "amount of leakage" is non-trivial but bounded; the (remaining) guarantee is described by a function on the transmitted messages.

Definition 3 (Leakage specification). *For some (discrete) set $\mathcal{S}$, a* leakage specification *is a family of functions $\mathcal{L} = \{\ell_i : \mathcal{M}^i \to \mathcal{S}\}_{i \geq 1}$.*

Functions ℓ_i on vectors of messages allow to capture, for example, channels that leak whether the same message is sent twice (as in deterministic encryption).

Definition 4 (Confidential channels). *For $\mathcal{L} = \{\ell_i : \mathcal{M}^i \to \mathcal{S}\}_{i \geq 1}$, let $\bullet \xrightarrow{\mathcal{L}} \bullet$ be the channel that, given inputs $m_1, \ldots, m_i$ at the A-interface, outputs the value $\ell_i(m_1, \ldots, m_i)$ at the E-interface (and only allows forwarding or deleting messages). A channel $\mathbf{C}$ is* $\mathcal{L}$-confidential *if there exists a simulator σ such that*

$$\Delta^{\mathcal{D}}(\bot^B\mathbf{C}, \bot^B\sigma^E(\bullet \xrightarrow{\mathcal{L}} \bullet)) = 0, \qquad \text{and} \qquad \Delta^{\mathcal{D}}(\bot^E\mathbf{C}, \bot^E(\bullet \xrightarrow{\mathcal{L}} \bullet)) = 0,$$

where $\mathcal{D}$ is the set of all distinguishers. If $\mathcal{M} \subseteq \{0,1\}^*$ and the leakage is restricted to $\ell_i : (m_1, \ldots, m_i) \mapsto |m_i|$ for all i, the channel is simply called confidential.

The condition of being $\mathcal{L}$-confidential is merely a restriction on the information leaked at the E-interface; there is no guarantee on the potential influence of the adversary on the delivered messages. In the security condition, this absence of guarantees is expressed by attaching the converter $\bot$ to the B-interface, which hides all messages from the distinguisher.

The goal of an encryption protocol is to construct a confidential channel from one that is not confidential. In particular, the one-time pad encryption achieves confidentiality in this sense.

Example 2 (Confidentiality achieved by the one-time pad). The ciphertext generated by the one-time pad encryption for the message $m \in \mathcal{M} = \bigcup_{l \leq n}\{0,1\}^l$ is an $|m|$-bit string of independent and uniformly distributed random bits. The information leaked to the adversary is exactly the length $|m|$ of the message: There is a simulator that, given the length $|m|$, generates a ciphertext that has exactly the same distribution as the "real" ciphertext for the message m.

This means that the leakage is described by $|\cdot| : \mathcal{M} \to \{1, \ldots, n\}$ (for multiple messages, ℓ_i maps $(m_1, \ldots, m_i)$ to $|m_i|$). The channel that is constructed by the one-time pad from the insecure channel is described in Examples 3 and 4. ⧫

3.2 Integrity

Encryption schemes in general do not protect the integrity of messages: If the adversary replaces the transmitted ciphertext c for a message $m \in \mathcal{M}$ by a ciphertext $c' \neq c$, the receiver will potentially obtain a different message $m' \in \mathcal{M}$ during the decryption. For the adversary (oblivious of m), replacing c by c' corresponds to selecting a transformation $F : \mathcal{M} \to \mathcal{M}$ that describes, for every *potentially* transmitted message $\tilde{m}$, which message $\tilde{m}' = F(\tilde{m})$ the receiver would obtain, given that the original message was $\tilde{m}$.

Example 3 (XOR-Malleability of the one-time pad). For the one-time pad encryption, the adversary can replace the transmitted ciphertext c by an arbitrary ciphertext c'. Assume that $c = m \oplus \kappa$ and $|c| = |c'|$, then this means that the receiver will compute $m' = c' \oplus \kappa = c' \oplus c \oplus m$. Hence, replacing c by c' corresponds to selecting the function $m \mapsto m \oplus (c \oplus c')$. ♦

In general, the distribution of each output at the B-interface depends on the previous inputs and outputs at all interfaces of the channel. But then, conditioned on the complete interaction at the E-interface—the adversary's knowledge—the channel "transforms" all inputs at the A- and all previous outputs at the B-interface into the next output at the B-interface; the interaction at the E-interface can be seen as a choice of a particular such *plaintext transformation*.

Definition 5 (Plaintext transformation). *Let $\mathcal{M}$ be a discrete set. A* plaintext transformation *F on $\mathcal{M}$ is a (probabilistic) transformation $\mathcal{M}^* \times \mathcal{M}^* \to \bar{\mathcal{M}}$.*

The arguments of the plaintext transformation are the sequence of messages transmitted by the sender, and the sequence of messages previously delivered to the receiver; the result is the next message delivered to the receiver. The set of all plaintext transformations available to the adversary formalizes the potential adversarial influence on the delivered messages. Of course, the less such transformations are available to the adversary, the stronger are the integrity guarantees of the channel. This is captured by the concept of *integrity specifications*.

Definition 6 (Integrity specification). *An* integrity specification *is a family $\mathcal{F} := \{\mathcal{F}_q\}_{q \in \mathbb{N}}$ of random variables with $\mathcal{F}_q \subseteq \bar{\mathcal{F}}$, where $\bar{\mathcal{F}}$ is a set of plaintext transformations.*

The random variables $\mathcal{F}_q \subseteq \bar{\mathcal{F}}$ formalize that, depending on the state of the channel, only a subset of the transformations might actually be accessible: After the qth query to the channel, the adversary may choose a transformation from the set $\mathcal{F}_q$ (note that this choice corresponds to replacing the transmitted ciphertext in the "real world"). The generality of this definition is indeed necessary to describe the malleability of certain encryption schemes, such as CBC mode [25]. There, the availability of certain transformations depends on the randomness used during the encryption, so $\mathcal{F}_q \neq \bar{\mathcal{F}}$.

Example 4 (XOR-malleability). Let m, m', c, and c' be as in Example 3. If we set $\delta := c \oplus c'$, the adversary's choice to replace c by $c' = c \oplus \delta$ can be interpreted as selecting the XOR-mask δ for the transmitted message. More generally, the plaintext transformations F_{i,j,δ_j} after i inputs at the A-interface and j inputs at the E interfaces, with $\delta_j \in \bigcup_{l \le n} \{0,1\}^l$, are described as follows:

- $i < j$: the output is a uniformly random $|\delta_j|$-bit string,
- $i \ge j$ and $|\delta_j| \le |m_j|$: the output is $m_j|_{|\delta_j|} \oplus \delta_j$,
- $i \ge j$ and $|\delta_j| > |m_j|$: the output is $m_j \oplus \delta_j$ followed by $|\delta_j| - |m_j|$ uniformly random bits.

The transformations available after i inputs at the A- and j inputs at the B-interface are, for each $\delta \in \bigcup_{l \le n} \{0,1\}^n$, the transformations $F_{i,j,\delta}$. ♦

The set $\mathcal{F}_q$ of transformations available after the qth query must be (implicitly of explicitly) known to the adversary; abstractly, a description of the set $\mathcal{F}_q$ is output to the adversary by the channel. Of course, for a confidential channel, this description must not leak any information beyond the information specified by the leakage. In the following definition, we refer to the number of queries at the A- and E-interfaces by i and j, respectively, and use $q := i + j$.

Definition 7 (Malleable confidential channel). *Let $\mathcal{L}$ be a leakage specification and $\mathcal{F}$ be an integrity specification such that the distribution of each $\mathcal{F}_q$ depends (only) on the leakage $\ell_s(m^s)$ for $1 \le s \le i$ of the messages $m_1, \ldots, m_i$, the previous sets $\mathcal{F}_1, \ldots, \mathcal{F}_{q-1}$, and the selected transformations $F_1, \ldots, F_j$. An* $\mathcal{F}$-malleable $\mathcal{L}$-confidential channel $\xrightarrow{\mathcal{L},\mathcal{F}}\!\!\bullet$ *(in the following only $\longrightarrow\!\!\bullet$ if $\mathcal{L}$ and $\mathcal{F}$ are clear) is an $\mathcal{L}$-confidential channel with malleability described by $\mathcal{F}$.*

On receiving m_i at the A-interface, $\longrightarrow\!\!\bullet$ outputs $\ell_i(m^i)$ and a description of $\mathcal{F}_q$ at the E-interface. Upon receiving a description of $F \in \mathcal{F}_q$ at the E-interface, $\longrightarrow\!\!\bullet$ evaluates the transformation F on the plaintexts and outputs the result at the B-interface. If the $\perp$-converter is attached to the E-interface, $\longrightarrow\!\!\bullet$ immediately forwards each input m_i from the A- to the B-interface.

As an example, we describe the XOR-malleable confidential channel and sketch the proof that the one-time pad constructs this channel from an insecure channel and a shared secret key.[3]

Example 5 (The XOR-malleable channel). The channel $-\!\!\oplus\!\!\rightarrow\!\!\bullet$ behaves as follows. Upon the ith input $m_i \in \mathcal{M}$ at the A-interface, leak the length $|m_i|$ at the E-interface. Upon the jth input $\delta_j \in \{0,1\}^n$ at the E-interface (after i inputs at the A-interface), output $m'_j := F_{i,j,\delta}(m)$ at the B-interface.
We use the following simulator σ to prove that the one-time pad indeed constructs $-\!\!\oplus\!\!\rightarrow\!\!\bullet$:

- Upon a message $l_i \in \{1, \ldots, n\}$ at the inner interface (i.e., from $-\!\!\oplus\!\!\rightarrow\!\!\bullet$), output a uniformly random l_i-bit string $\tilde{c}_i$ as the transmitted ciphertext at the outer interface.

[3] For simplicity, we only consider the case $i > j$. For the general case, cf. [25, Sect. 6.1].

– Upon a message $\tilde{c}'_j$ at the outer interface,
  - if $j > i$, input $\delta_j = 0^{|m_j|}$ at $\multimap\!\!\oplus\!\!\bullet$,
  - if $j \leq i$ and $|\tilde{c}'_j| \geq |\tilde{c}_j|$, input $\delta_j = \tilde{c}_j|_{|\tilde{c}'_j|} \oplus \tilde{c}'_j$ at $\multimap\!\!\oplus\!\!\bullet$,
  - else, input $\delta_j = (\tilde{c}_j \oplus \tilde{c}'_j)|0^{|\tilde{c}'_j|-|\tilde{c}_j|}$ at $\multimap\!\!\oplus\!\!\bullet$.

The simulator σ is perfect, i.e., $\Delta^{\mathbf{D}}(\mathsf{otp\text{-}enc}^A\mathsf{otp\text{-}dec}^B(\longrightarrow \| \bullet\!\!=\!\!\bullet), \sigma^E(\multimap\!\!\oplus\!\!\bullet)) = 0$ for all distinguishers $\mathbf{D}$:

– On input the ith message m_i at the A-interface, in both cases a $|m_i|$-bit uniformly random string is output at the E-interface (generated either by $\mathsf{otp\text{-}enc}$ using the key or by σ).
– On input the jth message c'_j at the E-interface, the message output at the B interface also has the same distribution in both cases (by construction of σ; this is a simple check for each of the cases). ⧫

Consequently, the one-time pad constructs from the resources $\bullet\!\!=\!\!\bullet$ and $\longrightarrow$ the channel $\multimap\!\!\oplus\!\!\bullet$. This channel is confidential according to Definition 4, the simulator assumed in the definition is trivial (both $\bullet\!\!\longrightarrow\!\!\bullet$ and $\multimap\!\!\oplus\!\!\bullet$ leak exactly the length of the message).

4 Relation to Game-Based Security Definitions

In game-based security definitions for encryption schemes, the attacker has access to oracles for encrypting plaintext messages and decrypting or checking the correctness of ciphertexts, sometimes with additional constraints on the number or order of queries. The attacker's goal is either to generate a ciphertext that satisfies a certain condition, or to distinguish two cases in which it is provided with different sets of oracles. For many of these notions, it is not clear which guarantees the proven schemes provide when the ciphertexts are transmitted over a certain type of network.

In contrast, a constructive security statement makes these guarantees explicit: The confidentiality and integrity guarantees appear as the leakage functions and plaintext transformations of the constructed channel. In this section, we analyze the semantics of game-based notions from the literature by proving the (in)equivalence with corresponding constructive notions.

4.1 Goals and Attack Models

Security properties defined using games are often characterized by a *goal* and an *attack model*. The goal is essentially specified by the winning condition (the monotone output switches to 1), and the attack model is characterized by the "oracle queries" the adversary has at its disposal.

The attack model roughly corresponds to adversarial access to the "real resources" used by the protocol in constructive security statements. The more capabilities the game provides, the weaker the security modeled by the real resources, and the stronger the requirements for the protocol. Roughly, the idea

of a chosen plaintext attack corresponds to the real resource being an authenticated channel, and a chosen ciphertext attack corresponds to the real resource being an insecure channel. The goal of a game corresponds to the attributes of the constructed resource. For instance, the IND-type of games are often connected with confidentiality, whereas NM (non-malleability) and INT (integrity) are integrity guarantees.

4.2 Indistinguishability of Ciphertexts

The standard security notions for confidentiality are IND-CPA and IND-CCA, i.e., indistinguishability (of ciphertexts) under chosen-plaintext and chosen-ciphertext attack, respectively. Several variants appear in the literature; in all variants, a bit $b \in \{0,1\}$ is chosen uniformly at random, and, depending on the variant, the adversary has access to one of the following settings of oracles:

- multiple queries at a "real-or-random" oracle where, in each query, the adversary inputs a plaintext m_0, the game chooses m_1 with $|m_0| = |m_1|$ uniformly at random, and returns an encryption of m_b;
- multiple queries at a "left-or-right" oracle where the adversary inputs two messages m_0 and m_1 with $|m_0| = |m_1|$ and obtains an encryption of m_b;
- multiple queries at an "encryption" oracle where, on input m, the adversary obtains an encryption of m, as well as one "real-or-random" query;
- multiple "encryption" queries and one "left-or-right" query.

Finally, the adversary has to guess the bit b (with probability non-negligibly different from $1/2$). It turns out that, for any encryption scheme, the advantages that can be achieved in the above games are related by a factor that is either a constant or linear in the number of queries [3].

IND-CPA. The term IND-CPA usually refers to a game where the adversary has access to the oracles described in one of the four settings above. While these settings correspond to assuming that the ciphertexts are transmitted via authenticated channels (and cannot be changed during the transmission), in several practical protocols such as SSL/TLS, the ciphertexts can actually be changed during the transmission. Indeed, as confidentiality in the sense of Definition 4 is defined by restricting only the adversarial interface (the output at the receiver's interface is ignored), one may hope that IND-CPA security will still imply this weak form of confidentiality in this setting. The following example shows that this is not the case.

Consider an encryption scheme where a certain ciphertext $\bar{c} \in \mathcal{C}$ is never used, and append in the encryption to each ciphertexts a perfectly hiding commitment on the plaintext. In particular, expand the secret key using a PRG, use the first part as key for the encryption and the remainder as randomness in the commitment. Also, modify the decryption to output the initial secret key if it receives the special ciphertext $\bar{c}$. As the decryption algorithm does not appear in the IND-CPA game and the erroneous decryption does not hurt correctness

(as $\bar{c}$ is never used), the modified scheme is IND-CPA secure. However, for any confidential channel, it is easy to construct a distinguisher that differentiates between the real and the ideal setting (input a message $m \in \mathcal{M}$ at A, input $\bar{c}$ at E, interpret the output at B as the secret key, expand by the PRG, and decrypt the output at E. If this decrypts to m and the decommitment was correct then say 0, otherwise say 1).

IND-CCA. In the IND-CCA game, the adversary is, in addition to one type of oracles of the IND-CPA game, given access to a decryption oracle where it can query ciphertexts that are different from those he obtained from the encryption oracle.[4] While IND-CCA is considered the standard notion for confidentiality in settings where the adversary can modify ciphertexts, it differs considerably from the notion implied by Definition 4. In particular:

1. IND-CCA is artificially strict: A scheme that allows "obvious" modifications of ciphertexts (e.g., appending bits that are ignored) is considered insecure.
2. The definition of IND-CCA already implies strong integrity guarantees.
3. These integrity guarantees seem artificial for symmetric encryption.

These issues are explained further in the following paragraphs.

Replayable CCA. Several authors [1, 11, 18, 19, 33] have noticed that IND-CCA is artificially strict in the sense that the decryption oracle will decrypt any ciphertext except for the exact challenge ciphertext. Schemes that allow for "obvious" ciphertext modifications are not IND-CCA secure, the typical separating example being an (otherwise IND-CCA secure) encryption scheme where the encryption always appends a single bit to the ciphertext, and this bit is ignored during decryption. While this modification does not hurt the security guarantees in any meaningful way, the resulting scheme is not IND-CCA secure.

In [11], several variants of "replayable" CCA security are analyzed.[5] In these games, not only the exact challenge ciphertext is disallowed in decryption queries, but also "related" ciphertexts. Intuitively, this means that encryption schemes may allow certain modifications to ciphertexts that do not change the result of the decryption. In more detail, the notions considered in [11] are:

- IND-RCCA, or "replayable CCA": any ciphertext that decrypts to one of the plaintexts issued to the encryption oracle is disallowed;
- IND-sd-RCCA, or "secretly detectable RCCA": intuitively, the receiver can detect whether an adversarially generated ciphertext was generated as a "modification" of an honestly generated one, or whether it is "independent" of all honestly generated ones, these "modified" ciphertexts are disallowed;
- IND-pd-RCCA, or "publicly detectable RCCA": the above distinction can be done publicly, i.e., without knowledge of the secret key.

[4] The reason for the latter restriction is that if the adversary were allowed to decrypt the challenge, winning the game would become trivial.

[5] Their original notions regard public-key schemes, but the extensions to symmetric schemes are also described.

The exact formalization is technically involved; for details, we refer to [11].

With respect to achieving secure communication, the guarantees provided by IND-CCA and IND-sd-RCCA secure schemes are indeed equivalent, which can be formalized via bisimulation. Intuitively, the simulator for the IND-sd-RCCA scheme can use the assumed detectability to decide whether a given ciphertext should be considered a replay.

Strong Integrity. An IND-sd-RCCA secure encryption scheme achieves a strong notion of integrity: The remaining malleability is described by the integrity specification $\mathcal{F}_{\text{NM}}$ with the set $\{f_{\bar{m}} : \mathcal{M} \to \mathcal{M}, m \to \bar{m}\}_{\bar{m}\in\mathcal{M}}$ of transformations, where NM refers to "non-malleable." The proof of the following theorem is deferred to the full version of this paper.

Theorem 2 (Informal). *Let* $(\mathsf{enc}, \mathsf{dec})$ *be a symmetric encryption protocol. If the protocol is* ε*-IND-sd-RCCA secure, then it constructs an* $\mathcal{F}_{\text{NM}}$*-malleable confidential channel from an insecure channel and a secret key within* ε*.*

Conversely, if the protocol constructs an $\mathcal{F}_{\text{NM}}$*-malleable confidential channel from an insecure channel and a secret key within* ε *(for distinguishers that issue at most* q *queries and with a special type of simulator) then it is* $(q^2+1)\varepsilon$*-IND-sd-RCCA secure (with respect to the class of adversaries that issue at most* q *queries). For large message spaces, the special type of simulator is general.*[6]

Unnatural Malleability. IND-CCA is not a natural security requirement for symmetric encryption: The adversary may generate valid ciphertexts for arbitrary plaintexts (but only independently of honestly sent messages). Realistic symmetric encryption schemes are either malleable (such as the one-time pad or CBC) or, if they are non-malleable, they will actually already implement the fully secure channel (such as authenticated encryption). Here, it becomes apparent that IND-CCA has evolved as a notion for public-key schemes, where the adversary knows the encryption key and can encrypt arbitrary messages.

4.3 Specific Variants of Integrity

Games that are used to characterize integrity properties express impossibilities (for the adversary) to generate ciphertexts that satisfy certain conditions. In constructive cryptography, integrity guarantees are expressed explicitly by specifying the set of transformations that model the capabilities of the adversary. The correspondence between these two paradigms is as follows: A scheme is secure according to a game if and only if it implements a channel that allows no transformations that contradict the game; the potential probability in winning the game translates into a distinguishing advantage in the constructive security statement.

[6] If the distinction between "modified" and "independent" ciphertexts can be performed without the key, then the condition on the size of the message space is not needed. If we assume that the distinction is perfect, the factor q^2+1 reduces to 1.

NM-CCA. The notion of non-malleable encryption has been introduced in [12] in the context on public-key schemes. Intuitively, no attacker (even given honestly generated ciphertexts) may be able to generate a ciphertext whose decryption relates to "honestly encrypted" messages in a meaningful way. NM-CCA is equivalent to IND-CCA [12]; this extends to the RCCA notions [11]. Consequently, these notions also correspond to $\mathcal{F}_{\text{NM}}$-malleable communication.

INT-CTXT. Integrity of ciphertexts has been introduced in [5, 6] and formalizes that the adversary cannot produce *any* fresh valid ciphertext. In more detail, an encryption scheme is said to achieve INT-CTXT security if no adversary with access to an encryption oracle can generate a valid ciphertext that is different from all ciphertexts obtained from the oracle. Here, "valid" means that the decryption outputs a message (not an error symbol). Note that existential unforgeability [17] and ciphertext unforgeability [18] are similar: The differences are, for example, that the definition from [5, 6] allows multiple queries to the challenge oracle, whereas [17] allows only one.

A symmetric encryption protocol that achieves confidentiality and is additionally INT-CTXT secure constructs a fully secure channel from an insecure channel. Yet, INT-CTXT, as IND-CCA, is artificially strict concerning modifications of ciphertexts. We describe a relaxation of INT-CTXT which is constructed analogously to IND-sd-RCCA. In particular, we also require the existence of a secretly (i.e., given the secret key) computable relation, called $\equiv_\kappa$, on $\mathcal{C}$ with the same properties as for IND-sd-RCCA; this relation formalizes the receiver's ability to distinguish "modified" and "independent" ciphertexts generated by the adversary.

We define INT-sd-CTXT security by changing the INT-CTXT game as follows: The adversary wins only if $\text{dec}(\kappa, c') \neq \boxslash$ and $\forall\ r \leq i : c' \not\equiv_\kappa c_r$ for all honestly generated c_r. Note that we also have to change the output of the oracle in the case that $c'_j \equiv_\kappa c_r$ holds (for some r) to be m_r. The proof of the following theorem is deferred to the full version of this paper.

Theorem 3 (Informal). *Let* $(\mathsf{enc}, \mathsf{dec})$ *be a symmetric encryption protocol that constructs a confidential channel from an insecure channel and a secret key within* ε_1*. If the protocol is* ε_2*-INT-sd-CTXT secure, then it constructs a secure channel from an insecure channel and a secret key within* $\varepsilon_1 + \varepsilon_2$*. Conversely, if the protocol constructs the secure channel within* ε *for distinguishers in* $\mathcal{D}_q$*, then it is* $(q^2 + 2)\varepsilon$*-INT-sd-CTXT secure with respect to* $\mathcal{D}_q$*.*[7]

INT-PTXT. Integrity of plaintexts has been defined in [5, 6] and is weaker than INT-sd-CTXT. The adversary is also given access to an encryption oracle, but to win the game, it has to fabricate a ciphertext that decrypts to a plaintext that has not been queried at the encryption oracle before. This notion is weaker than INT-sd-CTXT in the sense that the adversary may still be able to generate

[7] The factor $q^2 + 2$ appears for the same technical reasons as for IND-sd-RCCA.

a ciphertext that decrypts to plaintext that was queried at the encryption oracle but cannot be detected to be a modification of one particular honestly generated ciphertext (even if all ciphertexts are delivered). This weakens the guarantees in two aspects: First, the adversary can replay messages undetectably, and second, the adversary may fabricate messages that decrypt to any one of the previous messages with some probability *that may even depend on the plaintexts*. Consequently, if the adversary is able to determine which of the original plaintexts has been received, he will potentially obtain information about some transmitted plaintext.

An integrity specification is *value-preserving* if all transformations $F_\alpha : \mathcal{M}^* \times \mathcal{M}^* \to \bar{\mathcal{M}}$ have the property that the output message is either one of the input messages or $\boxslash$, but any one of these may appear with some probability (which may even depend on the plaintexts). The proof of the following theorem is deferred to the full version of this paper.

Theorem 4. *Let* $(\mathsf{enc}, \mathsf{dec})$ *be a symmetric encryption protocol that constructs a confidential channel from an insecure channel and a secret key within* ε_1*. If the protocol is* ε_2*-INT-PTXT secure, then it constructs an* $\mathcal{F}_{\text{VP}}$*-malleable confidential channel within* $\varepsilon_1+\varepsilon_2$*, with* $\mathcal{F}_{\text{VP}}$ *being value-preserving. Conversely, if the protocol constructs an* $\mathcal{F}_{\text{VP}}$*-malleable confidential channel within* ε_1 *such that* $\mathcal{F}_{\text{VP}}$ *is value-preserving, then it is* ε_1*-INT-PTXT secure.*

Namprempre [27] introduces a related but stricter notion called SINT-PTXT, which prohibits replaying messages arbitrarily. There, the adversary also wins the game if it generates ciphertexts such that the decryption outputs any plaintext *more often* than it was queried at the encryption before. Consequently, SINT-PTXT corresponds to a channel with this bounded type of replay.

Fixing the definition from [5, 6]. In the original game, the output of the verification oracle is one bit indicating whether the decrypted plaintext is valid. This renders the notion too weak: If (via a higher-level protocol), the adversary learns *which* of the valid plaintexts has been obtained by decrypting (this probability may depend on secret values), this is not captured. Hence, this notion cannot guarantee composability. A slight modification to the game fixes this issue: The verification oracle returns the decrypted message (instead of the single bit). The following (artificial) encryption scheme exemplifies the weakness.

Example 6. Consider a scheme $(\mathsf{enc}, \mathsf{dec})$ secure according to the stricter notion. Change the decryption such that for (n, c_0, c_1) with $\mathsf{dec}_\kappa(c_b) \neq \perp$, $b \in \{0, 1\}$, the output is $\mathsf{dec}_\kappa(c_{\kappa_n})$ (with κ_n the nth bit of κ). ♦

The change does not affect the security with respect to the notion of [5, 6]: The output of the oracle on (n, c_0, c_1) can be easily computed from the output on c_0 and c_1. In contrast, in the strengthened game, such queries reveal the secret key.

Plaintext Uncertainty. This notion from [13] attempts to capture that the adversary cannot "control" the result of a forgery. While the description is rather

informal, it captures that the decrypted message contains a certain amount of entropy (for each message, the probability that this message is obtained by decrypting is small). While this is hard to achieve at least for multiple decryptions—the only entropy in the (otherwise deterministic) decryption is "fresh" key material—the computational (pseudo-entropy) version might prove useful in applications.

The corresponding integrity specification is the set of transformations that have at least a certain min-entropy, meaning that for each input m and transformation F, the min-entropy of the random variable $F(m)$ is larger than some bound. Computational indistinguishability from such a channel means that the output at the receiver's interface has a certain pseudo-entropy.

Known-Plaintext Forgery. This notion from [13] is intended to capture that the adversary providing a forged ciphertext *can* predict the changes to the transmitted message. The (informal) description in [13] states that the adversary could have computed the outcome with overwhelming probability (this can be formalized by means of an extractor). In the language of integrity specifications, this means that all transformations in $\mathcal{F}$ are deterministic (and efficiently computable). Properties of this type can indeed be helpful, as can be seen in the proof of the soundness of Authenticate-then-Encrypt in [25].

4.4 Combining Notions of Confidentiality and Integrity

Traditionally, security requirements for schemes for protecting communication are expressed as a combination of separate properties for confidentiality and integrity [5, 7, 9, 13, 17, 27]. Such a combination, however, does not necessarily achieve the expected guarantees.

We revisit an example from [18] (modified in [25]): The composition of a tailor-made encryption scheme with a strongly unforgeable MAC. Briefly, the encryption first encodes each bit of the plaintext as two bits, such that the probability whether flipping one of these two bits has an effect depends on the original value (i.e., $0 \mapsto 00$, 01, or 10; $1 \mapsto 11$), and encrypts this expanded string using a one-time pad. Hence, if one encrypts an authenticated message, the probability that flipping a ciphertext bit changes the contained message—and the MAC verification fails—with a probability that depends on the original plaintext value. The resulting scheme achieves both confidentiality (by the one-time pad) and integrity (in the sense of INT-PTXT, by the unforgeability of the MAC), but the different success probabilities for the MAC verification leak information about the message, which is often described as a breach of confidentiality [18].

The described scheme implements a confidential $\mathcal{F}_{\text{VP}}$-malleable channel, where $\mathcal{F}_{\text{VP}}$ is value-preserving as described in Sect. 4.3: The weakness of this scheme is not a deficiency of confidentiality, but it only achieves a weak notion of integrity. Note that, in terms of integrity, INT-PTXT is equivalent to WUF-CMA[8], which is sufficient to construct an authenticated channel (where the adversary can only

[8] Weak unforgeability: Given an oracle for generating tags, it is infeasible for the adversary to generate a tag for a message that has not been queried at the oracle.

forward or delete messages) from an insecure channel. Indeed, for channels that are not confidential, the integrity guarantees specified by $\mathcal{F}_{\text{VP}}$ are equivalent to those of an authenticated channel: A simulator that knows the plaintext messages can sample according to distributions that depend on these messages. This equivalence does not hold if the considered channels are confidential.

4.5 A Critique of Game-Based Security Notions

Starting from [14], the major part of research on the security of encryption schemes has been pursued in game-based models. There, however, it is often not immediately clear which assumptions and guarantees are encoded by the oracle queries and winning conditions of games. For instance, which of the a priori different types of IND-CPA security described in Sect. 4.2 captures confidentiality "best" (and why)? This lack of semantics abets the prevalence of security notions that do not capture the security requirements *exactly* (see Sect. 4.2 and 4.3).

A further issue with game-based notions is that seemingly innocent changes may have a significant impact on the security guarantees. The security notion *indistinguishability from random bits* was introduced in [30] and is similar to IND-CPA. Yet, instead of an encryption of a random message, the game returns a uniformly random string of appropriate length. The way this length is chosen, however, is crucial: In the original definition, this is determined by a function of the length of the queried message. If this choice is changed (as done, for example, in [15]) to the length of an encryption of the queried message, this allows to leak information about the plaintext via the length of the ciphertext! A further example is the weakness of the INT-PTXT notion described in Sect. 4.3.

Moreover, several attack models in the definitions described in the literature seem inappropriate for practical applications. One example is IND-CCA1[9], where the receiver stops decrypting adversarially generated ciphertexts after the first message has been sent honestly. Also, certain terms such as NM-CPA are actually misleading: An attack exploiting the malleability of an encryption scheme is necessarily mounted by injecting or replacing ciphertexts. A more appropriate correspondence for this type of notion is a CCA attack on a single-use channel.

5 Conclusion

We have defined and analyzed confidentiality and integrity notions for symmetric encryption schemes using the paradigm of constructive cryptography. The resulting security definitions are composable and have clear semantics: The guarantees of a cryptographic protocol appear explicitly in the description of the constructed resource. We have shown how existing game-based notions can be translated into guarantees in this setting, which makes their semantics explicit. Additionally, this analysis has uncovered a weakness in the notion INT-PTXT, and it has shown that INT-CTXT and IND-CCA are artificially strict.

[9] In the CCA1 game, the adversary looses access to the decryption oracle after the first call to the challenge oracle. This corresponds to the situation where the receiver only decrypts messages until the first message has been generated by the sender.

Acknowledgments. We thank Kenny Paterson for fruitful discussions and the anonymous reviewers for their very helpful comments and suggestions. The work was supported by the Swiss National Science Foundation (SNF), project no. 200020-132794.

References

1. An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer-Verlag (2002)
2. Backes, M., Pfitzmann, B., Waidner, M.: The reactive simulatability (RSIM) framework for asynchronous systems. Information and Computation 205(12), 1685–1720 (December 2007)
3. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: Proceedings of the 38th Symposium on Foundations of Computer Science. pp. 394–403. IEEE (1997)
4. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 26–45. Springer (1998)
5. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer-Verlag (2000)
6. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. Journal of Cryptology 21(4), 469–491 (October 2008)
7. Bellare, M., Rogaway, P.: Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 317–330. Springer-Verlag (2000)
8. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science. pp. 136–145. IEEE (2001)
9. Canetti, R., Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 453–474. Springer-Verlag (2001)
10. Canetti, R., Krawczyk, H.: Universally composable notions of key exchange and secure channels. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 3027, pp. 337–351. Springer-Verlag (2002)
11. Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing chosen-ciphertext security. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 565–582. Springer-Verlag (2003)
12. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography. SIAM Journal on Computing 30(2), 391–437 (2000)
13. Gligor, V.D., Donescu, P., Katz, J.: On message integrity in symmetric encryption (February 2002)
14. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
15. Iwata, T.: New blockcipher modes of operation with beyond the birthday bound security. In: Robshaw, M. (ed.) FSE 2006. LNCS, vol. 4047, pp. 310–327. Springer-Verlag (2006)

16. Katz, J., Yung, M.: Complete characterization of security notions for probabilistic private-key encryption. In: Proceedings of the thirty-second annual ACM symposium on Theory of computing. pp. 245–254. ACM (2000)
17. Katz, J., Yung, M.: Unforgeable encryption and chosen ciphertext secure modes of operation. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 284–299. Springer-Verlag (2000)
18. Krawczyk, H.: The order of encryption and authentication for protecting communications. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 310–331. Springer-Verlag (2001)
19. Krohn, M.: On the definitions of cryptographic security: Chosen-ciphertext attack revisited. Senior Thesis, Harvard University (1999)
20. Maurer, U.: Indistinguishability of random systems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 110–132. Springer-Verlag (2002)
21. Maurer, U.: Constructive cryptography—A primer. In: Sion, R., Curtmola, R., Dietrich, S., Kiayias, A., Miret, J.M., Sako, K., Sebé, F. (eds.) FCDS 2010. LNCS, vol. 6054, p. 1. Springer-Verlag (2010)
22. Maurer, U.: Constructive cryptography: A new paradigm for security definitions and proofs. In: Mödersheim, S., Palamidessi, C. (eds.) TOSCA 2011. LNCS, Springer-Verlag (2011)
23. Maurer, U., Renner, R.: Abstract cryptography. In: Innovations in Computer Science. Tsinghua University Press (2011)
24. Maurer, U., Schmid, P.: A calculus for security bootstrapping in distributed systems. Journal of Computer Security 4(1), 55–80 (1996)
25. Maurer, U., Tackmann, B.: On the soundness of Authenticate-then-Encrypt: Formalizing the malleability of symmetric encryption. In: ACM Conference on Computer and Communications Security. ACM (2010)
26. Micali, S., Rogaway, P.: Secure computation. In: Feigenbaum, J. (ed.) CRYPTO 91. LNCS, vol. 576, pp. 392–404. Springer-Verlag (1991)
27. Namprempre, C.: Secure channels based on authenticated encryption schemes: A simple characterization. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 111–118. Springer-Verlag (2002)
28. Paterson, K.G., Ristenpart, T., Shrimpton, T.: Tag size does matter: Attacks and proofs for the TLS record protocol. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 372–389. Springer-Verlag (2011)
29. Pfitzmann, B., Waidner, M.: A model for asynchronous reactive systems and its application to secure message transmission. In: Proceedings of the 2001 IEEE Symposium on Security and Privacy. pp. 184–200. IEEE (2001)
30. Rogaway, P., Bellare, M., Black, J.: OCB: A block-cipher mode of operation for efficient symmetric encryption. ACM Transactions on Information and System Security (TISSEC) 6(3), 365–403 (2003)
31. Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer-Verlag (2006)
32. Rüedlinger, A.: Restricted Types of Malleability in Encryption Schemes. Master's thesis, ETH Zürich (2011)
33. Shoup, V.: A proposal for an ISO standard for public key encryption. Cryptology ePrint Archive, Report 2001/112 (2001)
34. Shrimpton, T.: A characterization of authenticated-encryption as a form of chosen-ciphertext security. Cryptology ePrint Archive, Report 2004/272 (2004)