On the Soundness of Authenticate-then-Encrypt

Formalizing the Malleability of Symmetric Encryption

Ueli Maurer
ETH Zurich, Department of Computer Science
CH-8092 Zurich, Switzerland
maurer@inf.ethz.ch

Björn Tackmann
ETH Zurich, Department of Computer Science
CH-8092 Zurich, Switzerland
bjoern.tackmann@inf.ethz.ch

ABSTRACT

A communication channel from an honest sender A to an honest receiver B can be described as a system with three interfaces labeled A, B, and E (the adversary), respectively, where the security properties of the channel are characterized by the capabilities provided at the E-interface.

A security mechanism, such as encryption or a message authentication code (MAC), can be seen as the transformation of a certain type of channel into a stronger type of channel, where the term "transformation" refers to a natural simulation-based definition. For example, the main purpose of a MAC can be regarded as transforming an insecure into an authenticated channel, and encryption then corresponds to transforming an authenticated into a fully secure channel; this is the well-known Encrypt-then-Authenticate (EtA) paradigm.

In the dual paradigm, Authenticate-then-Encrypt (AtE), encryption first transforms an insecure into a confidential channel, and a MAC transforms this into a secure channel. As pointed out by Bellare and Namprempre [5], and Krawczyk [17], there are encryption schemes for which AtE does not achieve the expected guarantees.

We highlight two reasons for investigating nevertheless AtE as a general paradigm: First, this calls for a definition of confidentiality; what separates a confidential from a secure channel is its (potential) malleability. We propose the first systematic analysis of malleability for symmetric encryption, which, in particular, allows us to state a generic condition on encryption schemes to be sufficient for AtE. Second, AtE is used in practice, for example in TLS. We show that the schemes used in TLS (stream ciphers and CBC encryption) satisfy the condition. This is consistent with Krawczyk's results on similar instantiations of AtE in game-based models.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and Protection*

General Terms

Security, Theory

CCS'10, October 4–8, 2010, Chicago, Illinois, USA.

1. INTRODUCTION

Many day-to-day applications such as online banking or remote file access assume that the communication between the involved computers is secure, where the term *secure* covers two major aspects: First, the transfer must be *confidential* in the sense that it does not cause any "harmful" information leakage. Second, the messages must be received *authentically*, which means that the receiver only accepts messages that originate from the supposed sender. In practice, however, the communication channels are insecure. This paper studies the problem of achieving secure communication over insecure channels, assuming that the sender and the receiver already share a secret key.

1.1 Modeling Secure Communication

We model the communication between two honest entities as a channel, that is, as a system with three interfaces that takes as input messages from the sender and provides as output (potentially different) messages to the receiver. The security properties of the channel are modeled by the capabilities provided to a third (hypothetical) entity: the adversary. We consider the following types of channels, using the notation introduced in [23].

- $\longrightarrow$ An *insecure channel* leaks the transferred messages, and allows the adversary to change the messages before delivering them to B.
- $\bullet\!\!\longrightarrow$ An *authenticated channel* leaks the transferred messages, but the adversary may only forward the messages or completely abort the channel.
- $\bullet\!\!\longrightarrow\!\!\bullet$ A *secure channel* hides the messages[1], and only allows forwarding the messages or aborting the channel.
- $\longrightarrow\!\!\bullet$ A *confidential channel* also hides the messages, but does not guarantee integrity of the messages.

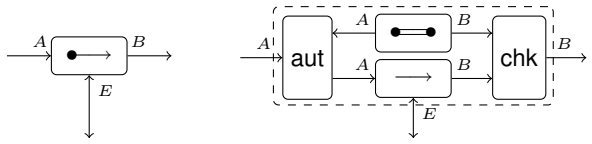
The interpretation of the symbol "$\bullet$" is that the marked interface of the channel is exclusive to the connected party. A channel with exclusive access for the receiver is confidential (no other party learns the message), and a channel where the same holds for the sender is authenticated (no other party can input messages).

Additionally, we use a system $\bullet\!\!=\!\!\bullet$ that outputs a random key at both interfaces A and B. This system models the shared secret key that is assumed by (symmetric) encryption and authentication schemes and could result from a key agreement protocol. The E-interface of $\bullet\!\!=\!\!\bullet$ is inactive and will often be discarded.

1.2 Constructing Secure Channels

Following the paradigm of constructive cryptography [19, 22], security mechanisms such as encryption or MAC schemes are interpreted as transformations from one type of channel into a "stronger"

[1]Except for the length of the messages.



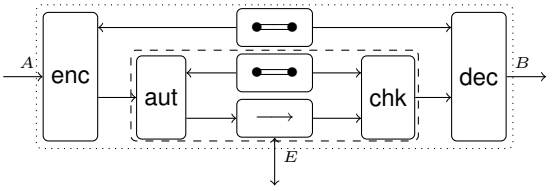
(a) The authenticated channel $\bullet\!\!\longrightarrow$.

(b) Construction of the channel $\bullet\!\!\longrightarrow$, denoted as $\mathsf{aut}^A\mathsf{chk}^B(\longrightarrow \| \bullet\!\!=\!\!\bullet)$ below.

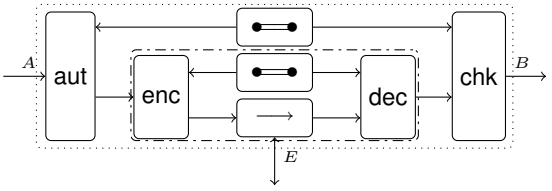
Figure 1: The authentication protocol $(\mathsf{aut}, \mathsf{chk})$ transforms the insecure channel $\longrightarrow$ into the authenticated channel $\bullet\!\!\longrightarrow$.

type of channel. Such a transformation is illustrated in Figure 1: The authenticated channel $\bullet\!\!\longrightarrow$ is constructed (see Figure 1(b)) from the insecure channel $\longrightarrow$ and the shared secret key $\bullet\!\!=\!\!\bullet$ by the authentication protocol $(\mathsf{aut}, \mathsf{chk})$. The systems aut and chk interact with the respective interfaces of $\longrightarrow$ and $\bullet\!\!=\!\!\bullet$, and the "outside" interfaces of aut and chk become the interfaces of the newly constructed (dashed) system. We say that the protocol $(\mathsf{aut}, \mathsf{chk})$ transforms the insecure channel $\longrightarrow$ into the authenticated channel $\bullet\!\!\longrightarrow$ by use of the shared key $\bullet\!\!=\!\!\bullet$, if the behavior of the systems in Figures 1(a) and 1(b) is essentially the same, where the exact notion of comparing the behavior of systems is defined in a simulation-based sense made precise in Section 2.

Regarding protocols as transformations of channels is dual to their common interpretation as transformations of messages. This duality becomes evident in the analysis of protocols for secure communication. Given an encryption scheme for confidentiality and a MAC for authenticity, there are two natural approaches to constructing secure channels. The first approach is depicted in Figure 2(a): One first applies the MAC to the insecure channel $\longrightarrow$ and a shared secret key $\bullet\!\!=\!\!\bullet$ to construct an authenticated channel, and then uses the encryption scheme to obtain a secure channel. As in Figure 1(b), the composition of systems grouped by the dashed box behaves as an authenticated channel $\bullet\!\!\longrightarrow$. An encryption scheme guarantees that the (dotted) system constructed from the authenticated channel and a shared key $\bullet\!\!=\!\!\bullet$ is a secure channel. In the literature [5, 17], this transformation is referred to as *Encrypt-then-Authenticate* (EtA), since the first operation applied to the plaintext input at the outside A-interface is the encryption. From our perspective of channel transformations, however, the first mechanism applied to the insecure channel $\longrightarrow$ is the authentication. For the sake of consistency with previous literature, we will maintain the term EtA for this transformation, keeping in mind that the permuted appearance of the terms "encryption" and "authentication" is an effect of the paradigm shift underlying our analysis.



(a) Encrypt-then-Authenticate, the structure is described as $\mathsf{enc}^A\mathsf{dec}^B(\mathsf{aut}^A\mathsf{chk}^B(\longrightarrow \| \bullet\!\!=\!\!\bullet)\| \bullet\!\!=\!\!\bullet)$ below.



(b) Authenticate-then-Encrypt, the structure is described as $\mathsf{aut}^A\mathsf{chk}^B(\mathsf{enc}^A\mathsf{dec}^B(\longrightarrow \| \bullet\!\!=\!\!\bullet)\| \bullet\!\!=\!\!\bullet)$ below.

Figure 2: Generic constructions of secure channels using encryption $(\mathsf{enc}, \mathsf{dec})$ and authentication $(\mathsf{aut}, \mathsf{chk})$.

The paradigm dual to EtA is called *Authenticate-then-Encrypt* (AtE), and uses the encryption to transform the insecure channel $\longrightarrow$ into a confidential channel $\longrightarrow\!\!\bullet$ indicated by the dashed box in Figure 2(b). This channel is transformed into a secure channel by a MAC. While EtA is secure under widely-used assumptions, the case of AtE is substantially more involved.

In both Figures 2(a) and 2(b), the keys for the encryption and MAC schemes originate from two distinct systems $\bullet\!\!=\!\!\bullet$. This represents the fact that the keys used in the schemes are independent. In practice, both keys can be derived from a single one using a pseudo-random generator (PRG).

In a third approach, called Encrypt-and-Authenticate (E&A), the MAC is also applied to the plaintext, but the generated tag is not encrypted. E&A is also not secure under the usual assumptions. Additionally, several monolithic *authenticated encryption* schemes that perform the complete transformation in a single step are described in the literature (e.g., [27, 29]).

1.3 Generic Security of AtE

The standard notion for the security of symmetric encryption schemes is chosen-plaintext security. Yet, such encryption schemes are not generally sufficient to achieve authenticated encryption via the AtE composition even with strong MACs [5, 17]. As the one-time pad and CBC encryption—without achieving a stricter standard notion of security—are indeed sufficient [17], the usual interpretation of these results is that AtE is not "generically secure".

From the constructive perspective, the insufficiency of chosen-plaintext secure encryption schemes translates into the statement that a general confidential channel cannot be transformed into a secure channel by a MAC. This is due to the unrestricted malleability of the encryption and motivates the explicit and general formalization of malleability in Section 4. This formalization enables us to state explicit conditions on encryption schemes, and to generically prove the security of AtE for all schemes that meet the conditions.

Our analysis suggests that AtE can be regarded as a sound transformation, but chosen-plaintext security is not the appropriate security definition for encryption schemes in this setting.

1.4 The SSL/TLS Protocol Suite

The most important application of the AtE paradigm is the widely deployed TLS protocol [12], which secures client-server connections in many current internet protocols. While, as shown in Sections 5 and 6, the AtE transformation is secure for the encryption schemes used in TLS, such an analysis only covers a small part of the full-fledged TLS protocol, which contains further subprotocols such as key exchange and session management. For instance, verbose error messages in CBC mode have rendered TLS 1.0 insecure [28], and the recent renegotiation attacks [25] exploit a vulnerability in the handling of keys.

1.5 Related Work

A major part of research on symmetric encryption and authentication schemes has been carried out using game-based models. The most widely-used definitions of confidentiality for symmetric

encryption (IND-CPA and IND-CCA) have been adapted from the corresponding public-key notions [2]. For authenticity, two slightly different notions of unforgeability (WUF-CMA and SUF-CMA[2]) have emerged as the standard notions [1, 4]. Schemes that protect both the confidentiality and the authenticity are called *authenticated encryption schemes*, and their security is defined by a combination of properties for confidentiality and integrity [5, 27].

Simulation-based definitions of secure communication have been given by Pfitzmann and Waidner [24] for Reactive Simulatability and by Canetti and Krawczyk [10] for Universal Composability. Surprisingly, the corresponding security proofs are performed in a single step and do not exploit the composability guaranteed by the respective frameworks. A "hybrid" approach is followed in the definition of [9]: authenticity is simulation-based, while confidentiality is game-based.

The paradigm of constructive cryptography has been explicitly introduced by Maurer [19]. The idea is to describe resources (such as channels) as systems, and to consider cryptographic protocols as transformations that construct "stronger" systems from "weaker" systems. The notion of a transformation can be formalized using the approach of Maurer and Renner [21]. In this work, we apply the constructive paradigm in the setting of secure communication, resulting in natural security definitions as well as specifications for the most common types of channels.

Notions of non-malleable encryption have first appeared for public-key schemes [3, 13], and have later been translated to the symmetric case [5]. Several related notions such as unforgeability of ciphertexts have been discussed [15, 16]. These notions can be considered as very restricted types of malleability and expressed using our more general approach.

The EtA transformation of an encryption and a MAC secure under the game-based standard notions is secure as an authenticated encryption, while the corresponding statement does not hold for AtE [5, 17]. In contrast, CBC encryption and stream ciphers [17] as well as nonce-based encryption schemes [26] are indeed sufficient. The focus of our work is different: Instead of proving the composition for each scheme individually, *we formulate generic conditions on the encryption schemes.* Moreover, our analysis of the concrete schemes is closer to TLS (compared to [17]) in two aspects: We do not impose an (unsuitable) size restriction on the MAC, and we take into account the padding for CBC encryption.

Ferguson and Schneier [14] compare EtA and AtE from a practical perspective. On the one hand, they argue that AtE is favorable for two reasons. First, the MAC is "protected" by the encryption, which makes attacking the authenticity of the combined scheme more difficult. Second, the authentication is applied to the plaintext, while in EtA, only the ciphertext is authenticated. On the other hand, they note that EtA is generically secure and more resilient to certain Denial-of-Service attacks. This paper provides further foundations for the comparison of the two paradigms.

1.6 Outline

The remainder of the paper is organized as follows. In Section 2, we apply the paradigm of constructive cryptography in the setting of secure communication and obtain a natural security definition along with a composition theorem. In Section 3, we describe the basic types of channels and show how encryption and authentication appear as transformations of these channels. We introduce the general model for malleability of encryption in Section 4, and in Section 5, we show that a certain restriction on the malleability is sufficient for the applicability of the encryption schemes in the AtE construction. In Section 6, we show that the schemes used in the TLS protocol comply with this restriction.

[2]The terms correspond to *weak unforgeability against chosen message attacks* and *strong unforgeability*, respectively. For further details, see Section 3.3.

2. PRELIMINARIES

2.1 Notation

For tuples of variables m_i, we often use the notation $m^i = (m_1, \ldots, m_i)$. The term *negligible* for functions $\nu : \mathbb{N} \to [0, 1]$ has its usual meaning: for $k \to \infty$, the function ν vanishes faster than the inverse of any polynomial.

The security definition used in this paper is based on comparing probabilities in random experiments that are defined by protocol executions. In general, for such a random experiment $\mathbf{G}$, the probability that the event E occurs is denoted by $\mathsf{P}^{\mathbf{G}}(E)$. In this context, we will often be interested in sequences of binary values $C_1, C_2, \ldots$; such a sequence is called *monotone* if $C_i = 1$ implies that $C_j = 1$ for all $j \geq i$. The *statistical distance* of two random variables A and B is denoted by $d(A, B)$.

2.2 Model of Protocol Execution

To analyze the security of a cryptographic protocol, we explicitly model the context in which the protocol is used. Both the protocol machines and the resources available to the protocols (such as communication channels or shared random keys) are described as (probabilistic) *discrete systems* that communicate by passing messages, where the term discrete refers to both the sets of messages and the time. A protocol execution is formalized as an interaction of these systems. The exact model of computation used to formalize the systems is not of interest; any formulation that is closed under composition will suffice. In particular, the formulations based on interactive Turing machines [7] or I/O automata [24] are valid instantiations. A more abstract formulation of discrete systems can be based on the random systems approach [18, 20], which exactly captures the relevant input/output behavior of the systems. Note that all these models allow to define families of systems indexed by a *security parameter* $k \in \mathbb{N}$, and by a system $\mathbf{S}$ we will implicitly refer to a family of systems $\{\mathbf{S}_k\}_{k\in\mathbb{N}}$. This allows one to formulate security statements in an asymptotic sense. In particular, all of the above models include a notion of *efficiency*, which is usually a variant of polynomial time.

We examine the security of protocols in the basic three-party setting: Two honest parties want to communicate securely in presence of an adversary. We generally distinguish between two types of systems: *Resources* provide two interfaces labeled A and B for the honest parties and one interface E for the adversary. *Converters*, such as protocol machines, have two distinct interfaces: They connect to resources via their *inner* interface and provide their *outer* interface to the environment. The set of all (efficient) converters is denoted by Σ. The composition of a resource $\mathbf{R}$ and a converter σ is indicated by the notation $\sigma^E\mathbf{R}$, where the identifier E means that the inner interface of σ is attached to the E-interface of the resource $\mathbf{R}$. The composed system exposes the A and B-interfaces of $\mathbf{R}$ and the outer interface of σ (as E-interface), so $\sigma^E\mathbf{R}$ is again a resource. A *protocol* is a pair of converters $\pi = (\pi_1, \pi_2)$ for the honest parties, and applying π to a resource $\mathbf{R}$ is defined as attaching the converters to the honest interfaces: $\pi_1^A\pi_2^B\mathbf{R}$.

If two resources $\mathbf{R}$ and $\mathbf{S}$ are used (mutually asynchronously) in parallel, this is denoted as $\mathbf{R} \,\|\, \mathbf{S}$. This system is called the *parallel composition* of $\mathbf{R}$ and $\mathbf{S}$, and is again a resource with interfaces A, B, and E. Each of these interfaces allows to explicitly access the corresponding interfaces of the two sub-systems $\mathbf{R}$ and $\mathbf{S}$. For example, in Figure 1, a shared key ⊷ and an insecure channel

$\longrightarrow$ are composed in parallel. The converters aut and chk connect with their inner interfaces to both $\bullet\!\!=\!\!\bullet$ and $\longrightarrow$. During the setup phase, both converters interact with $\bullet\!\!=\!\!\bullet$ to obtain the key, and later, they use the channel $\longrightarrow$ for the communication. Using the notation introduced above, we describe the setting in Figure 1(b) as $\mathsf{aut}^A\mathsf{chk}^B(\longrightarrow \| \bullet\!\!=\!\!\bullet)$.

2.3 Definition of Security

The definition of security used in this paper is derived from the paradigm of constructive cryptography [19]: Both the resources used by a protocol and the desired functionality are specified as systems, and a protocol is deemed secure if it constructs the functionality from the given resources. This paradigm is in sharp contrast to the widely used property-based notions, where security is characterized by properties that are defined by an adversary's inability to win a certain game.

Technically, the definition is based on the work of Maurer and Renner [21], and is similar in spirit to previous simulation-based definitions [7, 24]. In particular, it also involves a comparison of two different systems: The "real" system corresponds to the construction and is defined by connecting the protocol π to the honest interfaces of the resource $\mathbf{R}$. In the "ideal" system, the *ideal functionality* $\mathbf{S}$ describing the security goals is executed with a simulator σ connected to the E-interface. The purpose of σ is to convert the E-interface of $\mathbf{S}$ such that it resembles the corresponding interface of $\pi^A\pi^B\mathbf{R}$. (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.) If these two systems $\pi_1^A\pi_2^B\mathbf{R}$ and $\sigma^E\mathbf{S}$ behave equivalently, then the ideal system $\mathbf{S}$ can be safely replaced by the implementation $\pi_1^A\pi_2^B\mathbf{R}$. The comparison of the behavior is formalized by *distinguishers* (often called *environments*), which are systems that connect to all interfaces A, B, and E of either $\pi_1^A\pi_2^B\mathbf{R}$ or $\sigma^E\mathbf{S}$. The distinguisher interacts with the connected system arbitrarily, which means that it inputs messages at the system's interfaces in an arbitrary order and obtains the output of the system as a reply. Similar to [7], each such interaction consists of a single input and a single output message and is called a *query* to the system. To count the queries that have been issued to the A, B, and E interfaces of a system, we use q_A, q_B, and q_E, respectively. For the total number of queries, we write $q := q_A + q_B + q_E$. After the complete interaction, the distinguisher makes a "guess" which system it was connected to. If no distinguisher can differentiate between the two systems, the systems can be used interchangeably in any environment.

The complete interaction of the distinguisher $\mathbf{D}$ and the system $\mathbf{S}$ defines a random experiment $\mathbf{DS}$. The final output of $\mathbf{D}$ is denoted by the random variable W, and the probability that $\mathbf{D}$ outputs 1 is written as $\mathsf{P}^{\mathbf{DS}}(W = 1)$.

DEFINITION 1. *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 final output of $\mathbf{D}$.

The distinguishing advantage for a set $\mathcal{D}$ of distinguishers is defined as $\Delta^{\mathcal{D}}(\mathbf{U}, \mathbf{V}) := \sup_{\mathbf{D}\in\mathcal{D}} \Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V})$. Let $\mathcal{D}_q$ be the set of all distinguishers that issue at most q queries to the connected system. We set $\Delta_q(\mathbf{U}, \mathbf{V}) := \Delta^{\mathcal{D}_q}(\mathbf{U}, \mathbf{V})$. Moreover, by $\mathcal{E}$, we denote the class of all efficient distinguishers.

Using Definition 1, the security of a protocol is defined by comparing the "real" and "ideal" executions. A further requirement for the protocol is availability: If no adversary is present, the protocol must implement the specified functionality. This requirement excludes trivial protocols. For the definition of availability, we use the special converter "$\perp$" that, when attached to the E-interface of a system, blocks the E-interface for the distinguisher.

DEFINITION 2. *The protocol* π constructs $\mathbf{S}$ from the resource $\mathbf{R}$ with error ε and with respect to the distinguisher class $\mathcal{D}$ *if*

$$\exists\,\sigma:\ \Delta^{\mathcal{D}}(\pi_1^A\pi_2^B\mathbf{R}, \sigma^E\mathbf{S}) \le \varepsilon \qquad \text{(security)}$$

where σ is an efficient converter, and

$$\Delta^{\mathcal{D}}(\pi_1^A\pi_2^B\perp^E\mathbf{R}, \perp^E\mathbf{S}) \le \varepsilon. \qquad \text{(availability)}$$

An important property of Definition 2 is its composability (in the asymptotic setting). That is, if a resource $\mathbf{S}$ is used in the construction of a larger system, then the composability enables us to replace the resource $\mathbf{S}$ by a construction $\pi^A\pi^B\mathbf{R}$ without affecting the security of the composed system. Theorem 1 shows that the indistinguishability of $\sigma^E\mathbf{S}$ and $\pi^A\pi^B\mathbf{R}$ is preserved under both the application of a protocol and the parallel composition with further resources.

The sequential composition of converters is denoted by $\psi\circ\pi$, and is defined as $(\psi \circ \pi)^A\mathbf{R} = \psi^A(\pi^A\mathbf{R})$. The parallel composition $\psi\|\pi$ of converters is defined as $(\psi\|\pi)^A(\mathbf{R}\|\mathbf{S}) = (\psi^A\mathbf{R})\|(\pi^A\mathbf{S})$. The term id means that the interfaces of the corresponding subsystem are accessible through the interfaces of the combined system (intuitively, id only relays the interface of the sub-system it is attached to).

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 $\mathbf{R}$ with error ε_π and ψ constructs $\mathbf{T}$ from $\mathbf{S}$ with error ε_ψ.*

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 $\mathbf{R}$ with error $\varepsilon_\pi + \varepsilon_\psi$, $(\pi_1\|\mathsf{id}, \pi_2\|\mathsf{id})$ constructs $\mathbf{S}\|\mathbf{U}$ from $\mathbf{R}\|\mathbf{U}$ with error ε_π and $(\mathsf{id}\|\pi_1, \mathsf{id}\|\pi_2)$ constructs $\mathbf{U}\|\mathbf{S}$ from $\mathbf{U}\|\mathbf{R}$ with error ε_π.

PROOF. By the assumptions on the protocols π and ψ, there exist simulators σ_π and σ_ψ such that $\Delta^{\mathcal{D}}(\pi_1^A\pi_2^B\mathbf{R}, \sigma_\pi^E\mathbf{S}) = \varepsilon_\pi$ and $\Delta^{\mathcal{D}}(\psi_1^A\psi_2^B\mathbf{S}, \sigma_\psi^E\mathbf{T}) = \varepsilon_\psi$. Hence, for all $\mathbf{D} \in \mathcal{D}$,

$$\begin{aligned}
&\Delta^{\mathbf{D}}((\psi_1 \circ \pi_1)^A(\psi_2 \circ \pi_2)^B\mathbf{R}, (\sigma_\pi \circ \sigma_\psi)^E\mathbf{T}) \\
&\quad\le\ \Delta^{\mathbf{D}}((\psi_1 \circ \pi_1)^A(\psi_2 \circ \pi_2)^B\mathbf{R}, \psi_1^A\psi_2^B\sigma_\pi^E\mathbf{S}) \\
&\qquad + \Delta^{\mathbf{D}}(\psi_1^A\psi_2^B\sigma_\pi^E\mathbf{S}, (\sigma_\pi \circ \sigma_\psi)^E\mathbf{T}) \\
&=\ \Delta^{\mathbf{D}\psi_1^A\psi_2^B(\cdot)}(\pi_1^A\pi_2^B\mathbf{R}, \sigma_\pi^E\mathbf{S}) + \Delta^{\mathbf{D}\sigma_\pi^E(\cdot)}(\psi_1^A\psi_2^B\mathbf{S}, \sigma_\psi^E\mathbf{T}) \\
&\qquad\qquad \le \varepsilon_\pi + \varepsilon_\psi,
\end{aligned}$$

which means that $\psi \circ \pi$ constructs $\mathbf{T}$ from $\mathbf{R}$ with error $\varepsilon_\pi + \varepsilon_\psi$. The first step follows from the triangle inequality, and the last step is valid since $\mathcal{D}$ is closed under composition, so $\mathbf{D}\psi_1^A\psi_2^B(\cdot) \in \mathcal{D}$ and $\mathbf{D}\sigma_\pi^E(\cdot) \in \mathcal{D}$.

We also have to show the condition for parallel composition, which amounts to

$$\begin{aligned}
&\Delta^{\mathbf{D}}((\pi_1\|\mathsf{id})^A(\pi_2\|\mathsf{id})^B(\mathbf{R}\|\mathbf{U}), (\sigma_\pi\|\mathsf{id})^E(\mathbf{S}\|\mathbf{U})) \\
&\qquad = \Delta^{\mathbf{D}}((\pi_1^A\pi_2^B\mathbf{R})\|\mathbf{U}, (\sigma_\pi^E\mathbf{S})\|\mathbf{U}) \\
&\qquad\qquad = \Delta^{\mathbf{D}(\cdot\|\mathbf{U})}(\pi_1^A\pi_2^B\mathbf{R}, \sigma_\pi^E\mathbf{S}) \le \varepsilon_\pi,
\end{aligned}$$

where $\mathbf{D}(\cdot\|\mathbf{U})$ denotes the construction that runs $\mathbf{U}$ in parallel to the argument and connects the resulting system to $\mathbf{D}$. Hence, the resulting system is a distinguisher and we have $\mathbf{D}(\cdot\|\mathbf{U}) \in \mathcal{D}$. The condition for the availability is proven analogously. □

The following lemma is used in the proof of the main theorem. Abstractly, the statistical distance of two monotone binary sequences is bounded by the sum of the distances of the individual components.

LEMMA 1. *Let $A_1, \ldots, A_q$ and $B_1, \ldots, B_q$ be two monotone binary sequences of length q. The statistical distance of the sequences is at most*

$$d(A^q, B^q) \leq \sum_{i=1}^{q} |\alpha_i - \beta_i|,$$

where $\alpha_i := \mathsf{P}(A_i = 1 | A_{i-1} = 0)$ and β_i is defined analogously.

PROOF. For brevity, we write $\gamma_i^A := \prod_{j=1}^{i}(1-\alpha_j)$ and $\gamma_i^B := \prod_{j=1}^{i}(1-\beta_j)$. The statistical distance of the sequences is:

$$d(A^q, B^q) = \frac{1}{2}\Big[\Big[\sum_{i=1}^{q} |\gamma_{i-1}^A \alpha_i - \gamma_{i-1}^B \beta_i|\Big] + |\gamma_q^A - \gamma_q^B|\Big].$$

Using the following estimation:

$$\begin{aligned}\big|\gamma_{i-1}^A \alpha_i - \gamma_{i-1}^B \beta_i\big| + \big|\gamma_{i-1}^A (1-\alpha_i) - \gamma_{i-1}^B (1-\beta_i)\big| \\ \leq |\gamma_{i-1}^A - \gamma_{i-1}^B| + 2\gamma_{i-1}^A |\alpha_i - \beta_i|, \quad (1)\end{aligned}$$

we bound the statistical distance by induction on i:

$$\begin{aligned}&\frac{1}{2}\Big[\sum_{j=1}^{i} |\gamma_{j-1}^A \alpha_j - \gamma_{j-1}^B \beta_j| + |\gamma_i^A - \gamma_i^B|\Big] \\ &\leq \frac{1}{2}\Big[\sum_{j=1}^{i-1} |\gamma_{j-1}^A \alpha_j - \gamma_{j-1}^B \beta_j| + |\gamma_{i-1}^A - \gamma_{i-1}^B| + 2\gamma_{i-1}^A |\alpha_i - \beta_i|\Big] \\ &\leq \sum_{j=1}^{i-1} \big|\alpha_j - \beta_j\big| + |\alpha_i - \beta_i|,\end{aligned}$$

where the first estimation is by inequality (1). □

3. CONSTRUCTING SECURE CHANNELS

The goal of a protocol for secure communication is constructing a secure channel ●⟶● from an insecure channel ⟶ and a key ●—●. Given encryption schemes for confidentiality and MACs for authenticity, an interesting question is whether the construction can be performed in a modular way.

In Section 3.1, we describe the different types of channels and the shared secret key as discrete systems. In Section 3.2, we formulate encryption and MAC as transformations of channels, and in Section 3.3, we relate the game-based security notions for MACs to the transformation of channels.

3.1 Types of Channels

The purpose of a *channel* is to transmit messages $m_i \in \mathcal{M}$ from the A-interface to the B-interface, where $\mathcal{M}$ is the channel's *message space*. The channel takes as input messages $m_i \in \mathcal{M}$ at the interface A and potentially leaks information on m_i at the E-interface. Moreover, the E-interface may admit to modify transferred messages before the channel provides the potentially modified message m_i' as output at interface B. The channels introduced in Section 1.1 can be described as discrete systems that obtain as input messages $m_i \in \mathcal{M}$ at interface A:

- ⟶ An *insecure channel* leaks the complete messages $m_i \in \mathcal{M}$ at E. At interface E, the channel expects $m_j' \in \mathcal{M}$ and outputs m_j' at B.
- ●⟶ An *authenticated channel* also leaks the complete message $m_i \in \mathcal{M}$, but only allows E to forward m_i or to completely abort the channel.
- ●—● A *secure channel* leaks only the message length $|m_i|$, and only allows E to forward m_i or to abort the channel.
- —● *Confidential channels* are discussed in Section 4.

We assume that the channels output the special symbol $\perp \notin \mathcal{M}$ to the receiver on abort. One can consider further variations of the above channels, such as channels that allow the adversary to reorder messages or to delete single messages without completely aborting the channel.

The *shared secret key* ●—● with key space $\mathcal{K}$ is a resource with interfaces A and B, as well as an inactive interface E for the adversary. After obtaining an initialization message at both the A and the B-interface, ●—● draws a key $\kappa \in \mathcal{K}$ uniformly at random and provides κ to both A and B.

3.2 Protocols as Channel Transformations

Following the paradigm of constructive cryptography, cryptographic protocols are interpreted as transformations of channels.

Encryption.

An *encryption protocol* is a pair $\mathsf{SC} = (\mathsf{enc}, \mathsf{dec})$ with key space $\mathcal{K}$, message space $\mathcal{M}$, and ciphertext space $\mathcal{C}$. The converters enc and dec connect with their inner interfaces to a shared secret key ●—● with key space $\mathcal{K}$ and to some channel with a message space $\mathcal{M}' \supseteq \mathcal{C}$. The resulting resource is a channel with message space $\mathcal{M}$.

The security goal of encryption can be interpreted as transforming an authenticated channel ●⟶● into a secure channel ●—● by use of a secret key ●—●, which corresponds to the standard notion of chosen-plaintext security.[3]

Authentication.

An *authentication protocol* is a pair $\mathsf{AUT} = (\mathsf{aut}, \mathsf{chk})$ of converters with key space $\mathcal{K}$, input message space $\mathcal{M}$, and output message space $\mathcal{M}'$. aut and chk connect with their inner interfaces to a shared secret key ●—● with key space $\mathcal{K}$ and to a channel with message space $\mathcal{M}'' \supseteq \mathcal{M}'$. The resulting resource is a channel with message space $\mathcal{M}$.

The transformation of an insecure channel ⟶ into an authenticated channel ●⟶● is closely related to the standard game-based security notions for MACs, as discussed in Section 3.3.

Composition.

Using Theorem 1, one sees that the EtA-composition of an encryption and an authentication protocol that are sufficient for the above transformations constructs a secure channel ●—● from an insecure channel ⟶ and two independent shared secret keys ●—●.

3.3 Example: Authentication with MACs

MACs can be used as building blocks for authentication protocols. In this section, we describe the application of the MAC scheme in TLS as an authentication protocol and show that, if the

[3]The condition is equivalent to the standard game-based notion of IND-CPA security [2].

MAC is secure according to the game-based standard notions, the resulting protocol transforms an insecure channel $\longrightarrow$ into an authenticated channel $\bullet\!\!\longrightarrow$.

A *MAC scheme* is a pair of stateless converters $\mathsf{MAC} = (\mathsf{tag}, \mathsf{vrf})$ with key space $\mathcal{K}$, message space $\mathcal{M}$, and tag space $\mathcal{T}$. tag and vrf connect to a shared secret key with key space $\mathcal{K}$, as depicted in Figure 3. At the outside interface, tag obtains messages M_i from the set $\mathcal{M}$ and answers with tags T_i from $\mathcal{T}$. vrf expects as inputs pairs V_j from $\mathcal{M} \times \mathcal{T}$ and answers with a single bit F_j.

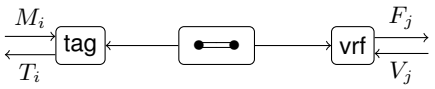


Figure 3: Tag generation and verification in a MAC scheme.

The security of a MAC scheme is usually defined by a game. In this paper, we concentrate on MACs that achieve *strong unforgeability under chosen-message attack* (SUF-CMA). Intuitively, SUF-CMA security means that no (efficient) attacker, even with access to an oracle that produces valid tags for chosen messages, will be able to generate a valid message-tag pair other than the pairs obtained from the oracle. The SUF-CMA condition after q queries with $q = q_A + q_B$ is

$$S_q = S_{q-1} \vee \big(F_{q_B} = 1 \wedge \forall i \leq q_A : (M_i, T_i) \neq V_{q_B}\big). \quad (2)$$

Consider a system $\mathbf{W}$ that connects to the outside interfaces of tag and vrf. The MAC scheme is $\varepsilon(q)$-SUF-CMA secure if for all efficient $\mathbf{W}$,

$$\mathsf{P}^{\mathbf{W}(\mathsf{tag}^A \mathsf{vrf}^B(\bullet\!\!\longrightarrow\!\!\bullet))}\big(S_q = 1\big) \leq \varepsilon(q).$$

Weak unforgeability (WUF-CMA) is less restrictive than strong unforgeability in the sense that an adversary may be able to produce further tags $T_i' \neq T_i$ such that (M_i, T_i') is a valid pair and T_i was returned by tag on input M_i.

The Authentication Protocol.

Based on a MAC scheme, we specify an authentication protocol $\mathsf{AUT} = (\mathsf{aut}, \mathsf{chk})$ as depicted in Figure 4. The structure of this protocol resembles the scheme implemented in TLS [12], and is based on the following components.

- **tag/vrf:** A (SUF-CMA secure) MAC scheme with message space $\mathcal{M} \times \mathbb{N}$, key space $\mathcal{K}$, and tag space $\mathcal{T}$.
- **η/η^{-1}:** An efficiently computable and efficiently invertible (injective) encoding $\eta : \mathcal{M} \times \mathcal{T} \to \mathcal{M}'$. We assume that η^{-1} returns $\perp$ on values not in the range of η. The function η also subsumes the padding scheme (if present).
- **seq:** A system maintaining a counter $i \in \mathbb{N}$ (initially $i = 0$): For each input m_i, output (m_i, i) and set $i \leftarrow i + 1$.
- **seq′:** A system maintaining a counter $j \in \mathbb{N}$ (initially $j = 0$): For each input (m_j', t_j') from η^{-1}, query $((m_j', j), t_j')$ at vrf. If η^{-1} input $\perp$ or if vrf returned 0, output $\perp$ and abort. Otherwise, output m_j' and set $j \leftarrow j + 1$.

The messages input at the converter aut are denoted by $\tilde{A}_i$, and the authenticated messages produced by aut are denoted by A_i. The messages output at the receiver's interface of the (insecure) channel are referred to as B_j, and the messages output at the B-interface of the complete system are denoted as $\tilde{B}_j$.

The protocol AUT guarantees that fulfilling the condition

$$C_q = C_{q-1} \vee ((q_A < q_E \vee A_{q_E} \neq B_{q_E}) \wedge \tilde{B}_{q_E} \neq \perp) \quad (3)$$

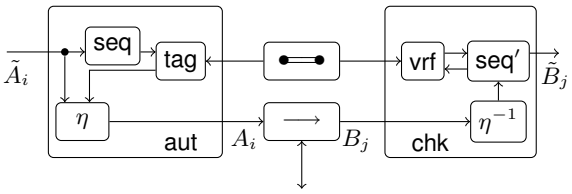


Figure 4: The authentication protocol $(\mathsf{aut}, \mathsf{chk})$ is based on the MAC scheme $(\mathsf{tag}, \mathsf{vrf})$. The systems seq and seq′ handle sequence numbers; η and η^{-1} convert the message-tag pairs for the message space of $\longrightarrow$.

implies forging the MAC. This holds since $C_q = 1$ means that there exists an $i \leq q$ such that $\tilde{B}_i \neq \perp$ (so $F_i = 1$) and for the corresponding i_A, i_E: $i_A < i_E \vee A_{i_E} \neq B_{i_E}$, which satisfies condition (2) since B_{i_E} does not originate from a previous query to tag (the sequence number is unique).[4]

A channel that guarantees the validity of condition (3) is authenticated: Either the messages delivered at the B-interface are unchanged, or the channel aborts. The simulator σ needed for the proof can be constructed easily: σ draws a random key $\kappa \in \mathcal{K}$ and uses the MAC scheme to simulate honestly computed tags.

4. FORMALIZING MALLEABILITY

4.1 Confidential Channels

The characterizing feature of a confidential channel $\longrightarrow\!\!\bullet$ is that transmitted messages (except for possibly their length) are hidden from the adversary. Yet, there is no authenticity or integrity guarantee: the adversary can, in principle, modify the transmitted message arbitrarily. In contrast to the insecure channel, this cannot be formalized by simply allowing the adversary to specify the new message, since the distribution of the output may depend on the transmitted plaintexts that are not known to the adversary.

EXAMPLE 1 (ONE-TIME PAD AND XOR-MALLEABILITY). *For a single-message instance of the one-time pad encryption, the message $m \in \{0,1\}^n$ is encrypted by computing bit-wise XOR with the key $\kappa \in \{0,1\}^n$, resulting in the ciphertext $c = m \oplus \kappa \in \{0,1\}^n$. If the adversary replaces the ciphertext c by $c' \in \{0,1\}^n$, then the receiver will compute $m' = c' \oplus \kappa = c' \oplus (c \oplus m) = (c \oplus c') \oplus m$. Hence, the channel obtained by applying the one-time pad to the insecure channel can be seen as the confidential channel that allows the adversary to specify a function $f_\delta : \{0,1\}^n \to \{0,1\}^n, m \mapsto m \oplus \delta$ by replacing c with $c \oplus \delta$.*

Using Definition 2, one can show that (for channels with a fixed message length) the malleability of the one-time pad encryption is completely described by the channel described in Example 1. Formally, the absence of authenticity or integrity guarantees for the B-interface is described by preventing the distinguisher from accessing this interface. This is formalized by a converter $\perp$ that does not provide any output at its outer interface.

DEFINITION 3. *A channel $\mathbf{C}$ is* confidential with error ε *if*

$$\exists\, \sigma :\ \Delta^{\mathcal{D}}(\perp^B(\mathbf{C}), \perp^B \sigma^E(\bullet\!\!\longrightarrow\!\!\bullet)) \leq \varepsilon.$$

The channel is called perfectly confidential *if $\varepsilon = 0$, and* confidential *if ε is negligible.*

[4]Even the weaker WUF-CMA [1] notion is sufficient, but the requirements turn out to be equivalent for the schemes used in TLS.

4.2 A Model for Malleability

The concept of *malleability* of a confidential channel $\longrightarrow\!\!\bullet$ captures the adversarial influence on the transferred messages. For each message, this influence can be described as a transformation that is applied to the message before it is delivered at the B-interface. Hence, the malleability of a channel is specified by the set of all available such transformations, and, intuitively, the smaller the set, the more secure the channel.

In general, the distribution of the messages output at the B-interface depends on all previous messages at the A, B, and E-interfaces of the channel. From the adversary's perspective, the messages at the E-interface determine a (probabilistic) transformation $F : \mathcal{M}^* \times \mathcal{M}^* \to \mathcal{M}$, where the first parameter corresponds to the inputs at the A-interface and the second parameter corresponds to the outputs at the B-interface. From an operational point of view, the adversary's input at the E-interface corresponds to a choice of a specific transformation F from the set of all available transformations. The outcome of the transformation F is the output at the B-interface.

DEFINITION 4. *An* $\mathcal{F}$-malleable confidential channel $\longrightarrow\!\!\bullet$ *is a confidential channel such that the malleability is described by 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* $F_\alpha : \mathcal{M}^* \times \mathcal{M}^* \to \mathcal{M}$ *and, after* q *queries, the random variable* $\mathcal{A}_q \subseteq \mathcal{A}$ *describes the eligible transformations.*

On receiving input m_{q_A} *at the* A*-interface,* $\longrightarrow\!\!\bullet$ *outputs* $|m_{q_A}|$ *and a description of* $\mathcal{A}_q$ *at the* E*-interface. Upon receiving input* $\alpha \in \mathcal{A}_q$ *at the* E*-interface,* $\longrightarrow\!\!\bullet$ *evaluates the transformation* F_α *on the plaintexts and outputs the result at the* B*-interface. If the* $\perp$*-converter is attached to the* E*-interface,* $\longrightarrow\!\!\bullet$ *immediately delivers* m_{q_A} *at the* B*-interface.*

The distribution of each $\mathcal{A}_q$ *depends on the lengths* $|m_\ell|$ *of the messages input at the* A*-interface, and the previous* $\mathcal{A}_1, \ldots, \mathcal{A}_{q-1}$ *and* $\alpha_1, \ldots, \alpha_{q_E}$*.*[5]

An encryption protocol $\mathsf{SC} = (\mathsf{enc}, \mathsf{dec})$ *is called* $\mathcal{F}$-malleable *if it transforms an insecure channel* $\longrightarrow$ *into an* $\mathcal{F}$*-malleable confidential channel* $\longrightarrow\!\!\bullet$ *by use of a shared secret key* $\bullet\!\!=\!\!\bullet$*.*

4.3 Capturing Existing Notions

While the concept of malleability appears in various parts of the literature, only specific notions of non-malleability have been formalized previously. Non-malleability for encryption schemes was first considered for the public-key case [3, 13] and was later transferred to symmetric encryption [5]. These notions are defined as games: The adversary is given access to oracles that describe the attack, such as encryption or decryption oracles. The adversary's goal in these games is to specify a ciphertext such that the decrypted plaintext is "meaningfully related"—in a well-defined sense—to the honestly generated plaintexts.

Our model captures notions of non-malleability by defining suitably restricted classes of malleability functions (for the standard notion [5], only forwarding and deleting messages as well as injecting fresh plaintexts are allowed).[6] Further notions such as plaintext integrity [15] and notions such as existential unforgeability [15, 16] can be captured similarly. An exhaustive comparison is out of the scope of this work.

[5]In particular, there may be further communication at the E interface to communicate the set $\mathcal{A}_q$ to the adversary. The conditions make sure that this communication can be simulated efficiently.

[6]Technically, the game-based "attack model" corresponding to our definition is slightly weaker than chosen-ciphertext security. See [11] for a related discussion.

5. AUTHENTICATE-THEN-ENCRYPT

As a general confidential channel cannot be transformed into a secure channel by a MAC, a natural question is in which way the requirements on the confidential channel must be strengthened to allow for this transformation. In Section 5.1, we formulate a suitable restriction on the malleability of confidential channels, and in Section 5.2, we prove that such channels are converted into secure channels by MACs.

5.1 Restricted Malleability

To provide intuition for the types of malleability that are insufficient, we modify the example given by Krawczyk [17] to highlight the impact of the MAC: A message $m \in \{0,1\}^n$ is encoded bitwisely to a message $m' \in \{0,1\}^{2n}$, before m' is encrypted with a one-time pad to guarantee confidentiality. The encoding of bits, however, is asymmetric: A 0-bit is encoded to 00, while a 1-bit is encoded to either $10, 01$, or 11. Hence, if the adversary flips two subsequent bits c_{2i}, c_{2i+1} of the ciphertext, the corresponding plaintext bit m_i will always flip if $m_i = 0$, but flips with probability only $\frac{1}{3}$ if $m_i = 1$. As the verification of a strongly unforgeable MAC will succeed if and only if the authenticated plaintext is unchanged, an adversary can guess the bit m_i with substantial probability by flipping c_{2i}, c_{2i+1} and checking whether the verification succeeds. Hence, the verification of the MAC leaks the bit at the corresponding position, which breaks confidentiality.

More abstractly, if the malleability allows the adversary to transform the plaintext such that the probability of remaining constant differs substantially for two different values of the plaintext, then the adversary can use the result of the MAC verification (valid or invalid) to detect which one of the two plaintexts was sent.

Definition 5 describes classes of transformations that exclude this behavior. For *forwarding* transformations, the probability for the plaintexts to remain constant is independent of the exact plaintext, so the (successful) verification of the MAC does not leak information on the plaintext. *Deleting* transformations change any plaintext with overwhelming probability, so the MAC verification will always fail.[7] For *reconstructible* transformations, the result of the transformation does not depend on the "target" message. This means that, if the correct message occurs with some good probability, one can compute the output using only the other plaintext messages, which contradicts the unforgeability of the MAC.

DEFINITION 5. *Let* $F : \mathcal{M}^* \times \mathcal{M}^* \to \mathcal{M}$ *be a transformation on the plaintext space. After* $q = q_A + q_E$ *queries,* F *is*

- forwarding with error $\delta(q)$ *if, for all* $m^{q_A}, \tilde{m}^{q_A} \in \mathcal{M}^{q_A}$ *and all* $m'^{q_E}, \tilde{m}'^{q_E} \in \mathcal{M}^{q_E}$ *with* $|m_i| = |\tilde{m}_i|$ *and* $|m'_j| = |\tilde{m}'_j|$*,*

$$\big|\mathsf{P}(F(m^{q_A}, m'^{q_E}) = m_{q_E+1}) - \mathsf{P}(F(\tilde{m}^{q_A}, \tilde{m}'^{q_E}) = \tilde{m}_{q_E+1})\big| \leq \delta(q),$$

- deleting with error $\delta(q)$ *if, for all* $m^{q_A} \in \mathcal{M}^{q_A}$ *and all* $m'^{q_E} \in \mathcal{M}^{q_E}$*,*

$$\mathsf{P}(F(m^{q_A}, m'^{q_E}) = m_{q_E+1}) \leq \delta(q),$$

- reconstructible with error $\delta(q)$ *if there is an efficient algorithm* R *such that for all* $m^{q_A} \in \mathcal{M}^{q_A}$ *and all* $m'^{q_E} \in \mathcal{M}^{q_E}$*,*

$$\big|\mathsf{P}(F(m^{q_A}, m'^{q_E}) = m_{q_E+1}) - \mathsf{P}(\mathsf{R}(\mathsf{M}) = m_{q_E+1})\big| \leq \delta(q),$$

[7]This is actually a subclass of the forwarding transformations, the separation will become clear in Definition 6.

where $\mathsf{M} \subseteq \{m_1, \ldots, m_{q_A}, m'_1, \ldots, m'_{q_E}\} \setminus \{m_{q_E+1}\}$.

DEFINITION 6. *Let* $\longrightarrow\!\!\bullet$ *be an* $\mathcal{F}$*-malleable perfectly confidential channel with* $\mathcal{F} = (\{F_\alpha\}_{\alpha\in\mathcal{A}}, \{\mathcal{A}_i\}_{i\in\mathbb{N}})$. *Then,* $\longrightarrow\!\!\bullet$ *is* AtE-compatible with error $\delta(q)$ *if, given that all* $\alpha_1, \ldots, \alpha_{q_E}$ *referred to forwarding transformations (that are not also deleting transformations) and* $q_A \geq q_E$*, each* F_α *for* $\alpha \in \mathcal{A}_{q+1}$ *is either forwarding, deleting, or reconstructible each with error* $\delta(q+1)$.

An encryption scheme is AtE-compatible with error $\delta(q)$ *if it transforms an insecure channel* $\longrightarrow$ *into a confidential channel* $\longrightarrow\!\!\bullet$ *that is also AtE-compatible with error* $\delta(q)$.

5.2 Soundness of Authenticate-then-Encrypt

The AtE-transformation is sound for confidential channels with malleability that can be described by the classes introduced in Definition 5. If each transformation chosen by the adversary is of one of the given types, then each single step can be simulated. Using Lemma 1, we conclude that the authentication protocol transforms the confidential into a secure channel.

THEOREM 2. *Let* $\mathsf{AUT} = (\mathsf{aut}, \mathsf{chk})$ *be the authentication protocol based on a* $\varepsilon(q)$*-SUF-CMA MAC* MAC *and an encoding* η *as described in Section 3.3, such that there is a publicly computable mapping* $\ell : \mathbb{N} \to \mathbb{N}$ *with* $\ell(|m|) = |\eta(m,t)|$. *Let* $\longrightarrow\!\!\bullet$ *be an AtE-compatible channel with error* $\delta(q)$ *according to Definition 5. Then,* AUT *transforms* $\longrightarrow\!\!\bullet$ *into the secure channel* $\bullet\!\!\longrightarrow\!\!\bullet$. *Formally, there is a simulator* σ *such that*

$$\Delta_q(\mathsf{aut}^A\mathsf{chk}^B(\longrightarrow\!\!\bullet \parallel \bullet\!\!\longrightarrow\!\!\bullet), \sigma^E(\bullet\!\!\longrightarrow\!\!\bullet)) \leq q(\delta(q) + \varepsilon(|\mathsf{M}_q|)) + \varepsilon(q),$$

where $|\mathsf{M}_q|$ *is an upper bound for the number of messages needed by the reconstruction algorithms.*

PROOF. We use the notation $\mathbf{R} := \mathsf{aut}^A\mathsf{chk}^B(\longrightarrow\!\!\bullet \parallel \bullet\!\!\longrightarrow\!\!\bullet)$ and $\mathbf{S} := \sigma^E(\bullet\!\!\longrightarrow\!\!\bullet)$. The setting considered in this proof is similar to the setting in Figure 4, but the authentication protocol is applied to the confidential channel $\longrightarrow\!\!\bullet$. Throughout the proof, the inputs at the A-interface of either $\mathbf{R}$ or $\mathbf{S}$ are denoted by $\tilde{A}_i$, and the outputs at the B-interface are denoted by $\tilde{B}_j$. In the system $\mathbf{R}$, for the messages from aut to $\longrightarrow\!\!\bullet$ we use A_i, and for those from $\longrightarrow\!\!\bullet$ to chk we use B_j.

We use the following simulator σ: Obtaining the message length l_i from $\bullet\!\!\longrightarrow\!\!\bullet$, simulate the E-interface of $\longrightarrow\!\!\bullet$ for the message length $\ell(l_i)$. (By Definition 4, it is guaranteed that the distribution of $\mathcal{A}_i$ only depends on information known to σ.) Obtaining a message α_i at the outside interface, abort the channel $\bullet\!\!\longrightarrow\!\!\bullet$ if $\alpha_i \notin \mathcal{A}_i$. If $\alpha_i \in \mathcal{A}_i$, proceed as follows.

- If F_{α_i} is forwarding, the message remains constant with a certain probability γ. Hence, forward the message with probability γ and abort the channel with probability $1-\gamma$.
- Otherwise, abort the channel.

Let $C_i = C_{i-1} \vee ((i_A < i_E \vee A_{i_A} \neq B_{i_E}) \wedge \tilde{B}_{i_E} \neq \perp)$ be the monotone binary sequence that represents the security of the authentication protocol, as in condition (3). In the following analysis, we compare the system $\mathbf{R}$ conditioned on the monotone binary sequence $C_1, C_2, \ldots$ with the system $\mathbf{S}$. Then, we use [20, Lemma 5] to bound the distinguishing advantage.

For each query at the A-interface, the output at the E-interfaces of $\mathbf{R}$ and $\mathbf{S}$ is distributed identically. Conceptually, we can absorb the generation of this output into the distinguisher, which then only distinguishes $\mathbf{R}$ and $\mathbf{S}$ based on the monotone binary sequence defined by "$B_i = \perp$". For each fixed such distinguisher $\mathbf{D}'$, the advantage is bounded by the statistical distance of the corresponding sequences for $\mathbf{R}$ and $\mathbf{S}$, so we can apply Lemma 1 and bound the advantage by the individual summands for each invocation of the malleability. By the assumption that $\longrightarrow\!\!\bullet$ is AtE-compatible, we can distinguish the three cases from Definition 5.

Forwarding transformations.

The simulator σ forwards or deletes the messages with some appropriate probability γ. By Definition 5, the distance is at most

$$\max_{m^{i_A}, m'^{i_E-1}} \left|\mathsf{P}(F_\alpha(m^{i_A}, m'^{i_E-1}) = m_{i_E}) - \gamma\right| \leq \delta(i).$$

Deleting transformations.

The simulator σ aborts the channel. By Definition 5, the distance is at most

$$\max_{m^{i_A}, m'^{i_E-1}} \mathsf{P}(F_\alpha(m^{i_A}, m'^{i_E-1}) = m_{i_E}) \leq \delta(i).$$

Reconstructible transformations.

The simulator simply aborts the channel. We bound the distinguishing advantage using a reduction: Using the algorithm R_α guaranteed by Definition 5, we construct a system that breaks the SUF-CMA property of the MAC scheme.

Since the channel is aborted in the ideal case, we have to bound the probability for $\tilde{B}_{i_E} \neq \perp$ in $\mathbf{R}$. For the reduction, we construct a system $\mathbf{H}$ that simulates $\mathbf{S}$ and relays the communication between $\mathbf{S}$ and the distinguisher $\mathbf{D}$. Once $\mathbf{D}$ chooses a reconstructible transformation F_α, $\mathbf{H}$ invokes tag on the pairs (m_i, i) required by R_α and provides M_α to R_α. By Definition 5, the algorithm R_α produces a wrong output with probability at most $\delta(i)$, which implies, by the triangular inequality, that the distance is at most

$$\delta(i) + \mathsf{P}^{\mathbf{H}(\mathbf{D}\parallel\mathsf{tag}^A\mathsf{vrf}^B(\bullet\!\!\longrightarrow\!\!\bullet))}\big(S_i = 1\big) \leq \delta(i) + \varepsilon(|\mathsf{M}_\alpha|).$$

With Lemma 1, the triangle inequality, and [20, Lemma 5], we bound the distinguishing advantage by $q(\delta(q) + \varepsilon(|\mathsf{M}_\alpha|)) + \varepsilon(q)$.

The availability follows from the correctness of the MAC scheme and the availability of $\longrightarrow\!\!\bullet$. □

6. ANALYZING TLS

The TLS protocol [12] implements two different types of encryption: The stream cipher RC4, and block ciphers (3DES and AES) in CBC-mode. We analyze the malleability of these schemes and show their sufficiency for AtE in Sections 6.1 and 6.2, respectively. In Section 6.3, we examine the padding scheme used for CBC encryption in TLS.

6.1 Stream Ciphers

A stream cipher can be seen as a one-time pad encryption with a pseudo-random key stream. We use the one-time pad as an abstraction for the stream ciphers, and note that one can replace the perfectly random key stream with a pseudo-random one using the composition theorem. The *XOR-malleability* exhibited by the one-time pad is sketched in Example 1, but further effects originating from variable message lengths must be considered.

To describe the full XOR-malleability $\mathcal{F}^{\mathsf{xor}}$, we use the following notation: For a fixed state of the channel, the number of messages m_j input at the A-interface is denoted by i_A, and the number of messages m'_j output at the B-interface by i_E. We define $L_j^A := \sum_{\ell=1}^{j} |m_j|$ (and the corresponding term for the B-interface). The concatenation of the messages at the A-interface is a bit-stream,

and the ℓ^{th} bit is denoted by $m[\ell]$ (analogously, $m'[\ell]$ for the B-interface). The ciphertext bits output and input at the E-interface are denoted by $c[\ell]$ and $c'[\ell]$, respectively, and $x[\ell]$ refers to the mask bits, that is, $x[\ell] = c[\ell] \oplus c'[\ell]$.

The transformations are of the following type. For each i_E, the transformation is described by the length l_{i_E} of the output message and a mask of bits $x[\ell]$ with ℓ starting at position $\left(\sum_{\ell=1}^{i_E-1} l_\ell\right) + 1$. For those ℓ where $m[\ell]$ is defined, the output message is $m'[\ell] := m[\ell] \oplus x[\ell]$, and the exceeding part is chosen uniformly at random. When the bits $m[\ell]$ corresponding to such an exceeding part are later specified at the A-interface, the bits $m[\ell] \oplus m'[\ell]$ are provided at the E-interface. Since $m'[\ell]$ was chosen uniformly at random, this does not leak any information on $m[\ell]$ (in particular, the described channel is perfectly confidential according to Definition 3).

LEMMA 2. *The one-time pad protocol* $(\mathsf{otp\text{-}enc}, \mathsf{otp\text{-}dec})$ *constructs a perfectly confidential* $\mathcal{F}^{\mathsf{xor}}$*-malleable channel from an insecure channel and a random key, without error.*

PROOF. For the (generic) state of the channel after i_A messages at the A-interface and i_E messages at the B-interface, the simulator σ proceeds as follows:

1. Receiving the $i_A + 1^{\text{st}}$ message at the inner interface:
   - If $L^A_{i_A} < L^B_{i_E}$, then the bits $x[\ell] = m[\ell] \oplus m'[\ell]$ for $L^B_{i_E} < \ell \leq \min(L^A_{i_A+1}, L^B_{i_E})$ are provided to σ. Simulate the ciphertext bits $c[\ell] = x[\ell] \oplus c'[\ell]$.
   - The remaining bits are chosen uniformly at random.

   The resulting strings are concatenated and output at the outer interface.
2. Receiving the $i_E + 1^{\text{st}}$ message at the outer interface:
   - If $L^A_{i_A} < L^B_{i_E}$, then provide the mask bits $x[\ell] = c[\ell] \oplus c'[\ell]$ for $L^B_{i_E} < \ell \leq \min(L^A_{i_A}, L^B_{i_E+1})$ to the channel.
   - For the remaining $L^B_{i_E+1} - \min(L^A_{i_A+1}, L^B_{i_E})$ bits, tell the channel to choose random bits.

The lemma follows by checking that the simulation is perfect, and by concluding availability from the correctness of the protocol. □

LEMMA 3. *The confidential channel with* $\mathcal{F}^{\mathsf{xor}}$*-malleability is AtE-compatible with error* $\delta(q) = 0$.

PROOF. The lemma is shown by an inductive argument: At each query, the only eligible transformation that is not deleting is specified by the 0-mask with the appropriate length. First, if all previous transformations have been of this type, this transformation is forwarding with error 0. Second,

- if the transformation results in a message of different length, it is deleting without error,
- if the transformation specifies a different mask, then it is deleting without error.

Induction completes the proof. □

We apply Theorem 2 to compute the bound for the AtE composition of the one-time pad with an $\varepsilon(q)$-SUF-CMA MAC.

COROLLARY 1. *Let* $\mathsf{OTP} = (\mathsf{otp\text{-}enc}, \mathsf{otp\text{-}dec})$ *be the one-time pad protocol and* AUT *be the authentication protocol based on an* $\varepsilon(q)$*-SUF-CMA MAC as in Section 3.3. Then the AtE-protocol based on* OTP *and* AUT *transforms an insecure channel* $\longrightarrow$ *into a secure channel* $\bullet\!\!\longrightarrow\!\!\bullet$ *with error* $\varepsilon(q)$.

Our bound is different from that in [17] in two aspects: First, since we consider the one-time pad with a perfectly random key stream, we do not exhibit the quadratic "collision" term in the bound. Second, [17] requires a MAC that is SUF-CMA for one plaintext query, whereas in our case, the MAC has to be secure for q queries. The reason for this is that our analysis of the composition is valid for more general encryption schemes.

6.2 CBC Mode Encryption

The CBC encryption protocol based on a shared uniform random permutation[8] $\mathbf{P} : \{0,1\}^n \to \{0,1\}^n$ is a protocol $\mathsf{CBC} = (\mathsf{cbc\text{-}enc}, \mathsf{cbc\text{-}dec})$ that uses as resources a channel with message space $\{0,1\}^{n\ell}$ for $\ell \in \mathbb{N}$ and the shared URP $\mathbf{P}$. The channel implemented by the CBC encryption also transmits messages in $\{0,1\}^{n\ell}$ for $\ell \in \mathbb{N}$.

In the following, we use $m_i = m_{i,1}m_{i,2}\ldots$ to denote the plaintext messages at interface A consisting of n-bit blocks $m_{i,j} \in \{0,1\}^n$, and $c_i = c_{i,0}c_{i,1}\ldots$ for the corresponding ciphertexts output at interface E. Ciphertexts input at E and plaintexts output at B are denoted by $c'_i = c'_{i,0}c'_{i,1}\ldots$ and $m'_i = m'_{i,0}m'_{i,1}\ldots$, respectively. The encryption proceeds as follows: Messages m_i are encrypted by computing $c_{i,j} \leftarrow \mathbf{P}(c_{i,j-1} \oplus m_{i,j})$, where $c_{i,0}$ is an IV chosen uniformly at random. The decryption computes the plaintext m'_i via $m'_{i,j} \leftarrow \mathbf{P}^{-1}(c'_{i,j}) \oplus c'_{i,j-1}$.

The malleability $\mathcal{F}^{\mathsf{blk}}$ of the channel implemented by CBC encryption resembles the block-oriented structure of the encryption scheme. During the decryption, the ciphertext is split into blocks, and each of the blocks (except for the first one) corresponds to one block of plaintext. In the transformations of $\mathcal{F}^{\mathsf{blk}}$, each such block $m'_{i,j} \in \{0,1\}^n$ is either (nearly) uniformly random or specified by a block $m_{r,s}$ or $m'_{r,s}$ from a previous message and a mask $x \in \{0,1\}^n$. Yet, for each i_E, the eligible transformations only include masks such that $x = c_{r,s-1} \oplus c'_{i_E,j-1}$, where $c'_{i_E,j-1}$ is the ciphertext corresponding to the preceding block (and may be chosen arbitrarily if $j = 1$ or the preceding block is chosen as uniformly random). Upon receiving a nl-bit message from the sender, the next set $\mathcal{A}_i$ is described by issuing an $n(l+1)$-bit string with the corresponding masks at the E-interface. This string is chosen uniformly at random from the set of all strings such that no blockwise collision occurs.

LEMMA 4. *The protocol* $\mathsf{CBC} = (\mathsf{cbc\text{-}enc}, \mathsf{cbc\text{-}dec})$ *implementing CBC encryption based on a URP with block length* n *transforms an insecure channel into a perfectly confidential* $\mathcal{F}^{\mathsf{blk}}$*-malleable channel, with error* $(q\ell)^2/2^n$ *and* $|\mathsf{M}_q| = \min(q_A, l)$.

PROOF. First we see that the channel is indeed perfectly confidential: The only information about $A_\ell = m_\ell$ that is provided at the E-interface is a string of length $|m_\ell| + n$ whose distribution is independent from the plaintext value.

The simulator σ, upon obtaining an $|m_\ell| + n$-bit string from the channel, uses this string as the simulated ciphertext c_i. Upon receiving a ciphertext c'_ℓ at the E-interface, σ splits the ciphertext c'_ℓ into blocks and chooses the malleability at the channel as follows:

1. For blocks that have not occurred previously, σ randomizes the corresponding plaintext block.

[8] A uniform random permutation (URP) is a system that behaves as a bijective function $\pi : \mathcal{M} \to \mathcal{M}$ chosen uniformly at random from the set of all such functions. URPs are often used as an abstraction for block-ciphers with uniformly chosen keys. A shared URP is an ABE-system that allows to evaluate and invert a URP at both interfaces A and B and has a trivial E-interface. We prove the security of CBC based on a shared URP and apply the composition theorem to instantiate the URP by a block cipher and a shared secret key.

2. Otherwise, σ chooses the corresponding plaintext block with the appropriate masks.

It is easy to see that σ chooses the masks in such a way that the transformation is eligible.

Let $C_1, C_2, \ldots$ be the monotone binary sequence that describes that, for ciphertexts generated at the E-interfaces, no blocks collide with previous ciphertext blocks (either generated or input at the E-interface). Moreover, let $B_1, B_2, \ldots$ be the monotone binary sequence describing that no collision of the following form occurs: For two pairs of indices $(i,j) \neq (i',j')$, the values $c_{i,j-1} \oplus m_{i,j}$ (or $c'_{i,j-1} \oplus m'_{i,j}$) and $c_{i',j'-1} \oplus m_{i',j'}$ (or $c'_{i',j'-1} \oplus m'_{i',j'}$) collide.

Given that the "real" system conditioned on $C_1, C_2, \ldots$ and the "ideal" system conditioned on $B_1, B_2, \ldots$ behave identically, [20, Lemma 5] yields that the distinguishing probability is at most the probability to provoke that either $C_q = 1$ in the "real" system or $B_q = 1$ in the "ideal" system.

First, we see that the two systems behave identically unless $B_q = 1$ or $C_q = 1$. On input m_ℓ at the A-interface, both systems output an $|m_\ell| + n$-bit string with the same distribution at the B-interface. For a ciphertext c'_i provided at the E-interface, the distribution is as follows:

- If all blocks $c'_{i,j}$ (except for the IV) are taken from the previously used ciphertexts, the resulting message is determined by the same deterministic computation in both cases.
- If there is a new block $c'_{i,j}$, the URP generates a (plaintext) block that is different from all previous inputs, and the preceding block is used as a mask. The ideal system conditioned on $\mathcal{B}$ generates an output with the same distribution.

To complete the proof, it is sufficient to analyze the probabilities of provoking either of the conditions $C_q = 1$ or $B_q = 1$.

Provoking $C_q = 1$ in the real system: This is the probability that, during the encryption, two inputs to $\mathbf{P}$ are equal. If the output of $\mathbf{P}$ were uniformly distributed, the probability would be $(ql)^2/2^{n+1}$. But $\mathbf{P}$ is a URP, which adds another $(ql)^2/2^{n+1}$ by the switching lemma. Hence, this is bounded by $(ql)^2/2^n$.

Provoking $B_q = 1$ in the ideal system: This is the probability that either during the (simulated) decryption, the randomized block is chosen such that it is equal to any previous input to $\mathbf{P}$, or a generated mask corresponds to a collision of inputs to $\mathbf{P}$. By the same argument as for $C_q = 1$, this is bounded by $(ql)^2/2^n$.

The reconstruction algorithm R_α for $\alpha \in \mathcal{A}$ for messages with at most l blocks uses at most $\min(q_A, l)$ previous messages. The availability of the implemented channel follows from the correctness of the protocol. □

LEMMA 5. *The $\mathcal{F}^{\mathsf{blk}}$-malleable channel is AtE-compatible with error $ql^2/2^n$.*

PROOF SKETCH. Several classes of transformations can be distinguished:

1. Transformations to messages with different length are deleting without error.
2. The exact same block sequence is forwarding with probability 1 and without error.
3. If one block is at its original position but other parts are changed, the transformation is deleting without error.
4. A transformation including a randomized block is deleting with error $1/(2^n - ql)$.
5. For (block-aligned) concatenations of sub-strings of m_i with $i \neq \ell$, we can provide algorithms R that extract the corresponding subsequences and use the provided masks to compute m_ℓ. Thus, the transformation is reconstructible with no error.
6. For a (block-aligned) concatenation containing a block from m_ℓ, this block occurs at a different position (see cases 2 and 3). If the block is *not* the final block, then there are i, j such that $m_{\ell,r+1} \oplus c_{\ell,r} = m_{i,j} \oplus c_{i,j-1}$ with probability $\leq ql/2^n$. Otherwise, the same argument applies to the preceding block and such a transformation is deleting with error $ql^2/2^n$.

Overall, the $\mathcal{F}^{\mathsf{blk}}$-malleable channel is AtE-compatible with error $ql^2/2^n$. □

The security of the AtE-protocol based on CBC encryption and $\varepsilon(q)$-SUF-CMA MAC is shown using Theorem 2.

COROLLARY 2. *Let* CBC *be the CBC protocol based on a URP $\mathbf{P} : \{0,1\}^n \to \{0,1\}^n$ and let* AUT *be the authentication protocol based on a $\varepsilon(q)$-SUF-CMA MAC as in Section 3.3. Then, the AtE-protocol based on* CBC *and* AUT *transforms an insecure channel $\longrightarrow$ into a secure channel $\bullet\!\!\longrightarrow\!\!\bullet$ with error*

$$\frac{(ql)^2}{2^{n-1}} + q\varepsilon(\min(q_A, l)) + \varepsilon(q).$$

The bound we obtain is slightly different from the one given by Krawczyk [17]. This is due to using a generic composition and to the more general modeling of the MAC: In Krawczyk's analysis, the block length of the cipher and the length of the MAC must be equal. In contrast, we restrict neither the size of the MAC nor its position in the authenticated plaintext.

6.3 The Padding Scheme

For CBC mode encryption, TLS uses a padding scheme to ensure that the length of authenticated plaintexts is a multiple of n bits (the input length of the block cipher). Given a plaintext message m with $l_m := |m|$, the length of the padded message generated by the basic version of the TLS padding scheme is $l = n\lceil (l_m + 8)/n \rceil$ and the last $l^{\mathsf{byte}} = (l - l_m)/8$ bytes are filled with the value l^{byte}. The described variant fulfills the condition described in Section 3.3: The padding is injective, and invalidly padded messages are rejected. Yet, TLS additionally allows the padding to extend the message by further blocks, so the argument does not extend to the complete TLS padding scheme.

7. CONCLUSION AND FUTURE WORK

The analysis of secure communication using the paradigm of constructive cryptography [19] brings up natural security definitions for cryptographic tasks such as authenticated or secure transmission of messages. These definitions coincide with previous property-based definitions in some cases, and are different in other cases. In particular, the analysis of the AtE paradigm leads to a general formalization of the malleability of confidential channels and encryption schemes, and the previous notions of non-malleability appear as special instances of the proposed formalization.

In particular, the AtE-transformation is sound for *all* encryption schemes with suitably restricted malleability. This is in contrast to the previous interpretation of the results on AtE, which considered this type of composition as not "generically secure". The encryption schemes used in the TLS protocol fulfill the stated conditions, which is consistent with the previous results by Krawczyk [17].

An interesting question is whether the idea of specifying the malleability of schemes that do not achieve full non-malleability can be applied to further classes of cryptographic protocols.

8. ACKNOWLEDGMENTS

We would like to thank Steven Myers and Stefano Tessaro as well as the anonymous reviewers for helpful and detailed comments on earlier versions of this work.

9. REFERENCES

[1] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In N. Koblitz, editor, *Advances in Cryptology — CRYPTO 1996*, volume 1109 of *LNCS*, pages 1–15. IACR, Springer-Verlag, 1996.

[2] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *Proceedings of the 38th Symposium on Foundations of Computer Science*, pages 394–403. IEEE, 1997.

[3] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *Advances in Cryptology — CRYPTO 1998*, volume 1462 of *LNCS*, pages 26–45. IACR, Springer-Verlag, 1998.

[4] M. Bellare, J. Kilian, and P. Rogaway. The security of the cipher block chaining authentication code. *Journal of Computer and System Sciences*, 61(3):362–399, 2000.

[5] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 531–545. IACR, Springer, 2000. Journal version in [6].

[6] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology*, 21(4):469–491, October 2008.

[7] R. 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, 2001. Extended version in [8].

[8] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067, December 2005.

[9] R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In B. Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 453–474. IACR, Springer-Verlag, 2001.

[10] R. Canetti and H. Krawczyk. Universally composable notions of key exchange and secure channels. In L. R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume 3027 of *LNCS*, pages 337–351. IACR, Springer-Verlag, 2002.

[11] R. Canetti, H. Krawczyk, and J. B. Nielsen. Relaxing chosen-ciphertext security. In D. Boneh, editor, *Advances in Cryptology — CRYPTO 2003*, volume 2729 of *LNCS*, pages 565–582. IACR, Springer-Verlag, 2003.

[12] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) protocol version 1.2. RFC 5246, August 2008.

[13] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

[14] N. Ferguson and B. Schneier. *Practical Cryptography*. Wiley, 2003.

[15] V. D. Gligor, P. Donescu, and J. Katz. On message integrity in symmetric encryption. Manuscript, February 2002.

[16] J. Katz and M. Yung. Unforgeable encryption and chosen ciphertext secure modes of operation. In B. Schneier, editor, *FSE 2000*, volume 1978 of *LNCS*, pages 284–299. IACR, Springer-Verlag, 2000.

[17] H. Krawczyk. The order of encryption and authentication for protecting communications. In J. Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, volume 2139 of *LNCS*, pages 310–331. IACR, Springer-Verlag, 2001.

[18] U. Maurer. Indistinguishability of random systems. In L. R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 110–132. IACR, Springer-Verlag, 2002.

[19] U. Maurer. Constructive cryptography—a primer. In R. Sion, R. Curtmola, S. Dietrich, A. Kiayias, J. M. Miret, K. Sako, and F. Sebé, editors, *Financial Cryptography and Data Security*, volume 6054 of *LNCS*, page 1. Springer-Verlag, 2010.

[20] U. Maurer, K. Pietrzak, and R. Renner. Indistinguishability amplification. In A. Menezes, editor, *Advances in Cryptology — CRYPTO 2007*, volume 4622 of *LNCS*, pages 130–149. Springer-Verlag, Aug. 2007.

[21] U. Maurer and R. Renner. Abstract cryptography. In preparation, 2010.

[22] U. Maurer, R. Renner, and S. Wolf. Unbreakable keys from random noise. In P. Tuyls, B. Škoric, and T. Kevenaar, editors, *Security with Noisy Data*, pages 21–44. Springer-Verlag, 2007.

[23] U. Maurer and P. Schmid. A calculus for security bootstrapping in distributed systems. *Journal of Computer Security*, 4(1):55–80, 1996.

[24] 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. IEEE, 2001.

[25] M. Ray and S. Dispensa. Renegotiating TLS. Preprint, November 2009.

[26] P. Rogaway. Authenticated encryption with associated data. In *ACM Conference on Computer and Communications Security*, pages 98–107. ACM Press, 2002.

[27] P. Rogaway, M. Bellare, and J. Black. OCB: A block-cipher mode of operation for efficient symmetric encryption. *ACM Transactions on Information and System Security (TISSEC)*, 6(3):365–403, August 2003.

[28] S. Vaudenay. Security flaws induced by CBC padding — applications to SSL, IPSEC, WTLS... In L. R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 534–545. IACR, Springer-Verlag, 2002.

[29] D. Whiting, R. Housley, and N. Ferguson. Counter with CBC-MAC (CCM). Submission to NIST, June 2002.