A Constructive Perspective on Key Encapsulation

Sandro Coretti, Ueli Maurer, and Björn Tackmann

Department of Computer Science, ETH Zürich, Switzerland
{corettis,maurer,bjoernt}@inf.ethz.ch

Abstract. A *key-encapsulation mechanism (KEM)* is a cryptographic primitive that allows anyone in possession of some party's public key to securely transmit a key to that party. A KEM can be viewed as a key-exchange protocol in which only a single message is transmitted; the main application is in combination with symmetric encryption to achieve public-key encryption of messages of arbitrary length.
The security of KEMs is usually defined in terms of a certain game that no efficient adversary can win with non-negligible advantage. A main drawback of game-based definitions is that they often do not have clear semantics, and that the security of each higher-level protocol that makes use of KEMs needs to be proved by showing a tailor-made security reduction from breaking the security of the KEM to breaking the security of the combined protocol.
We propose a novel approach to the security and applications of KEMs, following the constructive cryptography paradigm by Maurer and Renner (ICS 2011). The goal of a KEM is to *construct* a resource that models a shared key available to the honest parties. This resource can be used in designing and proving higher-level protocols; the composition theorem guarantees the security of the combined protocol without the need for a specific reduction.

1 Introduction

Key establishment is a cryptographic primitive that allows two parties to obtain a shared secret key, which can subsequently be used in cryptographic mechanisms such as encryption schemes or message authentication codes (MACs). The most important application of key-establishment protocols is in the setup phases of protocols for secure communication, such as TLS or IPSec, but, furthermore, their unidirectional variant—*key-encapsulation mechanisms (KEMs)*—are an important building block in most practical public-key encryption schemes.

This paper is dedicated to Johannes Buchmann on the occasion of his 60^{th} birthday. The topic of the paper, the key-establishment problem, a fundamental problem in cryptography, is one of the areas to which he has contributed significantly (e.g., [3–5, 21]).

In this paper, we focus on the particular case where keys are established using KEMs and only unidirectional communication. We build on [8], where public-key encryption is treated in constructive cryptography, and some parts of this work are taken from that paper.

1.1 Security Notions for Key-Encapsulation Mechanisms

An important question for the application of KEMs is which level of KEM security is required in order for a higher-level protocol that makes use of a KEM to be secure. To define KEM security, game-based security notions for public-key encryption have been adapted to work with KEMs. A game-based definition is usually characterized by a security property that is to be maintained in the presence of an adversary launching a certain attack against the scheme in question. Both security property and attack are encoded only implicitly into the security game. As a consequence, the traditional answer to the above question is that for each protocol one needs to identify the appropriate security notion and provide a reduction proof to show that a KEM satisfying this notion yields a secure protocol.

An alternative approach is to capture the semantics of a security notion by characterizing directly what it achieves, making explicit in which applications it can be used securely. The constructive cryptography framework [15, 16] was proposed with this general goal in mind. Resources such as different types of communication channels and keys are modeled explicitly, and the goal of a cryptographic protocol or scheme π is to *construct* a stronger or more useful resource S from an assumed resource R, denoted as $R \overset{\pi}{\Longrightarrow} S$. Two such construction steps can then be composed, i.e., if we additionally consider a protocol ψ that assumes the resource S and constructs a resource T, the composition theorem states that

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

where $\psi \circ \pi$ denotes the composed protocol.

Following the constructive paradigm, a protocol is built in a modular fashion from isolated construction steps. A security proof guarantees the soundness of one such step, and each proof is independent of the remaining steps. The composition theorem then guarantees that several such steps can be composed. While the general approach to protocol design based on reduction proofs is in principle sound, it is substantially more complex, more error-prone, and not suitable for re-use.

In this spirit, we treat the use of KEMs as such a construction step. We show how one can construct, using KEMs, shared keys from authenticated and insecure channels.

1.2 Constructing Keys using KEMs

From the perspective of constructive cryptography [15, 16], the purpose of a KEM is to construct a shared key from non-confidential communication, which is modeled as channels. Channels and keys are *resources* that involve a sender, a receiver, and—to model different levels of security—an attacker. The security properties of a particular resource are captured by the capabilities available

to the attacker. In the case of a channel they might, e.g., include reading or modifying the messages in transmission.

The parties access resources through the interfaces they provide; these are specific to each party. For example, the sender's interface of a channel allows to input messages, and the receiver's interface allows to receive them. We refer to the interfaces by labels A, B, and E, where A and B are the sender's and the receiver's interfaces, respectively, and E is the adversary's interface. In this work, we consider two types of channels and two types of keys:[1]

- An *insecure channel*, denoted $- \!\!\rightarrow\!\!\!\!\rightarrow$, allows the adversary to read, deliver, and to delete all messages input at A, as well as to inject her own messages.
- An *authenticated channel*, denoted $\bullet\!\!-\!\!\diamond\!\!\rightarrow\!\!\!\!\rightarrow$, still allows to read all messages, but the adversary is limited to forwarding or deleting messages input at interface A.
- A *unilateral key*, denoted $=\!\!\diamond\!\!=\!\!\bullet$, allows A to request keys. The adversary at interface E may choose to deliver or delete keys requested at interface A to B or to inject her own keys.
- A *bilateral key*, denoted $\bullet\!\!=\!\!\diamond\!\!=\!\!\bullet$, also allows A to request keys, but the adversary may not inject any keys.

In the straightforward application of a KEM, the receiver initially generates a key pair and transmits the public key to the sender. The sender needs to obtain the correct public key, which corresponds to assuming that the channel from B to A is authenticated ($\longleftarrow\!\!\bullet$[2]). The sender, using the public key, generates a key and corresponding ciphertext, which he sends to the receiver over a channel that could either be authenticated or completely insecure.

The type of key that is constructed depends on the type of assumed channel used to transmit the ciphertext to the receiver: We show that if the assumed channel is authenticated ($\bullet\!\!-\!\!\diamond\!\!\rightarrow\!\!\!\!\rightarrow$) and the KEM scheme is ind-cpa-secure, the constructed key is bilateral ($\bullet\!\!=\!\!\diamond\!\!=\!\!\bullet$). If the assumed channel is insecure ($- \!\!\rightarrow\!\!\!\!\rightarrow$) and the PKE scheme is ind-cca-secure, the constructed key is unilateral ($=\!\!\diamond\!\!=\!\!\bullet$).

Using the above notation, for protocols π and π' based on ind-cpa and ind-cca KEMs, respectively, these constructions can be written as

$$[\longleftarrow\!\!\bullet, \bullet\!\!-\!\!\diamond\!\!\rightarrow\!\!\!\!\rightarrow] \quad \overset{\pi}{\Longrightarrow} \quad \bullet\!\!=\!\!\diamond\!\!=\!\!\bullet$$

and

$$[\longleftarrow\!\!\bullet, - \!\!\rightarrow\!\!\!\!\rightarrow] \quad \overset{\pi'}{\Longrightarrow} \quad =\!\!\diamond\!\!=\!\!\bullet,$$

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

[1] The notation is taken from [18]: The "$\bullet$" in the notation signifies that the capabilities at the marked interface, i.e., sending or receiving, are exclusive to the respective party. If the "$\bullet$" is missing, the adversary also has these capabilities. The $\diamond$-symbol is explained in Section 2.4.

[2] The simple arrow indicates that $\longleftarrow\!\!\bullet$ is a single-use channel, i.e., only one message can be transmitted.

The notion of constructing the bilateral or unilateral keys from the two assumed non-confidential channels is made precise in a simulation-based sense [16, 15], where the simulator can be interpreted as translating all attacks on the protocol into attacks on the constructed (ideal) key. As the constructed key is secure by definition, there are no attacks on the protocol.

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

1.3 Related Work

Game-based security. The notion of KEMs was introduced by Cramer and Shoup [9, 22], when they proposed an ISO standard for public-key encryption (PKE). They adapted the standard security notions for PKE schemes to work with KEMs. Today's standard security notions for PKE are indistinguishability under chosen-plaintext attack (ind-cpa) and (the stronger) indistinguishability under chosen-ciphertext attack (ind-cca). Bellare et al. [2] compare them to various other PKE security notions. Herranz et al. [12] analyze the security of PKE schemes obtained from KEMs in conjunction with symmetric encryption schemes with various security levels.

Real-world/ideal-world security. The idea of defining protocol security with respect to an ideal execution was first proposed by Goldreich et al. [10]; the concept of a simulator can be traced back to the seminal work by Goldwasser et al. [11] on zero-knowledge proofs. Canetti [6] introduced the universal composability framework (UC), which allows the formalization of arbitrary functionalities to be realized by cryptographic protocols. The framework is designed from a bottom-up perspective (starting from a selected machine model), whereas we follow the top-down approach of [16], which leads to simpler, more abstract definitions and statements. Nagao et al. [20] give a treatment of KEMs in the UC framework. They show that combining a cca-secure KEM with cca-secure symmetric encryption realizes an "ideal KEM" functionality, which can be used to implement the secure channel functionality of [7].

The constructive cryptography paradigm has been introduced by Maurer and Renner [16, 15]. Several primitives have been modeled and analyzed following this paradigm, including symmetric encryption [17] and public-key encryption [8]. Similarly to this work, both papers model the primitives as constructions and compare the resulting security definitions to (game-based) notions from the literature. The unilateral key resource we use here is described in [19], where it is

shown to be achievable by an interactive protocol based on (signatures and) a cpa-secure KEM.

2 Preliminaries

2.1 Systems: Resources, Converters, Distinguishers, and Reductions

Resources and converters are modeled as systems. At the highest level of abstraction (following the hierarchy in [16]), systems are objects with interfaces by which they connect to (interfaces of) other systems; each interface is labeled with an element of a label set and connects to only a single other interface. This concept, which we refer to as *abstract systems*, captures the topological structures that result when multiple systems are connected in this manner.

The abstract systems concept, however, does not model the behavior of systems, i.e., *how* the systems interact via their interfaces. Consequently, statements about cryptographic protocols are statements at the next (lower) abstraction level. In this work, we describe all systems in terms of (probabilistic) discrete systems, which we explain in Section 2.2.

Resources and converters. *Resources* in this work are systems with three interfaces labeled by A, B, and E. A protocol is modeled as a pair of two so-called *converters* (one for each honest party), which are directed in that they have an *inside* and an *outside* interface, denoted by `in` and `out`, respectively. As a notational convention, we generally use upper-case, bold-face letters (e.g., $\mathbf{R}$, $\mathbf{S}$) or channel symbols (e.g., $\bullet\!\!-\!\!\diamond\!\!-\!\!\twoheadrightarrow$) to denote resources and lower-case Greek letters (e.g., α, β) or sans-serif fonts (e.g., snd, rcv) for converters. We denote by Φ the set of all resources and by Σ the set of all converters.

The topology of a composite system is described using a term algebra, where each expression starts from one (or more) resources on the right-hand side and is subsequently extended with further terms on the left-hand side. An expression is interpreted in the way that all interfaces of the system it describes can be connected to interfaces of systems which are appended on the left. For instance, for a single resource $\mathbf{R} \in \Phi$, all its interfaces A, B, and E are accessible.

For $I \in \{A, B, E\}$, a resource $\mathbf{R} \in \Phi$, and a converter $\alpha \in \Sigma$, the expression $\alpha^I\mathbf{R}$ denotes the composite system obtained by connecting the inside interface of α to interface I of $\mathbf{R}$; the outside interface of α becomes the I-interface of the composite system. The system $\alpha^I\mathbf{R}$ is again a resource (cf. Figure 1 on page 10).

For two resources $\mathbf{R}$ and $\mathbf{S}$, $[\mathbf{R}, \mathbf{S}]$ denotes the parallel composition of $\mathbf{R}$ and $\mathbf{S}$. For each $I \in \{A, B, E\}$, the I-interfaces of $\mathbf{R}$ and $\mathbf{S}$ are merged and become the *sub-interfaces* of the I-interface of $[\mathbf{R}, \mathbf{S}]$, which we denote by $I.1$ and $I.2$. A converter α that connects to the I-interface of $[\mathbf{R}, \mathbf{S}]$ has two inside sub-interfaces, denoted by $\mathtt{in}.1$ and $\mathtt{in}.2$, where the first one connects to $I.1$ of $\mathbf{R}$ and the second one connects to $I.2$ of $\mathbf{S}$.

Any two converters α and β can be composed sequentially by connecting the inside interface of β to the outside interface of α, written $\beta \circ \alpha$, with the effect

that $(\beta \circ \alpha)^I \mathbf{R} = \beta^I \alpha^I \mathbf{R}$. Moreover, converters can also be taken in parallel, denoted by $[\alpha, \beta]$, with the effect that $[\alpha, \beta]^I [\mathbf{R}, \mathbf{S}] = [\alpha^I \mathbf{R}, \beta^I \mathbf{S}]$.

We assume the existence of an identity converter $\mathsf{id} \in \Sigma$ with $\mathsf{id}^I \mathbf{R} = \mathbf{R}$ for all resources $\mathbf{R} \in \Phi$ and interfaces $I \in \{A, B, E\}$ and of a special converter $\bot \in \Sigma$ with an inactive outside interface.

Distinguishers. A *distinguisher* is a special type of system $\mathbf{D}$ that connects to all interfaces of a resource $\mathbf{U}$ and outputs a single bit at the end of its interaction with $\mathbf{U}$. In the term algebra, this appears as the expression $\mathbf{DU}$, which defines a binary random variable. The *distinguishing advantage of a distinguisher* $\mathbf{D}$ *on two systems* $\mathbf{U}$ *and* $\mathbf{V}$ is defined as

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

The advantage of a class $\mathcal{D}$ of distinguishers is defined as

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

The distinguishing advantage measures how much the output distribution of $\mathbf{D}$ differs when it is connected to either $\mathbf{U}$ or $\mathbf{V}$. There is an equivalence notion on systems (which is defined on the discrete systems level), denoted by $\mathbf{U} \equiv \mathbf{V}$, which implies that $\Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V}) = 0$ for all distinguishers $\mathbf{D}$. The distinguishing advantage satisfies the triangle inequality, i.e., $\Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{W}) \leq \Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V}) + \Delta^{\mathbf{D}}(\mathbf{V}, \mathbf{W})$ for all resources $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ and distinguishers $\mathbf{D}$.

Games. We capture games defining security properties as distinguishing problems in which an adversary $\mathbf{A}$ tries to distinguish between two *game systems* $\mathbf{G}_0$ and $\mathbf{G}_1$. Game systems (or simply *games*) are single-interface systems, which appear, similarly to resources, on the right-hand side of the expressions in the term algebra. The adversary is similar to a distinguisher, except that it connects to a game instead of a resource.

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

To reduce distinguishing two systems $\mathbf{S}, \mathbf{T}$ to distinguishing two systems $\mathbf{U}, \mathbf{V}$, one exhibits a reduction $\mathbf{C}$ such that $\mathbf{CS} \equiv \mathbf{U}$ and $\mathbf{CT} \equiv \mathbf{V}$.[3] Then, for all distinguishers $\mathbf{D}$, we have $\Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V}) = \Delta^{\mathbf{D}}(\mathbf{CS}, \mathbf{CT}) = \Delta^{\mathbf{DC}}(\mathbf{S}, \mathbf{T})$. The last equality follows from the fact that $\mathbf{C}$ can also be thought of as being part of the distinguisher.

[3] For instance, we consider reductions from distinguishing game systems to distinguishing resources. Then, $\mathbf{C}$ connects to a game on the inside and provides interfaces A, B, and E on the outside.

2.2 Discrete Systems

Protocols that communicate by passing messages and the respective resources are described as (probabilistic) discrete systems. Their behavior can be formalized by random systems as in [14], i.e., as families of conditional probability distributions of the outputs (as random variables) given all previous inputs and outputs of the system. For systems with multiple interfaces, the interface to which an input or output is associated is explicitly specified as part of the input or output. For the restricted (but here sufficient) class of systems that for each input provide (at most) a single output, an execution of a collection of systems is defined as the consecutive evaluation of the respective random systems (this is similar to the models in [6, 13]).

2.3 The Notion of Construction

Recall that we consider resources with interfaces A, B, and E, where A and B are interfaces of honest parties and E is the interface of the adversary. We formalize the security of protocols via the following notion of *construction* (which is a special case of the abstraction notion from [16]):

Definition 1. *Let Φ and Σ be as in Section 2.1. A protocol $\pi = (\pi_1, \pi_2) \in \Sigma^2$* constructs resource $\mathbf{S} \in \Phi$ from resource $\mathbf{R} \in \Phi$ within ε and with respect to distinguisher class $\mathcal{D}$, *denoted*

$$\mathbf{R} \quad \overset{(\pi,\varepsilon)}{\Longrightarrow} \quad \mathbf{S},$$

if

$$\begin{cases} \Delta^{\mathcal{D}}(\pi_1^A \pi_2^B \perp^E \mathbf{R}, \perp^E \mathbf{S}) \leq \varepsilon & (\textit{availability}) \\ \exists \sigma \in \Sigma: \quad \Delta^{\mathcal{D}}(\pi_1^A \pi_2^B \mathbf{R}, \sigma^E \mathbf{S}) \leq \varepsilon & (\textit{security}). \end{cases}$$

The availability condition captures that a protocol must correctly implement the functionality of the constructed resource in the absence of the adversary. The security condition models the requirement that everything the adversary can achieve in the *real-world system* (i.e., the assumed resource with the protocol) he can also accomplish in the *ideal-world system* (i.e., the constructed resource with the simulator). More details can be found in [15].

An important property of Definition 1 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. Security and availability are preserved under composition. More formally, if for some resources $\mathbf{R}$, $\mathbf{S}$, and $\mathbf{T}$ and protocols π and ϕ,

$$\mathbf{R} \quad \overset{(\pi,\varepsilon)}{\Longrightarrow} \quad \mathbf{S} \qquad \text{and} \qquad \mathbf{S} \quad \overset{(\phi,\varepsilon')}{\Longrightarrow} \quad \mathbf{T},$$

then

$$\mathbf{R} \quad \overset{(\phi \circ \pi, \varepsilon + \varepsilon')}{\Longrightarrow} \quad \mathbf{T},$$

as well as

$$[\mathbf{R},\mathbf{U}] \overset{([\pi,\mathsf{id}],\varepsilon)}{\Longmapsto} [\mathbf{S},\mathbf{U}] \quad \text{and} \quad [\mathbf{U},\mathbf{R}] \overset{([\mathsf{id},\pi],\varepsilon)}{\Longmapsto} [\mathbf{U},\mathbf{S}]$$

for any resource $\mathbf{U}$. More details can be found in [15].

2.4 Important Resources

In this work, we consider two types of resources: channels and keys. Both types initially expect a special cheating bit $b \in \{0,1\}$ at interface E, indicating whether the adversary is present and intends to interfere with the transmission of the messages. The special converter $\perp$ (cf. Section 2.1) always sets $b = 0$, in which case all messages/keys input at the sender's interface are delivered immediately to the receiver. For simplicity, we will assume that whenever $\perp$ is not present, all cheating bits are set to 1.

An *authenticated channel* from A to B with message space $\mathcal{M}$ is a resource $\bullet\!\!-\!\!\diamond\!\!\!\rightarrow$ with interfaces A, B, and E and behaves as follows: When the i^{th} message $m \in \mathcal{M}$ is input at interface A, it is recorded as (i,m) and (msg,i,m) is output at interface E. When (dlv,i') is input at interface E, if (i',m') has been recorded, m' is delivered at interface B. An *insecure channel* $-\!\!\!\rightarrow$ behaves as $\bullet\!\!-\!\!\diamond\!\!\!\rightarrow$, but, additionally, when (inj,m') is input at interface E, m' is output at interface B.

A *bilateral key* between A and B with key space $\mathcal{K}$ is a resource $\bullet\!\!=\!\!\diamond\!\!=\!\!\bullet$ with interfaces A, B, and E and behaves as follows: When req is input for the i^{th} time at interface A, a key $k \leftarrow_R \mathcal{K}$ is chosen uniformly at random and is recorded as (i,k). Then, k is output at interface A and (key,i) at interface E. When (dlv,i') is input at interface E, if (i',k') has been recorded, k' is output at interface B. A *unilateral key* $=\!\!\diamond\!\!=\!\!\bullet$ behaves as $\bullet\!\!=\!\!\diamond\!\!=\!\!\bullet$, but, additionally, when (inj,k') is input at interface E, k' is output at interface B.[4]

For $\mathbf{R} \in \{-\!\!\!\rightarrow, \bullet\!\!-\!\!\diamond\!\!\!\rightarrow, =\!\!\diamond\!\!=\!\!\bullet, \bullet\!\!=\!\!\diamond\!\!=\!\!\bullet\}$, denote by $\overset{n}{\mathbf{R}}$ the resource that processes only the first n queries at A and E (e.g., $\overset{n}{\bullet\!\!-\!\!\diamond\!\!\!\rightarrow}$ only processes the first n messages input at A and the first n queries at E).

2.5 Key-Encapsulation Mechanisms

A key-encapsulation mechanism (KEM) with key space $\mathcal{K}$ is a triple of algorithms $\Pi = (K,E,D)$. The key generation algorithm K outputs a key pair $(\mathsf{pk},\mathsf{sk}) \leftarrow K$, the (probabilistic) encryption algorithm takes a public key pk and outputs a pair $(k,c) \leftarrow E_{\mathsf{pk}}$, where $k \in \mathcal{K}$ and c is the corresponding ciphertext, and the decryption algorithm takes a secret key sk and a ciphertext c and outputs $k' \leftarrow D_{\mathsf{sk}}(c')$, where $k' = \diamond$ indicates an invalid ciphertext.

A KEM is correct if for $(k,c) \leftarrow E_{\mathsf{pk}}$, $D_{\mathsf{sk}}(c) = k$ (with probability 1 over the randomness of E) for all key pairs $(\mathsf{pk},\mathsf{sk})$ generated by K. For security

[4] Note that neither the channels nor the keys prevent the adversary from reordering or replaying messages/keys sent by A. The $\diamond$-symbol suggests the "internal buffer" in which a channel/key stores messages input at A.

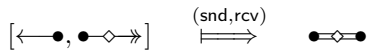
properties of KEM schemes which are defined via a bit-guessing game, it will be more convenient to phrase the game as a distinguishing problem between two game systems (cf. Section 2.1). We consider the following games, which correspond to the (standard) notions of ind-cpa (cpa for short) and ind-cca2 (cca).[5]

CPA game. Consider systems $\mathbf{G}_0^{\mathsf{cpa}}$ and $\mathbf{G}_1^{\mathsf{cpa}}$: For a KEM scheme Π, both initially run the key-generation algorithm to obtain $(\mathsf{pk}, \mathsf{sk})$ and output pk. When the query chall is input, they compute $(k, c) \leftarrow E_{\mathsf{pk}}$. Then, $\mathbf{G}_0^{\mathsf{cpa}}$ outputs (k, c) and $\mathbf{G}_1^{\mathsf{cpa}}$ outputs $(\bar{k}, c)$ for a randomly chosen $\bar{k} \in \mathcal{K}$. Ciphertext c is called the *challenge*.

CCA game. Consider systems $\mathbf{G}_0^{\mathsf{cca}}$ and $\mathbf{G}_1^{\mathsf{cca}}$: They proceed as in the CPA case but also answer decryption queries (dec, c') by returning $k' \leftarrow D_{\mathsf{sk}}(c')$ unless c' equals the challenge c (if defined), in which case the answer is test.

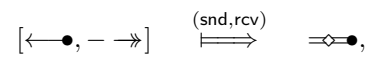
3 Bilateral and Unilateral Keys from KEMs

In the spirit of constructive cryptography, we explicitly consider the purpose of KEMs, namely to achieve a shared key between A and B over an untrusted network. As a constructive statement this means that we view the KEM as a protocol, a pair of converters $(\mathsf{snd}, \mathsf{rcv})$, whose goal is to construct a shared key from non-confidential channels. Differentiating between the two cases where the communication from the sender A to the receiver B is authenticated and unauthenticated, this is written as



(1)

and



(2)

respectively.

In both cases, the channel $\longleftarrow\!\bullet$ captures the ability of the sender to obtain the receiver's public key in an authenticated fashion. In construction (1), the communication from the sender A to the receiver B is authenticated, which is modeled by the channel $\bullet\!\!-\!\!\diamond\!\!-\!\!\twoheadrightarrow$. The goal is to achieve a bilateral key $\bullet\!\!=\!\!\diamond\!\!=\!\!\bullet$. In construction (2), the communication from A to B is completely insecure, which is captured by the insecure channel $-\!\!\twoheadrightarrow$. Here, the goal is to achieve a unilateral key $=\!\!\diamond\!\!=\!\!\bullet$, which does not prevent the adversary from sending its own keys to B.

In the following we first show how a KEM Π can be seen as a converter pair $(\mathsf{snd}, \mathsf{rcv})$. We then prove that $(\mathsf{snd}, \mathsf{rcv})$ achieves construction (1) if Π is cpa-secure, and construction (2) if Π is cca-secure.

[5] We consider the so-called real-or-random versions of these games, which are equivalent to the more popular left-or-right formulations (as shown in [1] for symmetric encryption).

3.1 KEMs as Protocols

Let $\Pi = (K, E, D)$ be a KEM. Based on Π, we define a pair of protocol converters $(\mathsf{snd}, \mathsf{rcv})$ for constructions (1) and (2). Both converters have two subinterfaces, denoted by $\mathtt{in}.1$ and $\mathtt{in}.2$, on the inside, as we connect them to a resource that is a parallel composition of two other resources (cf. Section 2.1).

Converter snd works as follows: It initially expects a public key pk at $\mathtt{in}.1$. When req is input at the outside interface, snd computes $(k, c) \leftarrow E_{\mathsf{pk}}$ and outputs k at the outside interface and c at the inside interface $\mathtt{in}.2$. Converter rcv initially generates a key pair $(\mathsf{pk}, \mathsf{sk})$ using key-generation algorithm K and outputs pk at $\mathtt{in}.1$. When rcv receives c' at $\mathtt{in}.2$, it computes $k' \leftarrow D_{\mathsf{sk}}(c')$ and, if $k' \neq \diamond$, outputs k' at the outside interface.

3.2 Bilateral Keys from Authenticated Channels

Towards proving that the protocol $(\mathsf{snd}, \mathsf{rcv})$ indeed achieves construction (1), note first that the correctness of the KEM immediately implies that the *availability* condition of Definition 1 is satisfied. To prove *security*, we need to exhibit a simulator σ such that the assumed resource $[\leftarrow\!\bullet, \bullet\!\diamond\!\!\rightarrow]$ with the protocol converters is indistinguishable from the constructed resource $\bullet\!\!=\!\diamond\!\!=\!\bullet$ with the simulator (cf. Figure 1). Theorem 1 implies that $(\mathsf{snd}, \mathsf{rcv})$ realizes (1) if the underlying KEM is cpa-secure.

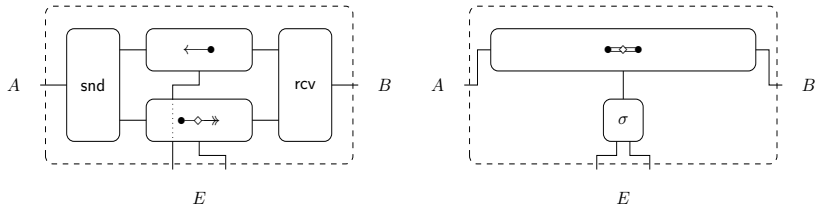


Fig. 1. Left: The assumed resource (two authenticated channels) with protocol converters snd and rcv attached to interfaces A and B, denoted $\mathsf{snd}^A\mathsf{rcv}^B[\leftarrow\!\bullet, \bullet\!\diamond\!\!\rightarrow]$. Right: The constructed resource (a bilateral key) with simulator σ attached to the E-interface, denoted $\sigma^E \bullet\!\!=\!\diamond\!\!=\!\bullet$. In particular, σ must simulate the E-interfaces of the two authenticated channels. The protocol is secure if the two systems are indistinguishable.

Theorem 1. *There exists a simulator σ and for any $n \in \mathbb{N}$ there exists a (efficient) reduction $\mathbf{C}$ such that for every $\mathbf{D}$,*

$$\Delta^{\mathbf{D}}(\mathsf{snd}^A\mathsf{rcv}^B[\leftarrow\!\bullet, \bullet\overset{n}{\diamond}\!\!\rightarrow], \sigma^E \bullet\overset{n}{=}\diamond\overset{n}{=}\bullet) \leq n \cdot \Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{cpa}}, \mathbf{G}_1^{\mathsf{cpa}}).$$

Proof. Consider the following simulator σ for interface E of $\bullet\!\!=\!\diamond\!\!=\!\bullet$: Initially, it generates a key pair $(\mathsf{pk}, \mathsf{sk})$ and outputs $(\mathsf{msg}, 1, \mathsf{pk})$ at $\mathtt{out}.1$. On (key, i) at the

inside interface $\mathtt{in}$, it generates $(k, c) \leftarrow E_{\mathsf{pk}}$ and outputs (msg, i, c) at $\mathtt{out}.2$. On (dlv, i') at $\mathtt{out}.2$, it outputs (dlv, i') at $\mathtt{in}$. Consider the two systems

$$\mathsf{snd}^A\mathsf{rcv}^B[\leftarrow\!\bullet, \bullet\!\overset{1}{\diamond}\!\twoheadrightarrow] \qquad \text{and} \qquad \sigma^E \bullet\!\overset{1}{\diamond}\!\twoheadrightarrow\bullet\,.$$

Distinguishing $\mathbf{G}_0^{\mathsf{cpa}}$ from $\mathbf{G}_1^{\mathsf{cpa}}$ can be reduced to distinguishing these two systems via the following reduction system $\mathbf{C}'$, which connects to a game on the inside and provides interfaces A, B, and E on the outside (cf. Section 2.1 for details on reduction systems): Initially, it takes a value pk on the inside and outputs $(\mathsf{msg}, 1, \mathsf{pk})$ at the (outside) $E.1$-interface. On req at the A-interface, it outputs chall at $\mathtt{in}$. On subsequent key and challenge (k, c) at $\mathtt{in}$, it outputs k at A and $(\mathsf{msg}, 1, c)$ at $E.2$. On $(\mathsf{dlv}, 1)$ at the $E.2$-interface, it outputs k at interface B. We have

$$\mathbf{C}'\mathbf{G}_0^{\mathsf{cpa}} \equiv \mathsf{snd}^A\mathsf{rcv}^B[\leftarrow\!\bullet, \bullet\!\overset{1}{\diamond}\!\twoheadrightarrow] \qquad \text{and} \qquad \mathbf{C}'\mathbf{G}_1^{\mathsf{cpa}} \equiv \sigma^E \bullet\!\overset{1}{=\!\diamond\!=}\!\bullet,$$

and thus

$$\begin{aligned}
&\Delta^{\mathbf{D}}(\mathsf{snd}^A\mathsf{rcv}^B[\leftarrow\!\bullet, \bullet\!\overset{n}{\diamond}\!\twoheadrightarrow], \sigma^E \bullet\!\overset{n}{=\!\diamond\!=}\!\bullet) \\
&\leq n{\cdot}\Delta^{\mathbf{DC}''}(\mathsf{snd}^A\mathsf{rcv}^B[\leftarrow\!\bullet, \bullet\!\overset{1}{\diamond}\!\twoheadrightarrow], \sigma^E \bullet\!\overset{1}{=\!\diamond\!=}\!\bullet) \\
&= n{\cdot}\Delta^{\mathbf{DC}''}(\mathbf{C}'\mathbf{G}_0^{\mathsf{cpa}}, \mathbf{C}'\mathbf{G}_1^{\mathsf{cpa}}) \\
&= n{\cdot}\Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{cpa}}, \mathbf{G}_1^{\mathsf{cpa}}),
\end{aligned}$$

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

3.3 Unilateral Keys from Unauthenticated Channels

We now prove that $(\mathsf{snd}, \mathsf{rcv})$ achieves (2). Note again that the correctness of the KEM implies that the *availability* condition of Definition 1 is satisfied. To prove *security*, we need to exhibit a simulator σ such that the assumed resource $[\leftarrow\!\bullet, - \twoheadrightarrow]$ with the protocol converters is indistinguishable from the constructed resource $=\!\diamond\!=\!\bullet$ with the simulator. Theorem 1 implies that $(\mathsf{snd}, \mathsf{rcv})$ realizes (2) if the underlying KEM is cca-secure.

Theorem 2. *There exists a simulator σ and for any $n \in \mathbb{N}$ there exists a (efficient) reduction $\mathbf{C}$ such that for every $\mathbf{D}$,*

$$\Delta^{\mathbf{D}}(\mathsf{snd}^A\mathsf{rcv}^B[\leftarrow\!\bullet, -\overset{n}{\twoheadrightarrow}], \sigma^E =\!\overset{n}{\diamond}\!=\!\bullet) \leq n \cdot \Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{cca}}, \mathbf{G}_1^{\mathsf{cca}}).$$

Proof. Simulator σ for interface E of $=\!\diamond\!=\!\bullet$ again has two outside sub-interfaces $\mathtt{out}.1$ and $\mathtt{out}.2$ and works as follows: Initially, it generates a key pair $(\mathsf{pk}, \mathsf{sk})$ and outputs $(\mathsf{msg}, 1, \mathsf{pk})$ at $\mathtt{out}.1$. When it receives (key, i) at the inside interface $\mathtt{in}$, it generates $(k, c) \leftarrow E_{\mathsf{pk}}$, outputs (msg, i, c) at $\mathtt{out}.1$, and records (c, i). When (inj, c') is input at $\mathtt{out}.2$, σ proceeds as follows: If (c', i') has been recorded for

some i', it outputs (dlv, i') at $\mathtt{in}$. Otherwise, it computes $k' \leftarrow D_{\mathsf{sk}}(c')$ and, if $k' \neq \diamond$, outputs (inj, k') at $\mathtt{in}$.

Denote by $\overset{n,q}{-\!\!\twoheadrightarrow}$ the insecure channel that processes the first n inputs at interface A and the first q inputs at interface E (and similarly for $\overset{n,q}{=\!\!\diamond\!\!\bullet}$). Consider now the problem of distinguishing the two systems

$$\mathbf{U} := \mathsf{snd}^A \mathsf{rcv}^B[\longleftarrow\!\!\bullet, \overset{1,n}{-\!\!\twoheadrightarrow}] \qquad \text{and} \qquad \mathbf{V} := \sigma^E \overset{1,n}{=\!\!\diamond\!\!\bullet},$$

which are depicted in Figure 2. A distinguisher $\mathbf{D}$ connected to the real-world system $\mathbf{U}$ initially sees a public key at interface $E.1$. If $\mathbf{D}$ inputs req at interface A, a key k is output at A and its corresponding encryption is output at interface $E.2$ (both created by snd). When $\mathbf{D}$ inputs a ciphertext c' at $E.2$, it sees a decryption of c' (by rcv) at B. The ideal-world system $\mathbf{V}$ behaves differently: Initially, $\mathbf{D}$ also sees a public key. But when it inputs req at A, an encryption c of a key k unrelated to the key $\bar{k}$ output at A is output at interface $E.2$ (by simulator σ). When c is input at interface $E.2$, $\bar{k}$ is output at B (as σ issues a dlv-instruction to the key). When $c' \neq c$ is input at E, the decryption k' of c' (injected by σ) is output at B.

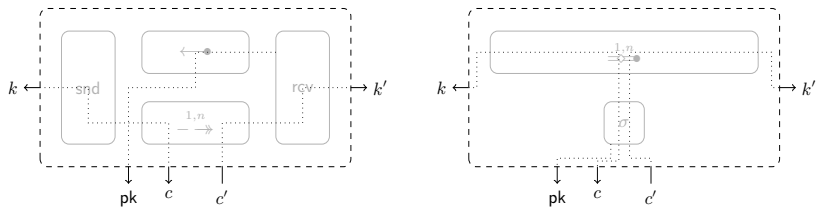


Fig. 2. The systems $\mathbf{U}$ and $\mathbf{V}$ with the "message flow" from the perspective of a distinguisher: Initially, a public-key pk is output at interface $E.1$. Requesting a key k at interface A causes a ciphertext c to be output at the $E.2$-interface. Note that c is the challenge in the cca-game. Inputting a ciphertext c' at interface $E.2$ results in a key k' being output at B. This corresponds to the decryption oracle in the cca-game.

The translation between the channel setting and the game setting is achieved by the following reduction system $\mathbf{C}'$: Initially, $\mathbf{C}'$ takes a value pk from the game and outputs $(\mathsf{msg}, 1, \mathsf{pk})$ at the $E.1$-interface. When req is input at interface A of $\mathbf{C}'$, chall is output to the game, which returns (k, c). Key k is output at A and challenge c is output as $(\mathsf{msg}, 1, c)$ at interface $E.2$. When (inj, c) is input at interface $E.2$, $\mathbf{C}'$ outputs k at interface B. When (inj, c') with $c' \neq c$ is input at interface $E.2$, $\mathbf{C}'$ passes c' to the game's decryption oracle and outputs the answer k' at interface B, provided $k' \neq \diamond$. We have

$$\mathbf{C}'\mathbf{G}_0^{\mathsf{cca}} \equiv \mathsf{snd}^A \mathsf{rcv}^B[\longleftarrow\!\!\bullet, \overset{1,n}{-\!\!\twoheadrightarrow}] \qquad \text{and} \qquad \mathbf{C}'\mathbf{G}_1^{\mathsf{cca}} \equiv \sigma^E \overset{1,n}{=\!\!\diamond\!\!\bullet},$$

and thus

$$\begin{aligned}\Delta^{\mathbf{D}}(\mathsf{snd}^A\mathsf{rcv}^B[\leftarrow\!\!\bullet, -\overset{n}{\twoheadrightarrow}], \sigma^E \overset{n}{=\!\diamond\!=\!\bullet}) &\leq n \cdot \Delta^{\mathbf{DC}''}(\mathsf{snd}^A\mathsf{rcv}^B[\leftarrow\!\!\bullet, -\overset{1,n}{\twoheadrightarrow}], \sigma^E \overset{1,n}{=\!\diamond\!=\!\bullet}) \\ &= n \cdot \Delta^{\mathbf{DC}''}(\mathbf{C}'\mathbf{G}_0^{\mathsf{cca}}, \mathbf{C}'\mathbf{G}_1^{\mathsf{cca}}) \\ &= n \cdot \Delta^{\mathbf{DC}}(\mathbf{G}_0^{\mathsf{cca}}, \mathbf{G}_1^{\mathsf{cca}}),\end{aligned}$$

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

The unilateral key $-\!\diamond\!\twoheadrightarrow\!\bullet$ is the best key one can construct from the two assumed channels. As the E-interface has the same capabilities as the A-interface at both the authenticated (from B to A) and the insecure channels, it will necessarily also be possible to inject keys to the receiver via the E-interface by simply applying the sender's protocol converter.

4 Conclusion

The purpose of this paper is to describe KEMs as a construction step that can be used within higher-level protocols. While the most widespread application is in the so-called KEM-DEM public-key encryption schemes, KEMs are also used as a method to establish keys in interactive protocols such as TLS.

The constructive approach allows to fully isolate the analysis of the KEM schemes from the analysis of the higher-level protocols that use the obtained keys; protocols that are proven secure assuming a (unilateral or bilateral) shared key remain secure if the respective key is obtained using a KEM. Hence, the approach supports a fully modular protocol design where each cryptographic mechanism is proven in isolation to achieve one construction step, and multiple such steps are composed by the general composition theorem.

Compared with the traditional approach of explicitly proving tailor-made security reductions from breaking the security of the underlying schemes to breaking the security of a complex protocol, this modular approach leads to security proofs which are simpler and consequently less prone to errors.

Acknowledgments. The work was supported by the Swiss National Science Foundation (SNF), project no. 200020-132794.

References

1. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A Concrete Security Treatment of Symmetric Encryption. In: 38th FOCS. pp. 394–403 (1997)
2. 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, Heidelberg (1998)
3. Buchmann, J., Düllmann, S., Williams, H.C.: On the Complexity and Efficiency of a New Key Exchange System. In: Quisquater, J.J., Vandewalle, J. (eds.) EUROCRYPT '89. LNCS, vol. 434, pp. 597–616. Springer, Heidelberg (1989)

4. Buchmann, J., Williams, H.C.: A Key-Exchange System Based on Imaginary Quadratic Fields. Journal of Cryptology 1(2), 107–118 (1988)
5. Buchmann, J., Williams, H.C.: A Key Exchange System Based on Real Quadratic Fields. In: Brassard, G. (ed.) CRYPTO '89. LNCS, vol. 435, pp. 335–343. Springer, Heidelberg (1989)
6. Canetti, R.: Universally Composable Security: A New Paradigm for Cryptographic Protocols. Cryptology ePrint Archive, Report 2000/067 (2000)
7. Canetti, R., Krawczyk, H.: Universally Composable Notions of Key Exchange and Secure Channels. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 337–351. Springer, Heidelberg (2002)
8. Coretti, S., Maurer, U., Tackmann, B.: Constructing Confidential Channels from Authenticated Channels—Public-Key Encryption Revisited. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, Springer, Heidelberg (2013)
9. Cramer, R., Shoup, V.: Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. SIAM Journal on Computing 33, 167–226 (2001)
10. Goldreich, O., Micali, S., Wigderson, A.: How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. In: 19th ACM STOC. pp. 218–229 (1987)
11. Goldwasser, S., Micali, S., Rackoff, C.: The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract). In: 17th ACM STOC. pp. 291–304 (1985)
12. Herranz, J., Hofheinz, D., Kiltz, E.: Some (in)sufficient conditions for secure hybrid encryption. Cryptology ePrint Archive, Report 2006/265 (2006), http://eprint.iacr.org/
13. Hofheinz, D., Shoup, V.: GNUC: A New Universal Composability Framework. Cryptology ePrint Archive, Report 2011/303 (2011)
14. Maurer, U.: Indistinguishability of Random Systems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 110–132. Springer, Heidelberg (2002)
15. Maurer, U.: Constructive Cryptography—A New Paradigm for Security Definitions and Proofs. In: Moedersheim, S., Palamidessi, C. (eds.) TOSCA 2011. LNCS, vol. 6993, pp. 33–56. Springer, Heidelberg (Apr 2011)
16. Maurer, U., Renner, R.: Abstract Cryptography. In: Chazelle, B. (ed.) The Second Symposium in Innovations in Computer Science, ICS 2011. pp. 1–21. Tsinghua University Press (Jan 2011)
17. Maurer, U., Rüedlinger, A., Tackmann, B.: Confidentiality and Integrity: A Constructive Perspective. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 209–229. Springer, Heidelberg (2012)
18. Maurer, U., Schmid, P.E.: A Calculus for Security Bootstrapping in Distributed Systems. Journal of Computer Security 4(1), 55–80 (1996)
19. Maurer, U., Tackmann, B., Coretti, S.: Key Exchange with Unilateral Authentication: Composable Security Definition and Modular Protocol Design. Cryptology ePrint Archive, Report 2013/555 (2013)
20. Nagao, W., Manabe, Y., Okamoto, T.: A Universally Composable Secure Channel Based on the KEM-DEM Framework. In: TCC. pp. 426–444 (2005)
21. Scheidler, R., Buchmann, J., Williams, H.C.: Implementation of a Key Exchange Protocol Using Some Real Quadratic Fields. In: Damgård, I. (ed.) EUROCRYPT '90. LNCS, vol. 473, pp. 98–109. Springer, Heidelberg (1990)
22. Shoup, V.: A Proposal for an ISO Standard for Public Key Encryption. IACR Cryptology ePrint Archive 2001, 112 (2001)