Sender-binding Key Encapsulation

Laurin Benz[1,2], Wasilij Beskorovajnov[3], Sarai Eilebrecht[3], Jörn Müller-Quade[1,2,3], Astrid Ottenhues[1,2], and **Rebecca Schwerdt**[1,2]

[1] Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
[2] KASTEL Security Research Labs, Karlsruhe, Germany
{laurin.benz, mueller-quade, ottenhues, schwerdt}@kit.edu
[3] FZI Research Center for Information Technology, Karlsruhe, Germany
{beskorovajnov, eilebrecht}@fzi.de

Abstract. Secure communication is gained by combining encryption with authentication. In real-world applications encryption commonly takes the form of KEM-DEM hybrid encryption, which is combined with ideal authentication. The pivotal question is how weak the employed key encapsulation mechanism (KEM) is allowed to be to still yield universally composable (UC) secure communication when paired with symmetric encryption and ideal authentication. This question has so far been addressed for public-key encryption (PKE) only, showing that encryption does not need to be stronger than sender-binding CPA, which binds the CPA secure ciphertext non-malleably to the sender ID. For hybrid encryption, prior research unanimously reaches for CCA2 secure encryption which is unnecessarily strong. Answering this research question is vital to develop more efficient and feasible protocols for real-world secure communication and thus enable more communication to be conducted securely.
In this paper we use ideas from the PKE setting to develop new answers for hybrid encryption. We develop a new and significantly weaker security notion—sender-binding CPA for KEMs—which is still strong enough for secure communication. By using game-based notions as building blocks, we attain secure communication in the form of ideal functionalities with proofs in the UC-framework. Secure communication is reached in both the classic as well as session context by adding authentication and one-time/replayable CCA secure symmetric encryption respectively. We furthermore provide an efficient post-quantum secure LWE-based construction in the standard model giving an indication of the real-world benefit resulting from our new security notion. Overall we manage to make significant progress on discovering the minimal security requirements for hybrid encryption components to facilitate secure communication.

Keywords: IND-SB-CPA · Key Encapsulation · Secure Communication · Authenticated Channels · UC.

1 Introduction

Secure communication has always been the first and foremost goal of cryptography. The common way to reach this goal is to combine encryption with authentication.

Development on the encryption side has come a long way from the roots of symmetric encryption schemes via public-key encryption (PKE) [1] to modern hybrid encryption [2], where keys are exchanged via a public-key key encapsulation mechanism (KEM) and subsequently used to symmetrically encrypt messages.

For secure communication via PKE it has long been known that CCA2 secure encryption is unnecessarily strong if authentication is already provided [3]. A recent breakthrough in this setting [4] showed that sender-binding encryption (SBE) and IND-SB-CPA security are the right concepts to realize secure message transfer from authenticated channels in the universal composability (UC) model. SBE is a PKE adaption which binds the ciphertext to the sender ID.The authors of [4], however, only consider PKE while real world applications have moved on to hybrid encryption.

In the field of hybrid encryption, on the other hand, the question of how strong (or weak) encryption should be for secure communication has been completely ignored. Constructing indistinguishability under adaptive chosen ciphertext attack (IND-CCA2) secure PKE is seen as the only significant goal, regardless of the fact that in practice encryption schemes are then usually paired with authentication via digital signatures to gain secure communication.

We bridge the gap between these two worlds by bringing sender-binding ideas to real world efficient KEM-DEM hybrid encryption [2, 5]:

Our Contribution. Our contribution includes an adaptation of the concept of SBE from the PKE to the KEM setting, yielding the notion of sender-binding key encapsulation mechanism (SB-KEM) with corresponding IND-SB-CPA security. We prove IND-SB-CPA$_{\text{SB-KEM}}$[4] security to be the weakest so far—other than plain CPA security—by investigating its relation to previously proposed (tag-)KEM security notions. Furthermore we show that IND-SB-CPA$_{\text{SB-KEM}}$ security is in fact the KEM equivalent of IND-SB-CPA$_{\text{SBE}}$. This directly leads us to the proof that IND-SB-CPA$_{\text{SB-KEM}}$ is still strong enough to facilitate secure communication via the KEM-DEM principle over authenticated channels. We present the security proofs both for the single-message setting as well as for session communication, resulting in the ideal functionalities of secure message transfer and secure channels respectively. Lastly we indicate the potential practical benefit of our theoretic advancements by giving a concrete IND-SB-CPA secure SB-KEM construction. Our construction is a simplified version of the recently proposed and, as far as we know, currently most efficient KEM construction in the standard model [6]. Overall we manage to provide a new and weaker–but still sufficiently strong–security notion for the public-key encryption part of secure communication which could lead to efficiency gains. The different parts of our contribution can be viewed in Figure 1. They are distributed throughout this paper as follows:

- In Section 4 we adapt the concept of SBE and IND-SB-CPA$_{\text{SBE}}$ to the KEM setting, developing SB-KEM and IND-SB-CPA$_{\text{SB-KEM}}$. We furthemore

[4] When not obvious, the type of scheme a security notion pertains to is given in subscript.

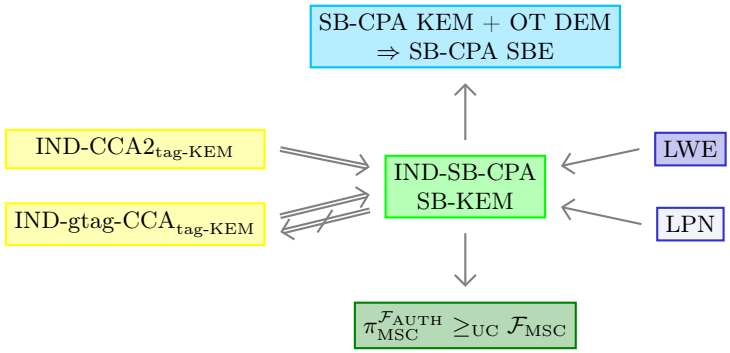

Fig. 1. Overview of Our Contribution[5]

highlight some relations to other KEM security notions in Section 4. In Appendix A we furthermore provide a generic transformation from dual receiver key encapsulation mechanism (DR-KEM) which is analogous to the SBE construction from dual receiver encryption (DRE). This aids us in separating IND-SB-CPA$_{\text{SB-KEM}}$ from IND-gtag-CCA$_{\text{tag-KEM}}$.

- In Section 5 we prove that IND-SB-CPA secure KEM can be combined with a one-time (OT) secure data encapsulation mechanism (DEM) to gain IND-SB-CPA$_{\text{SBE}}$. This in turn UC-realizes secure message transfer when authenticated channels are added.
- For Section 6 we switch from the classic setting to the setting of session communication and prove that IND-SB-CPA$_{\text{SB-KEM}}$ in conjuction with IND-CCA2$_{\text{DEM}}$ (or IND-RCCA$_{\text{DEM}}$) and authenticated channels UC-realize secure channels. This is an improvement over the results of [7] which needed CCA2 security from both the KEM and DEM component.
- In Section 7 we present an efficient post-quantum secure SB-KEM construction based on the standard learning with errors (LWE) assumption in the standard model and prove it to be IND-SB-CPA$_{\text{SB-KEM}}$ secure. In Appendix B we furthermore propose an efficient learning parity with noise (LPN) based construction.

2 Preliminaries

We start by providing some basic knowledge needed to understand our research. This includes an introduction to the KEM-DEM- as well as UC-framework and definitions of various game-based and ideal functionality security notions. Readers who are already familiar with these topics might want to skip this section and only come back to it later if they want to look something up.

[5] Duck or rabbit?

2.1 The KEM-DEM Framework

First, we briefly recap the KEM-DEM framework which was introduced in [5] and subsequently included in the encryption ISO standard in 2006 [2, 8]. The KEM-DEM framework is a special form of hybrid encryption which combines the advantages of both public-key and symmetric encryption: The symmetric encryption of messages makes encryption more efficient while the KEM public key infrastructure alleviates the need for a key exchange protocol. In particular, the KEM-DEM framework consists of two modular components. The first component is a public-key key encapsulation mechanism (KEM) which generates a symmetric key and encrypts it, while the second component is a symmetric data encapsulation mechanism (DEM) which uses this symmetric key to encrypt a message:

Definition (KEM): A *key encapsulation mechanism (KEM)* is given by a set of three probabilistic polynomial time (PPT) algorithms $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ with

$$\mathtt{gen} : 1^\lambda \mapsto (sk, pk), \qquad \mathtt{enc} : pk \mapsto (K, C), \qquad \mathtt{dec} : (sk, C) \mapsto K$$

such that the correctness property holds, i.e. $K = \mathtt{dec}(sk, C)$ whenever $(sk, pk) \leftarrow \mathtt{gen}(1^\lambda)$ and $(K, C) \leftarrow \mathtt{enc}(pk)$.

Definition (DEM): A *data encapsulation mechanism (DEM)* is given by a set of two PPT algorithms $(\mathtt{DEM.enc}, \mathtt{DEM.dec})$ with $\mathtt{DEM.enc} : (K, m) \mapsto c$ and $\mathtt{DEM.dec} : (K, c) \mapsto m$ such that $m = \mathtt{DEM.dec}(K, c)$ whenever $c \leftarrow \mathtt{DEM.enc}(K, m)$ (correctness).

The KEM-DEM framework comes in two flavors which slightly differ in the combination of the KEM and DEM. One construction–which we call *single-message* communication– generates a fresh symmetric key for each encryption of a message. This is the original definition of the KEM-DEM framework and intuitively yields a PKE scheme $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ where $\mathtt{Gen} \equiv \mathtt{gen}$ and

$\mathtt{Enc}(pk, m)$:
- $(K, C) \leftarrow \mathtt{enc}(pk)$.
- $c \leftarrow \mathtt{DEM.enc}(K, m)$.

$\hookrightarrow$ Return (C, c).

$\mathtt{Dec}(sk, (C, c))$:
- $K \leftarrow \mathtt{dec}(sk, C)$.
- $m \leftarrow \mathtt{DEM.dec}(K, c)$.

$\hookrightarrow$ Return m.

For *Session* communication on the other hand, one party (who does not need a KEM key pair themselves) generates a persistent symmetric key via the KEM and sends it to the communication partner once. This symmetric session key is then used for many messages between the two involved parties.

Tag-KEMs. A slight variation of classical KEMs are tag-key encapsulation mechanisms (tag-KEMs) [9] which additionally use tag to encapsulate and decapsulate the symmetric key. The encapsulation phase of the tag-KEM is split in two separate phases: A first phase that generates the symmetric key and a second phase that encapsulates the given symmetric key using the tag. The split is made to allow for the tag to depend on the symmetric key itself.

Definition (Tag-KEM): A *tag-KEM* is given by a set of four PPT algorithms $(\mathtt{gen}, \mathtt{key}, \mathtt{enc}, \mathtt{dec})$ with

$$\mathtt{gen}: \quad 1^\lambda \mapsto (sk, pk) \qquad \mathtt{key}: \quad pk \mapsto (K, aux)$$
$$\mathtt{enc}: \quad (\tau, aux) \mapsto C \qquad \mathtt{dec}: \quad (sk, \tau, C) \mapsto K$$

such that the correctness property holds, i.e. $K = \mathtt{dec}(sk, \tau, C)$ whenever $(sk, pk) \leftarrow \mathtt{gen}(1^\lambda)$, $(K, aux) \leftarrow \mathtt{key}(pk)$ and $C \leftarrow \mathtt{enc}(\tau, aux)$.

When introducing tag-KEMs, Abe et al. [9] use them in a slightly modified tag-KEM-DEM framework where the symmetrically encrypted message is used as the tag for encapsulation which allows for a weaker DEM to be used.

2.2 Game-based Security Notions

In this section we recap previously defined game-based security notions used in this paper. First we give definitions for PKE schemes, then KEM and finally DEM schemes. Whenever it is not immediately obvious for which type of scheme a security notion is intended, we denote it in its index, e.g. IND-CCA2$_{\text{PKE}}$.

IND-SB-CPA$_{\mathbf{SBE}}$. The PKE security notion which inspired this whole paper is called IND-SB-CPA and was recently introduced by Beskorovajnov et al. [4]. We use this notion as a basis for the new KEM security definition we introduce in Section 4. IND-SB-CPA$_{\text{SBE}}$ security pertains to the special PKE case of SBE where both encryption and decryption take the ID S of the encrypting (or sending) party as additional input, binding a ciphertext not only to the receiver (via their public key) but to the sender as well. The intuition behind IND-SB-CPA$_{\text{SBE}}$ security is that an adversary may be able to modify the message content of ciphertexts arbitrarily but is not able to change a ciphertext such that it is bound to a party ID other than that of the sender or receiver. More formally:

Definition (IND-SB-CPA$_{\mathbf{SBE}}$): An SBE scheme $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ with set of party IDs $\mathbf{P}$ satisfies *indistinguishability under sender-binding chosen plaintext attack (IND-SB-CPA)* (cf. [4]), iff for any PPT adversary $\mathcal{A}_{\text{SB-CPA}}$ the advantage to win the IND-SB-CPA game in Figure 2 is negligible in security parameter λ.

IND-gtag-CCA$_{\mathbf{TBE}}$ Tag-based encryption (TBE) [10] is closely related to SBE. Instead of party IDs the tags given to both encryption and decryption are taken from a dedicated tag space $\mathbf{T}$. For our paper we only need the weakest notion so far proposed for TBE–IND-gtag-CCA–which we later on adapt to KEMs to develop a better understanding of how strong (or rather weak) IND-SB-CPA$_{\text{SB-KEM}}$ is in comparison to other notions. The following definition is taken from [4].

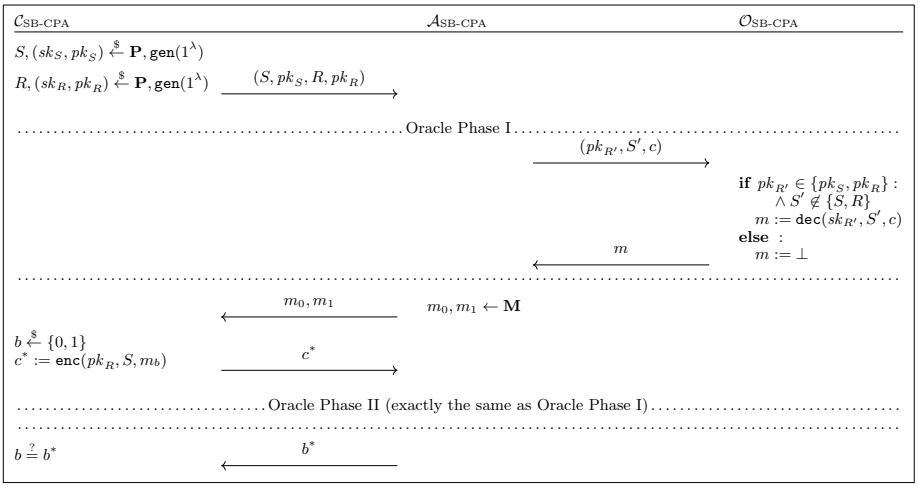

Fig. 2. The IND-SB-CPA$_{\text{SBE}}$ Game for SBE from [4].

Definition (IND-gtag-CCA$_{\text{TBE}}$): A TBE scheme $\Sigma = (\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ with tag space $\mathbf{T}$ satisfies *indistinguishability under given-tag weakly chosen ciphertext attack (IND-gtag-CCA)*, iff for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ the advantage

$$\mathsf{Adv}^{\text{gtag-CCA}}_{\mathcal{A},\Sigma}(\lambda) := \Big|\mathbb{P}\Big[b \leftarrow \mathcal{A}_2^{\mathcal{O}^*}(c^*, aux) \,\Big|\, \tau^* \xleftarrow{\$} \mathbf{T}; (sk, pk) \leftarrow \mathtt{gen}(1^\lambda); (aux, m_0, m_1) \leftarrow \mathcal{A}_1^{\mathcal{O}^*}(pk, \tau^*); b \xleftarrow{\$} \{0,1\}; c^* \leftarrow \mathtt{enc}(pk, \tau^*, m_b)\Big] - \tfrac{1}{2}\Big|$$

is negligible in λ, where $\mathcal{O}^*(\tau, c)$ returns $\bot$ for $\tau = \tau^*$ and $\mathtt{dec}(sk, \tau, c)$ otherwise.

IND-CCA2$_{\text{tag-KEM}}$ The following definition was taken from [9]. Note that in [9] there is a first oracle phase where the adversary only has pk as prior input. Since the adversary has equal oracle access for the second phase and only gains additional input inbetween (rather than making any output themselves), the first oracle phase is redundant and we choose to present the notion without it.

Definition (IND-CCA2$_{\text{tag-KEM}}$): A tag-KEM $\Sigma = (\mathtt{gen}, \mathtt{key}, \mathtt{enc}, \mathtt{dec})$ satisfies *IND-CCA2*, iff for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ the advantage

$$\mathsf{Adv}^{\text{CCA2}}_{\mathcal{A},\Sigma}(\lambda) := \Big|\mathbb{P}\Big[b \leftarrow \mathcal{A}_2^{\mathcal{O}^*}(C^*, aux_{\mathcal{A}}) \,\Big|\, (sk, pk) \leftarrow \mathtt{gen}(1^\lambda); (aux, K_0) \leftarrow \mathtt{key}(pk); K_1 \xleftarrow{\$} \{0,1\}^{|K_0|}; b \xleftarrow{\$} \{0,1\}; (\tau^*, aux_{\mathcal{A}}) \leftarrow \mathcal{A}_1^{\mathcal{O}}(pk, K_b); C^* \leftarrow \mathtt{enc}(aux, \tau^*)\Big] - \tfrac{1}{2}\Big|$$

is negligible in λ, where $\mathcal{O}$ denotes $\mathtt{dec}(sk, \cdot, \cdot)$ and $\mathcal{O}^*(\tau, C)$ returns $\bot$ for $(\tau, C) = (\tau^*, C^*)$ and $\mathtt{dec}(sk, \tau, C)$ otherwise.

For symmetric private-key security notions we follow the more descriptive notation of [11]. With the also commonly used PX-CY notation of [12], one-time attack (OT) corresponds to P0-C0, while CCA2 corresponds to P2-C2.

IND-OT$_{\mathbf{DEM}}$. The OT notion for DEMs is an even weaker security notion than classic CPA, as it does not even provide the adversary with an encryption oracle. We use this notion later on in Section 5 in combination with an IND-SB-CPA$_{\text{SB-KEM}}$ secure KEM to realize secure message transfer.

Definition (IND-OT$_{\mathbf{DEM}}$): A DEM $\Sigma = (\mathtt{DEM.enc}, \mathtt{DEM.dec})$ satisfies *indistinguishability under one-time attack (IND-OT)*, iff for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ the following advantage is negligible in λ:

$$\mathsf{Adv}^{\mathrm{OT}}_{\mathcal{A},\Sigma}(\lambda) := \Big|\mathbb{P}\Big[b \leftarrow \mathcal{A}_2(c^*, aux) \;\Big|\; K \xleftarrow{\$} \{0,1\}^{n(\lambda)}; (m_0, m_1, aux) \leftarrow \mathcal{A}_1(1^\lambda); \\ b \xleftarrow{\$} \{0,1\}; c^* \leftarrow \mathtt{enc}(K, m_b)\Big] - \tfrac{1}{2}\Big|.$$

IND-CCA2$_{\mathbf{DEM}}$ and IND-RCCA$_{\mathbf{DEM}}$. Session communication–where each symmetric key may be used more than once–requires stronger DEMs. We therefore recap the private key CCA2 security notion as well and formulate a replayable chosen ciphertext attack (RCCA) DEM notion corresponding to the respective PKE notion [3]. The intuition behind RCCA lies in replayability. This means an adversary is allowed to be able to modify ciphertexts to other valid ciphertexts as long as the message content is not changed, e.g. via rerandomization.

Definition (IND-CCA2$_{\mathbf{DEM}}$, IND-RCCA$_{\mathbf{DEM}}$): A DEM $\Sigma = (\mathtt{DEM.enc}, \mathtt{DEM.dec})$ satisfies *IND-CCA2*, iff for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ the advantage

$$\mathsf{Adv}^{\mathrm{CCA2}}_{\mathcal{A},\Sigma}(\lambda) := \Big|\mathbb{P}\Big[b \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathtt{enc}}, \mathcal{O}^*_{\mathtt{dec}}}(c^*, aux) \;\Big|\; K \xleftarrow{\$} \{0,1\}^{n(\lambda)}; \\ (m_0, m_1, aux) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathtt{enc}}, \mathcal{O}_{\mathtt{dec}}}(1^\lambda); \\ b \xleftarrow{\$} \{0,1\}; c^* \leftarrow \mathtt{enc}(K, m_b)\Big] - \tfrac{1}{2}\Big|$$

is negligible in λ, where $\mathcal{O}_{\mathtt{enc}}$ denotes the oracle $\mathtt{DEM.enc}(K, \cdot)$, $\mathcal{O}_{\mathtt{dec}}$ denotes $\mathtt{DEM.dec}(K, \cdot)$ and $\mathcal{O}^*_{\mathtt{dec}}(c)$ returns $\perp$ for $c = c^*$ and $\mathtt{DEM.dec}(K, c)$ otherwise.

The notion of *IND-RCCA* for DEMs differs only in the definition of $\mathcal{O}^*_{\mathtt{dec}}$, which returns $\perp$ whenever $\mathcal{O}_{\mathtt{dec}}(c) \in \{m_0, m_1\}$.

Now that we are familiar with all these game-based definitions let us jump to the parallel world of simulation-based and in particular UC security.

2.3 Simulation-based Security and UC

As we have seen in Section 2.2, game-based security notions are attack-centered. A scheme fulfills a game-based security notion if and only if one specific attack (e.g.,

distinguishing which message is contained in a ciphertext) can never be successful in specific circumstances (e.g., without oracle access). While this is a nice way to model simple and isolated properties, it is not easy to comprehensively define the security of real-world scenarios which usually require multiple interrelated properties and are conducted concurrent with other protocols. For this purpose simulation-based security and in particular universal composability (UC) were developed. We briefly introduce both concepts in this section, more details can be found in [13] and [14] respectively.

With simulation-based security, properties are not captured in individual games but the whole scenario is modeled as an ideal process which inherently captures all properties at once. This ideal process is called an *ideal functionality* $\mathcal{F}$ and can be thought of as a trusted third party which is handed all inputs of all parties via ideal secure channels, honestly conducts the actual protocol and distributes outputs again in an ideally secure way. Any adversarial powers to influence this process are specified within the ideal functionality and therefore explicitly known. Functionalities for different purposes are distinguished by name, while different instances of the same functionality are distinguished via session IDs *sid*. Security with respect to an ideal functionality $\mathcal{F}$ means that a protocol π solves the given problem *at least as securely* as the ideal functionality does. More concretely: Any real adversary $\mathcal{A}$ attacking an execution of the protocol can be simulated by some simulator $\mathcal{S}$ in an interaction with the ideal functionality such that the two are computationally indistinguishable. In this case the protocol π is said to *securely realize* the functionality $\mathcal{F}$.

UC security is a form of simulation-based security which is even stricter. Not only do transcripts of $\text{EXEC}_{\pi,\mathcal{A}}$ and $\text{IDEAL}_{\mathcal{F},\mathcal{S}}$ of the protocol and ideal experiment have to be computationally indistinguishable, but the distinguisher–called environment $\mathcal{Z}$–adaptively provides inputs to and receives outputs from the protocol parties trying to make protocol and ideal functionality diverge. The adaptivity of $\mathcal{Z}$ also means that standard techniques like rewinding are not feasible in the UC setting. The bright sight of this additional work is that UC secure protocols remain secure under arbitrary and concurrent composition (hence the name) while the same is not true in the classic stand-alone simulation-based security. The following definition stems from [14] with the exact formulation taken from [4]. It captures UC security more formally:

Definition (UC Security): Let $\mathcal{F}$ be an ideal functionality and π a protocol. We say that π *UC-realizes* the ideal functionality $\mathcal{F}$, iff for any PPT adversary $\mathcal{A}$ there is a PPT simulator $\mathcal{S}$ such that no PPT environment $\mathcal{Z}$ can distinguish $\text{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}$ from $\text{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ with non-negligible probability. In this case we write $\pi \geq_{\text{UC}} \mathcal{F}$.

Having two adversarial entities $\mathcal{Z}$ and $\mathcal{A}$ can be slightly hard to follow, but as Canetti showed in [14] we can equivalently consider an adversarial environment $\mathcal{Z}$ while reducing the adversary $\mathcal{A}$ to a mere dummy $\mathcal{D}$.

With this general knowledge of UC security we can go on to recap some specific ideal functionalities.

2.4 Ideal Functionalities

In this section we formally define the authenticated and secure channel functionalities $\mathcal{F}_{\text{AUTH}}$ and $\mathcal{F}_{\text{MSC}}$ we use in this paper. Although the secure message transfer (SMT) functionality $\mathcal{F}_{\text{MSMT}}$ is somewhat central to Section 5, it is sufficient to know that it does capture SMT. The detailed inner workings (see [4]) are not necessary to understand this paper. Furthermore we briefly encounter functionalities $\mathcal{F}_{\text{KEM}}$, $\mathcal{F}_{\text{KEM-DEM}}$, $\mathcal{F}_{\text{SIG}}$, $\mathcal{F}_{\text{CA}}$ and $\mathcal{F}_{\text{CERT}}$ in Section 3 but again do not require further details. Interested readers can find them in [7] and [15].

For PKE schemes, SMT functionalities are commonly used to model secure communication [14, 3, 4]. But for session communication we follow the lead of [16, 7] and use a secure channel functionality instead. This yields the same level of message security but is specifically designed for communication in sessions. The classic definition of $\mathcal{F}_{\text{SC}}$ can be found in [16]. For our proof in Section 6 we instead use an (equivalent) multi-session version $\mathcal{F}_{\text{MSC}}$, where some abort possibilities of the adversary (implicitly present in $\mathcal{F}_{\text{SC}}$) are made explicit as well.

$\mathcal{F}_{\mathbf{MSC}}$

Provides:
Multiple secure two-party communication sessions.

State:

- Active function $f_{\text{act}}\colon \mathbf{SID} \times \{\{A, B\} \mid A, B \in \mathbf{P}\} \to \{\mathsf{true}, \mathsf{false}, \mathtt{init}\}$ initialized to $f_{\text{act}} \equiv \mathsf{false}$.
- Function $p_{\text{Msg}} : \mathbf{SID} \times \mathbf{MID} \to \mathbf{M} \times \mathbf{P}^2$ of pending messages.

Behaviour:

- Upon receiving $(\mathtt{init}, sid, B)$ from some party A, set $f_{\text{act}}(sid, \{A, B\}) := \mathtt{init}$ and send $(\mathtt{inited}, sid, A, B)$ to the adversary $\mathcal{A}$.
- Upon receiving $(\mathtt{establish}, sid, A)$ from party B, check $f_{\text{act}}(sid, \{A, B\}) = \mathtt{init}$, set $f_{\text{act}}(sid, \{A, B\}) := \mathsf{true}$ and send $(\mathtt{established}, sid, A, B)$ to $\mathcal{A}$.
- Upon receiving $(\mathtt{send}, sid, R, m)$ from some party S, check $f_{\text{act}}(sid, \{S, R\}) = \mathsf{true}$, draw fresh mid, send $(\mathtt{send}, sid, mid, S, R)$ to the adversary $\mathcal{A}$ and append $(sid, mid) \mapsto (m, S, R)$ to p_{Msg}.
- Upon receiving $(\mathtt{send\ ok}, sid, mid)$ from the adversary look up $(m, S, R) := p_{\text{Msg}}(sid, mid)$. If it exists, and if $f_{\text{act}}(sid, \{S, R\}) = \mathsf{true}$, output $(\mathtt{sent}, sid, S, m)$ to R.
- Upon receiving $(\mathtt{expire}, sid, B)$ from some party A, set $f_{\text{act}}(sid, \{A, B\}) := \mathsf{false}$.

Once the notational differences are ignored there is only one distinction between $\mathcal{F}_{\text{SC}}$ and $\mathcal{F}_{\text{MSC}}$: $\mathcal{F}_{\text{MSC}}$ allows for multiple communication sessions between different pairs of communication partners within one instance (i.e. with the same *sid*) while a new $\mathcal{F}_{\text{SC}}$ instance (*sid*) is needed for each communication session.

Everything else is identical. For normal settings, where arbitrarily many sessions between arbitrary communication partners are allowed, multiple instances (or their multi-session extensions) are needed of both $\mathcal{F}_{\text{SC}}$ and $\mathcal{F}_{\text{MSC}}$ and the sole difference lies in whether or not a new *sid* is used for a new session.

$\mathcal{F}_{\text{AUTH}}$

Provides:
Single-receiver single-message single-sender authenticated message transfer with constant message size.
Behaviour:

- Upon invocation with input $(\texttt{send}, sid, R, m)$ from some party S, send backdoor message $(\texttt{send}, sid, S, R, m)$ to the adversary $\mathcal{A}$.
- Upon receiving $(\texttt{send ok}, sid)$ from adversary $\mathcal{A}$: If output not yet generated, then output $(\texttt{sent}, sid, S, R, m)$ to R.
- Ignore all further inputs.

With these previously known definitions fresh in our minds we go on to give some more context to our paper in the next section by discussing prior works.

3 Related Work

While there are lots of papers pertaining to the general topic of hybrid encryption via the KEM-DEM framework, most of the works focus on more efficient constructions of KEMs, such as the hybrid encryption scheme by Cramer and Shoup [5], the Kurosawa-Desmedt-KEM [17] or the newly standardized Kyber-KEM [18]. For this section, however, we stay with the main contribution of our paper and instead consider those papers which give proofs on what levels of secure communication can be reached with various KEM and DEM security notions.

As mentioned in Section 2.1, there are two branches of KEM-DEM-based hybrid encryption: Single-message and session communication. We start in the more common single-message setting. With one symmetric key per message IND-CCA2$_{\text{PKE}}$ has always been seen as the goal to construct secure communication. Hence the main security analysis is usually conducted by constructing an IND-CCA2$_{\text{PKE}}$ secure PKE scheme from successively weaker (and more efficient) KEM and DEM notions. When originally introducing the KEM-DEM framework, Shoup showed that combining a KEM and DEM which satisfy the respective notions of IND-CCA2 security yields an IND-CCA2$_{\text{PKE}}$ secure PKE as a result [2, 5]. Also in [5] it was shown that if one relaxes the security of the DEM to one-time-IND-CCA2$_{\text{DEM}}$ security (sometimes called IND-OTCCA [11]), the construction still suffices for an IND-CCA2 secure PKE as each symmetric key will only be used once. In [11], Herranz, Hofheinz, and Kiltz give an overview of all previously proposed game-based KEM and DEM security notions and comprehensively identify which combinations lead to which security notions for

the resulting PKE. One main finding was that CCA2 security could so far *only* be reached via a CCA2 secure KEM in conjunction with (one-time-)CCA2 DEM. All other combinations result in less secure PKE schemes. Kurosawa and Desmedt managed to present a KEM-DEM construction for an IND-CCA2$_{\text{PKE}}$ scheme in [17] where the employed KEM scheme is not IND-CCA2 secure [19]. However, it was shown in [20] that the Kurosawa-Desmedt-KEM is not far off, as it becomes IND-CCA2$_{\text{KEM}}$ secure with a slight twist. Abe et al. modify the KEM-DEM framework to a new tag-KEM-DEM framework [9] (cf. Section 2.1). They show that for this type of hybrid encryption an IND-CCA2 secure KEM together with an only IND-OT secure DEM yields an IND-CCA2 PKE as well. They also show that the aforementioned Kurosawa-Desmedt-KEM can be considered a tag-KEM in which case it actually satisfies IND-CCA2$_{\text{tag-KEM}}$ security. A similar not quite IND-CCA2 secure KEM construction was used in [6] as well.

In contrast to these works we employ the results of [3, 4] which state that IND-CCA2$_{\text{PKE}}$ is unnecessarily strong to realize secure communication and hence do not try to construct an IND-CCA2 secure PKE in this paper. Aiming for the weaker but sufficient notion of IND-SB-CPA$_{\text{SBE}}$ security [4], we develop the corresponding KEM notion of IND-SB-CPA$_{\text{SB-KEM}}$. We show that in combination with the weakest possible DEM–satisfying only IND-OT security–our new notion still provides IND-SB-CPA$_{\text{SBE}}$ security for the SBE scheme constructed via the classic KEM-DEM framework. Using such a weak DEM scheme was previously only possible via the more complex tag-KEM-DEM framework. Furthermore we show that if our SB-KEM is viewed as a (simpler) version of a tag-KEM, the KEM security notion IND-SB-CPA, which we introduce in this work, is strictly weaker than the IND-CCA2$_{\text{tag-KEM}}$ notion employed in the tag-KEM-DEM framework.

Although Canetti and Krawczyk consider various UC and non-UC security notions for key exchange and session key security in [21, 16], Nagao, Manabe, and Okamoto were the first to take the KEM-DEM framework into the world of UC security [7]. They also make the switch to session communication where each symmetric key is used not only for multiple messages but bi-directional communication as well. Nagao, Manabe, and Okamoto firstly introduce an ideal functionality $\mathcal{F}_{\text{KEM}}$ capturing the security intuitively expected from a KEM and prove a generic protocol to UC-realize $\mathcal{F}_{\text{KEM}}$ if and only if the KEM used in the protocol is IND-CCA2$_{\text{KEM}}$ secure. In a second step a complete KEM-DEM functionality $\mathcal{F}_{\text{KEM-DEM}}$ is defined and similarly shown that it is realized by a generic DEM-protocol in the $\mathcal{F}_{\text{KEM}}$-hybrid model if and only if the DEM satisfies IND-CCA2$_{\text{DEM}}$ security. Lastly it is shown that using $\mathcal{F}_{\text{KEM-DEM}}$ in conjunction with the signature and certification functionalities $\mathcal{F}_{\text{SIG}}$ and $\mathcal{F}_{\text{CA}}$ suffices without any other cryptographic building blocks to realize a single-session bi-directional secure channel $\mathcal{F}_{\text{SC}}$. An overview of this process is shown in Figure 3.

The additional functionalities $\mathcal{F}_{\text{SIG}}$ and $\mathcal{F}_{\text{CA}}$ are used for authentication during the key exchange and could equally be substituted by $\mathcal{F}_{\text{AUTH}}$, as it was shown in [15] that such a use of signatures combined with certification already UC-realizes $\mathcal{F}_{\text{AUTH}}$. The two equivalences between respective CCA2 security notions and ideal KEM and KEM-DEM functionalities from [7] could be taken to

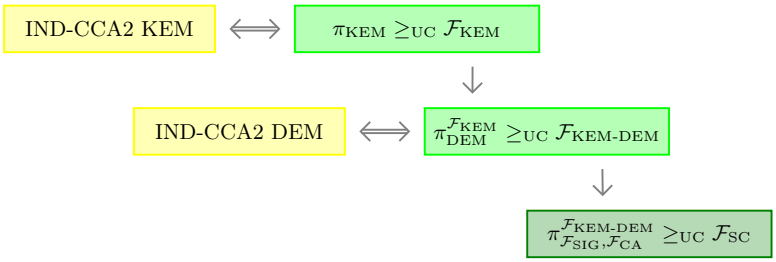

Fig. 3. Overview of Secure Channel Realization from [7].

indicate that CCA2 is a necessary condition for achieving secure channels via the session KEM-DEM framework. In this paper we show that this is not actually true. The main factor to realize here is that authentication in the form of $\mathcal{F}_{\text{SIG}}$ and $\mathcal{F}_{\text{CA}}$ is only used for the key exchange and added after the fact to the KEM-DEM functionality to realize $\mathcal{F}_{\text{SC}}$. Directly using the KEM on an authenticated channel and binding the key ciphertext to the sender lets us achieve the same level of security with significantly less security requirements on the KEM. For our proof in Section 6 we skip the detour via $\mathcal{F}_{\text{KEM}}$ and $\mathcal{F}_{\text{KEM-DEM}}$ and directly show that an IND-SB-CPA$_{\text{SB-KEM}}$ secure KEM combined with $\mathcal{F}_{\text{AUTH}}$ and an IND-CCA2$_{\text{DEM}}$ secure DEM UC-realize a secure channel.

4 Sender-binding Key Encapsulation

In this section we develop the security notion of IND-SB-CPA$_{\text{KEM}}$ and give some transformations to show its relation to other KEM security notions. Before doing so, we first introduce what it means for a KEM to be called sender-binding.

Definition (SB-KEM): A *sender-binding key encapsulation mechanism (SB-KEM)* is given by a set of three PPT algorithms $(\texttt{gen}, \texttt{enc}, \texttt{dec})$ with

$$\texttt{gen} : 1^\lambda \mapsto (sk, pk), \qquad \texttt{enc} : (pk, S) \mapsto (K, C), \qquad \texttt{dec} : (sk, S, C) \mapsto K$$

such that the correctness property holds, i.e. $K = \texttt{dec}(sk, S, C)$ whenever $(sk, pk) \leftarrow \texttt{gen}(1^\lambda)$ and $(K, C) \leftarrow \texttt{enc}(pk, S)$.

Note that so far, this is only the traditional KEM interface enhanced by a party ID as input for encapsulation and decryption. Although the denomination suggests this, the "sender" and "binding" part only become meaningful with the respective security notion. Any classic KEM instantly satisfies this definition when its input is adjusted to incorporate a party ID, regardless of whether this ID specifies some sender, receiver or just a random party, regardless of whether there is any binding property or the ID can be easily exchanged, even regardless of whether this ID is used at all in the protocol. The intended use, however, is that the sending or encapsulating party inserts its *own* ID upon encapsulation, this ID

is then non-malleably bound to an otherwise malleable ciphertext and decryption is only successful if the *same* ID is used. These properties are expressed in the following IND-SB-CPA$_{\text{KEM}}$ notion. The idea for this notion comes from the corresponding SBE notion introduced in [4], which we adapt to fit the KEM setting.

Definition (IND-SB-CPA$_{\text{SB-KEM}}$): An SB-KEM $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ satisfies *indistinguishability under sender-binding chosen plaintext attack (IND-SB-CPA)* security, iff for any PPT adversary $\mathcal{A}_{\text{SB-CPA}}$ the probability to win the IND-SB-CPA game shown in Figure 4 is negligible in λ.

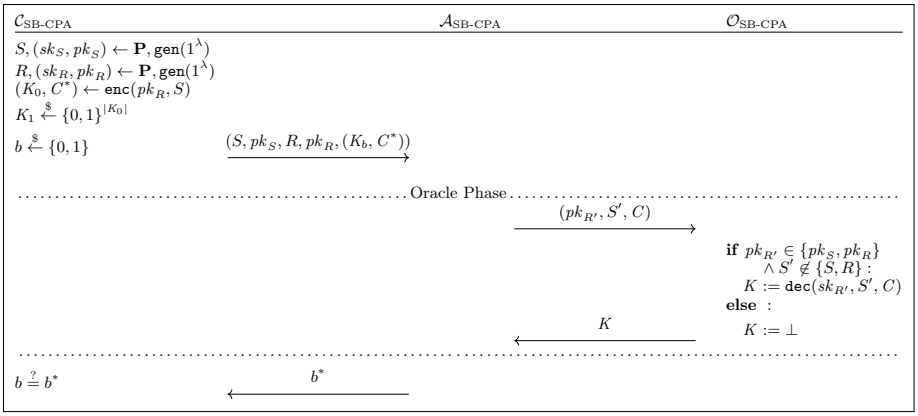

Fig. 4. The IND-SB-CPA$_{\text{SB-KEM}}$ Game for SB-CPA$_{\text{SB-KEM}}$

We would like to remark several things about this definition.
Firstly, IND-SB-CPA$_{\text{KEM}}$ looks very different from other KEM notions at first glance because it has only one oracle phase instead of two. This is not due to less oracle access but because this way is simpler but equivalent: For IND-SB-CPA, the first and second oracle phase permit exactly the same oracle queries (in constrast to CCA2 for instance). Furthermore in the KEM setting the adversary does not generate any outputs between oracle phases I and II. Hence with IND-SB-CPA$_{\text{KEM}}$ the adversary can save all oracle queries it would make in the first oracle phase and ask them in the second oracle phase instead. We therefore decided to simplify the definition by only including the second oracle phase.

Secondly, note that although the IND-SB-CPA$_{\text{KEM}}$ security notion contains a key pair (sk_S, pk_S) for party S, no such keys need to exist in any protocol. Especially in the session communication setting—but also if communication is one-directional in the single-message setting—only one party needs to have a key pair for the SB-KEM to set up a symmetrically encrypted session. The reason behind the existence of these keys in our security notion is that it makes the notion strictly weaker than if (sk_S, pk_S) were not picked by the challenger. Intuitively,

an IND-SB-CPA$_{\text{KEM}}$ secure KEM does not need to guarantee anything if S's keys may be adversarially chosen rather than honestly (and secretly) generated. This can clearly be seen when considering the generic DRE construction of an SB-KEM in Appendix A: For this construction each encapsulated key is decryptable by both the receiver *and* sender. Hence the adversary choosing or knowing sk_S would completely break the encapsulation.

Before we discuss in which ways IND-SB-CPA$_{\text{SB-KEM}}$ fits into the landscape of other security notions for KEMs, notice that IND-SB-CPA$_{\text{SBE}}$ obviously implies IND-SB-CPA$_{\text{SB-KEM}}$ by the standard PKE to KEM construction of randomly drawing and then encrypting a symmetric key. For KEM security notions, classifying IND-SB-CPA$_{\text{SB-KEM}}$ with respect to classic KEM security is unfortunately rather infeasible. While a classic KEM takes no input and requires secrecy and various forms of integrity about the internally determined key, IND-SB-CPA$_{\text{SB-KEM}}$ asserts only secrecy (no integrity) of the key but additionally provides integrity (without secrecy) of some user input–the identity S. Since those two settings are even more incompatible than comparing SBE to classic PKE notions, we will only consider IND-SB-CPA$_{\text{SB-KEM}}$ in relation to the similar setting of tag-KEMs.

Relation to tag-KEM Security Notions. Let $(\mathtt{gen}, \mathtt{key}, \mathtt{enc}, \mathtt{dec})$ be a tag-KEM. We construct an SB-KEM $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ in the natural way by using sender IDs as tags, and combining `key` and `enc` into a single encryption algorithm. I.e. $\mathtt{Gen} \equiv \mathtt{gen}$, $\mathtt{Enc}(pk_R, S) = (K, C)$ where $(aux, K) \leftarrow \mathtt{key}(pk_R)$ and $C \leftarrow \mathtt{enc}(aux, S)$, and $\mathtt{Dec} \equiv \mathtt{dec}$.

Lemma 1: *If* $(\mathtt{gen}, \mathtt{key}, \mathtt{enc}, \mathtt{dec})$ *is an IND-CCA2 secure tag-KEM then* $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ *is an IND-SB-CPA secure SB-KEM.*

Proof. Assume on the contrary that $\mathcal{A}_{\text{SB}}$ is an adversary with non-negligible success probability in winning the IND-SB-CPA$_{\text{SB-KEM}}$ game. We use $\mathcal{A}_{\text{SB}}$ to construct an equally successful adversary $\mathcal{A}_{\text{tag-KEM}}$. This adversary mainly forwards messages between $\mathcal{A}_{\text{SB}}$ and the challenger and oracle. Additionally it creates credentials for S and uses them to decrypt respective oracle queries. The detailed reduction can be found in Figure 5. □

It is easy to see from the reduction that the CCA2 game grants a lot more oracle access than we need which indicates that IND-SB-CPA$_{\text{SB-KEM}}$ is a lot weaker than IND-CCA2$_{\text{tag-KEM}}$. To further substantiate this claim we take the weakest security notion proposed for TBE, adapt it to the KEM setting and show that it still implies IND-SB-CPA$_{\text{SB-KEM}}$ via the above construction. Note that as far as we know, no weaker security notions than IND-CCA2$_{\text{tag-KEM}}$ have been proposed for tag-KEM so far, which is why we take the detour over a TBE notion. The TBE notion in question is IND-gtag-CCA$_{\text{TBE}}$ which was recapitulated in Section 2.2. The difference to IND-CCA2$_{\text{tag-KEM}}$ lies in the oracle access as well as when and by whom the challenge tag τ^* is chosen: Both oracle phases grant

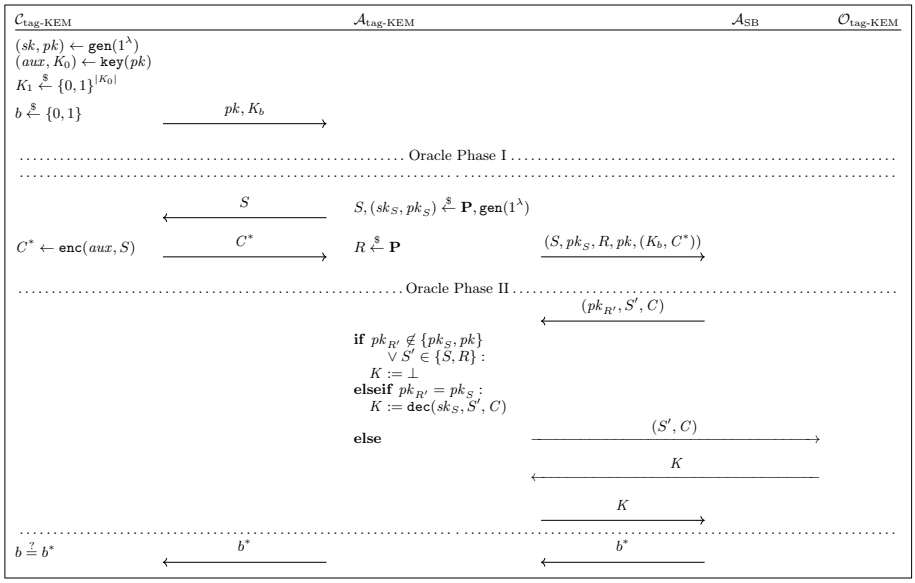

Fig. 5. Reduction for IND-CCA2$_{\text{tag-KEM}}$ Construction

access to a decryption oracle punctuated at $\tau = \tau^*$, i.e. the complete challenge tag is excluded from decryption rather than just the challenge tuple (τ^*, C^*). The challenge tag itself is not chosen adaptively and not even by the adversary at all anymore, but randomly drawn by the challenger. The adaptive interface of a tag-KEM—where encapsulation is divided into `key` and `enc` so that the tag may depend on the output of `key`—does not seem quite fitting anymore when in the security game the tag is drawn at random and hence independent of the output of `key`. Nevertheless we cannot rule out that such a security notion may still be meaningful for a tag-KEM with separate `key` and `enc` and therefore keep the division.

Definition (IND-gtag-CCA$_{\text{tag-KEM}}$): A tag-KEM $\Sigma = (\mathtt{gen}, \mathtt{key}, \mathtt{enc}, \mathtt{dec})$ satisfies *IND-gtag-CCA*, iff for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ the advantage

$$\mathsf{Adv}^{\text{gtag}}_{\mathcal{A},\Sigma}(\lambda) := \Big|\mathbb{P}\Big[b \leftarrow \mathcal{A}_2^{\mathcal{O}^*}(K_b, C^*) \,\Big|\, \tau^* \xleftarrow{\$} \mathbf{T}; (sk, pk) \leftarrow \mathtt{gen}(1^\lambda); \\ (aux_{\mathcal{A}}) \leftarrow \mathcal{A}_1^{\mathcal{O}^*}(pk, \tau^*); (aux, K_0) \leftarrow \mathtt{key}(pk); \\ C^* \leftarrow \mathtt{enc}(aux, \tau^*); K_1 \xleftarrow{\$} \{0,1\}^{|K_0|}; b \xleftarrow{\$} \{0,1\}\Big] - \tfrac{1}{2}\Big|$$

is negligible in λ, where $\mathcal{O}^*$ returns $\bot$ for $\tau = \tau^*$ and $\mathtt{dec}(sk, \tau, C)$ otherwise.

We go on to show that this weaker notion is still sufficient to imply IND-SB-CPA$_{\text{SB-KEM}}$.

Lemma 2: *If* $(\mathtt{gen}, \mathtt{key}, \mathtt{enc}, \mathtt{dec})$ *is an IND-gtag-CCA secure tag-KEM then* $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ *is an IND-SB-CPA secure SB-KEM.*

Proof. The proof of Lemma 2 works almost exactly the same as the proof of Lemma 1. The sole difference is that the identity S is randomly provided by the challenger $\mathcal{C}_{\text{tag-KEM}}$ rather than randomly drawn by $\mathcal{A}_{\text{tag-KEM}}$. Note that the provision $S' \notin \{S, R\}$ guarantees that oracle queries forwarded to $\mathcal{O}_{\text{tag-KEM}}$ get decrypted correctly. □

In Appendix A we furthermore show that the transformation from Lemma 2 is just an implication and no equivalence, proving IND-SB-CPA$_{\text{SB-KEM}}$ to be strictly weaker than IND-gtag-CCA$_{\text{tag-KEM}}$.

5 Realizing Secure Message Transfer

In this section we show that IND-SB-CPA$_{\text{SB-KEM}}$ is—in conjunction with IND-OT secure DEM and authenticated channels—strong enough to facilitate the realization of secure message transfer. Since Beskorovajnov et al. [4] already showed the same for IND-SB-CPA secure SBE with authenticated channels, we can build on their work and only fill in the gap: We show that IND-SB-CPA$_{\text{SB-KEM}}$ combined with IND-OT$_{\text{DEM}}$ via the KEM-DEM-framework yields an IND-SB-CPA secure SBE scheme.

Hence let $(\texttt{gen}, \texttt{enc}, \texttt{dec})$ be an IND-SB-CPA$_{\text{SB-KEM}}$ secure SB-KEM and $(\texttt{DEM.enc}, \texttt{DEM.dec})$ be a compatible IND-OT secure DEM. We construct an SBE scheme via the KEM-DEM principle by setting $\texttt{Gen} \equiv \texttt{gen}$ and:

$\texttt{Enc}(pk_R, S, m)$:
- $(K, C) \leftarrow \texttt{enc}(pk_R, S)$.
- $c \leftarrow \texttt{DEM.enc}(K, m)$.

↪ Return (C, c).

$\texttt{Dec}(sk_R, S, (C, c))$:
- $K := \texttt{dec}(sk_R, S, C)$.
- $m := \texttt{DEM.dec}(K, c)$.

↪ Return m.

Theorem 1: *The SBE scheme* $(\texttt{Gen}, \texttt{Enc}, \texttt{Dec})$ *is IND-SB-CPA secure.*

Proof. Assume there is an adversary $\mathcal{A}_{\text{SBE}}$ for the IND-SB-CPA game with success probability $\mathbb{P}\big[\mathcal{A}_{\text{SBE}} \text{ successful}\big] = \frac{1}{2} + \rho$, where ρ is non-negligible in λ. We use this to construct an adversary $\mathcal{A}_{\text{SB-KEM}}$ for the IND-SB-CPA$_{\text{SB-KEM}}$ game as follows: $\mathcal{A}_{\text{SB-KEM}}$ is started with input $(S, pk_S, R, pk_R, (K_b, C^*))$ by the KEM challenger $\mathcal{C}_{\text{SB-KEM}}$ and hands (S, pk_S, R, pk_R) on to $\mathcal{A}_{\text{SBE}}$. For any valid oracle query $(pk_{R'}, S', (C, c))$ from $\mathcal{A}_{\text{SBE}}$ the DEM key is decrypted via the SB-KEM oracle $\mathcal{O}_{\text{SB-KEM}}$ and subsequently used for DEM decryption of c. When $\mathcal{A}_{\text{SB-KEM}}$ receives challenge messages m_0, m_1 the adversary $\mathcal{A}_{\text{SB-KEM}}$ draws a random challenge bit $b' \xleftarrow{\$} \{0, 1\}$ and determines the challenge as $c^* \leftarrow \texttt{DEM.enc}(K_b, m_{b'})$. The following second oracle phase is conducted exactly as the first one was. Finally, in case $\mathcal{A}_{\text{SBE}}$ correctly answers with b', $\mathcal{A}_{\text{SB-KEM}}$ chooses to answer the challenger with $b^* = 0$, else it answers with $b^* = 1$. The detailed reduction is shown in Figure 6.

Let us briefly analyse the success probability of $\mathcal{A}_{\text{SB-KEM}}$. If $b = 0$, $\mathcal{A}_{\text{SB-KEM}}$ has the same success probability that $\mathcal{A}_{\text{SBE}}$ has. If $b = 1$ we claim that the success probability can only negligibly differ from $\frac{1}{2}$. We show this again by

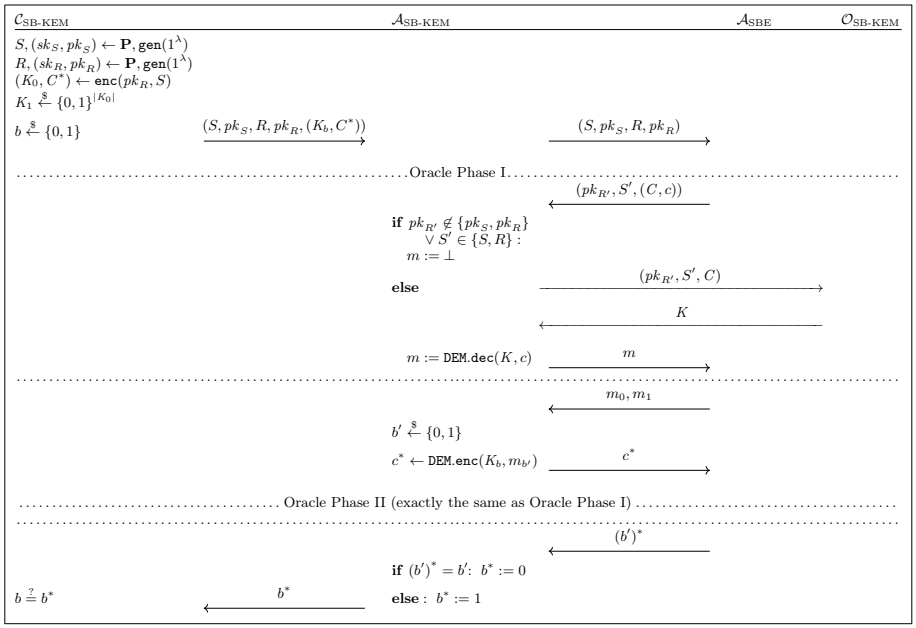

Fig. 6. Reduction from SBE to SB-KEM.

contradiction with a reduction to the IND-OT secure DEM scheme: Assume that when the game is conducted with $b = 1$, $\mathcal{A}_{\text{SBE}}$ has a success probability non-negligibly different from guessing—w.l.o.g. better (rather than worse) than one half. We use $\mathcal{A}_{\text{SBE}}$ to construct an adversary $\mathcal{A}_{\text{DEM}}$ against the DEM IND-OT game: $\mathcal{A}_{\text{DEM}}$ does not get any input from the challenger. It firstly draws S and R, generates $(sk_S, pk_S), (sk_R, pk_R)$, and hands (S, pk_S, R, pk_R) to $\mathcal{A}_{\text{SBE}}$. Every valid oracle query $(pk_{R'}, S', (C, c))$ is answered by using the corresponding secret key with $m := \mathtt{Dec}(sk_{R'}, S', (C, c))$. When $\mathcal{A}_{\text{SBE}}$ chooses challenge messages m_0, m_1 they are handed through to the DEM challenger $\mathcal{C}_{\text{DEM}}$ who responds with a corresponding challenge c^*. This challenge is paired with an output C^* from $\mathtt{enc}(pk_R, S)$ and handed to $\mathcal{A}_{\text{SBE}}$. The second oracle phase, again, is handled exactly as the first one was. Finally the answer b^* from $\mathcal{A}_{\text{SBE}}$ is passed on to the challenger. The detailed reduction is shown in Figure 7.

This reduction to the underlying IND-OT secure DEM shows that for $b = 1$ in the first reduction, the adversary $\mathcal{A}_{\text{SBE}}$ cannot perform non-negligibly better or worse than guessing. Hence, paired with the case $b = 0$, the adversary $\mathcal{A}_{\text{SB-KEM}}$ has success probability $\mathbb{P}\big[\mathcal{A}_{\text{SB-KEM}} \text{ successful}\big] = \frac{1}{2} + \frac{1}{2}\rho$. □

Corollary 1: *Combining the KEM-DEM framework from [2] with the encrypt-then-authenticate protocol from [4], an IND-SB-CPA$_{SB\text{-}KEM}$ secure KEM and IND-OT secure DEM suffice to UC-realize secure message transfer functionality $\mathcal{F}_{MSMT}$ in the $\mathcal{F}_{AUTH}$-hybrid model.*

The proof of this corollary follows directly from Theorem 1 and [4, Thm. 3].

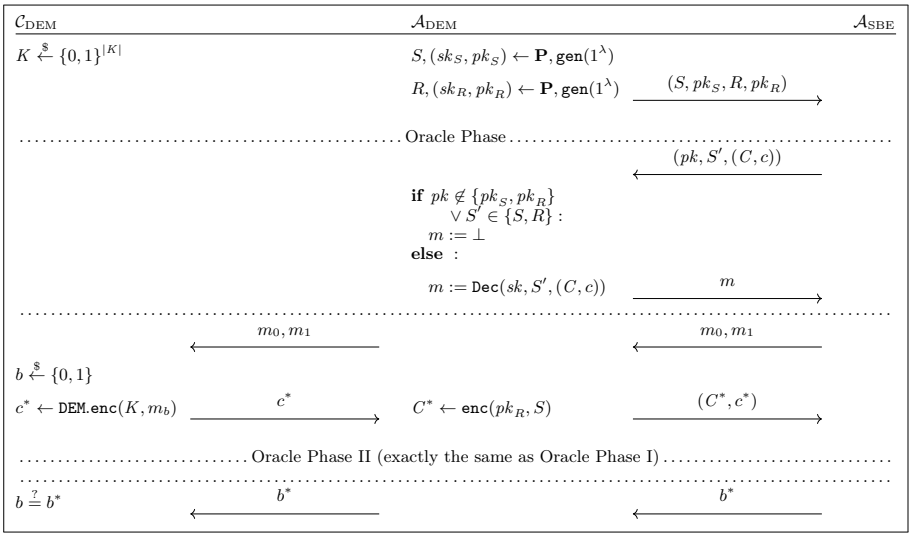

Fig. 7. Reduction from SBE to DEM.

6 Realizing Secure Channels

At this point we make the switch from single-message to session communication. This means symmetric keys are exchanged via the KEM and subsequently used by both parties to send messages encrypted with the corresponding DEM. The benefits are that only one communication partner needs credentials for the KEM and that secure communication can be achieved even if the authenticated channel is only used for the key exchange and not the actual messages. The employed DEM, on the other hand, needs to be stronger than for single-message KEM-DEM.[6] In this section we show how IND-SB-CPA$_{\text{KEM}}$ in conjuction with IND-CCA2$_{\text{DEM}}$ or just IND-RCCA$_{\text{DEM}}$ suffices to UC-realize secure channels $\mathcal{F}_{\text{MSC}}$ in the $\mathcal{F}_{\text{AUTH}}$-hybrid model. We do so by first providing a protocol π_{MSC} and corresponding simulator $\mathcal{S}_{\text{MSC}}$ before giving the actual theorem and proof.

Protocol π_{MSC}. Let $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ be an SB-KEM and $(\mathtt{DEM.enc}, \mathtt{DEM.dec})$ a compatible DEM. The idea behind π_{MSC} is the following: To establish a session between parties P and P', a new symmetric key is generated and encapsulated via $\mathtt{enc}(pk_{P'}, P)$ by P. The resulting ciphertext C is sent to P' via authenticated channel. When decryption $\mathtt{dec}(sk_{P'}, P, C)$ is successful, both parties can encrypt messages to the other party via $\mathtt{DEM.enc}$ and send them on a plain channel. All details can be found in the formal definition:

[6] Note that the security of the DEM can be significantly extenuated if we are willing to use authenticated channels for all messages.

$\pi_{\mathbf{MSC}}$

Realizes:
Multiple secure two-party communication sessions.

Parameters:
- Functionality $\mathcal{F}_{\text{AUTH}}$.
- KEM $(\texttt{gen}, \texttt{enc}, \texttt{dec})$.
- DEM $(\texttt{DEM.enc}, \texttt{DEM.dec})$.

State of party P:
- A personal KEM key function $f_{\text{KEM}}\colon sid \mapsto (pk, sk)$.
- A partial KEM key function $f_{\text{PK}}\colon (sid, P') \mapsto pk_{P'}$.
- A partial DEM session key function $f_{\text{SK}}\colon (sid, P') \mapsto K$.
- An (almost) boolean function $f_{\text{act}}\colon \mathbf{SID} \times \mathbf{P} \to \{\mathsf{true}, \mathsf{false}, \texttt{init}\}$ initialized to $f_{\text{act}} \equiv \mathsf{false}$.

Behaviour of Party P:

\\ Initialization
- Upon input $(\texttt{sent}, sid_{\text{AUTH}}, P', P, (sid, pk))$ from $\mathcal{F}_{\text{AUTH}}$, append $(sid, P') \mapsto pk_{P'}$ to f_{PK} if this entry does not yet exist.
- Upon input $(\texttt{init}, sid, P')$ from the environment:
  (1) If no entry $f_{\text{KEM}}(sid)$ exists set $f_{\text{KEM}}(sid) := (pk, sk) \leftarrow \texttt{gen}(1^\lambda)$.
  (2) Check that $f_{\text{act}}(sid, P') = \mathsf{false}$ and set $f_{\text{act}}(sid, P') := \texttt{init}$.
  (3) Draw fresh sid_{AUTH} and call $\mathcal{F}_{\text{AUTH}}$ with input $(\texttt{send}, sid_{\text{AUTH}}, P', (sid, pk))$.
- Upon input $(\texttt{establish}, sid, P')$ from the environment:
  (1) Look up $pk_{P'} := f_{\text{PK}}(sid, P')$.
  (2) $(K, C) \leftarrow \texttt{enc}(pk_{P'}, P)$.
  (3) Check that $f_{\text{act}}(sid, P') = \mathsf{false}$, set $f_{\text{act}}(sid, P') = \mathsf{true}$ and append $(sid, P') \mapsto K$ to f_{SK}.
  (4) Draw fresh sid_{AUTH} and call $\mathcal{F}_{\text{AUTH}}$ with input $(\texttt{send}, sid_{\text{AUTH}}, P', (sid, C))$.
- Upon input $(\texttt{sent}, sid_{\text{AUTH}}, P', P, (sid, C))$ from $\mathcal{F}_{\text{AUTH}}$:
  (1) Look up $(pk, sk) := f_{\text{KEM}}(sid)$.
  (2) $K := \texttt{dec}(sk, P', C)$.
  (3) Check that $f_{\text{act}}(sid, P') = \texttt{init}$, set $f_{\text{act}}(sid, P') = \mathsf{true}$ and append $(sid, P') \mapsto K$ to f_{SK}.

\\ Data Exchange
- Upon input $(\texttt{send}, sid, P', m)$ with $m \in \{0,1\}^l$ from environment $\mathcal{Z}$:
  (1) Check $f_{\text{act}}(sid, P') = \mathsf{true}$, look up $K := f_{\text{SK}}(sid, P')$ and set $c \leftarrow \texttt{DEM.enc}(K, m)$.
  (2) Send (sid, P, c) to P'
- Upon receiving message (sid, P', c):
  (1) Check $f_{\text{act}}(sid, P') = \mathsf{true}$, look up $K := f_{\text{SK}}(sid, P')$ and set $m \leftarrow \texttt{DEM.dec}(K, c)$.
  (2) Output $(\texttt{sent}, sid, P', m)$ to the environment.

\\ Session Expiration
- Upon input $(\texttt{expire}, sid, P')$ from the environment:
  (1) Check $f_{\text{act}} = \mathsf{true}$ and send $(\texttt{expire}, sid, P)$ to P'.
  (2) Erase $f_{\text{SK}}(sid, P')$ and set $f_{\text{act}}(sid, P') := \mathsf{false}$.
- Upon receiving message $(\texttt{expire}, sid, P')$ erase $f_{\text{SK}}(sid, P')$ and set $f_{\text{act}}(sid, P') := \mathsf{false}$.

Simulator $\mathcal{S}_{\mathrm{MSC}}$. To show that protocol π_{MSC} realizes $\mathcal{F}_{\mathrm{MSC}}$ we need to construct a simulator which interacts with $\mathcal{F}_{\mathrm{MSC}}$ in such a way that no environment $\mathcal{Z}$ can distinguish this ideal world from an interaction with the real protocol and (dummy) adversary $\mathcal{A}$. The idea behind our simulator $\mathcal{S}_{\mathrm{MSC}}$ is striving for near perfect simulation: It plays all honest parties (conducting protocol π_{MSC}) as well as the functionality $\mathcal{F}_{\mathrm{AUTH}}$ in its head, using $\mathcal{F}_{\mathrm{MSC}}$'s outputs to give them mock inputs from $\mathcal{Z}$ and using their outputs in turn to determine inputs to $\mathcal{F}_{\mathrm{MSC}}$. An overview can be found in Figure 8. For proof simplicity purposes—that become

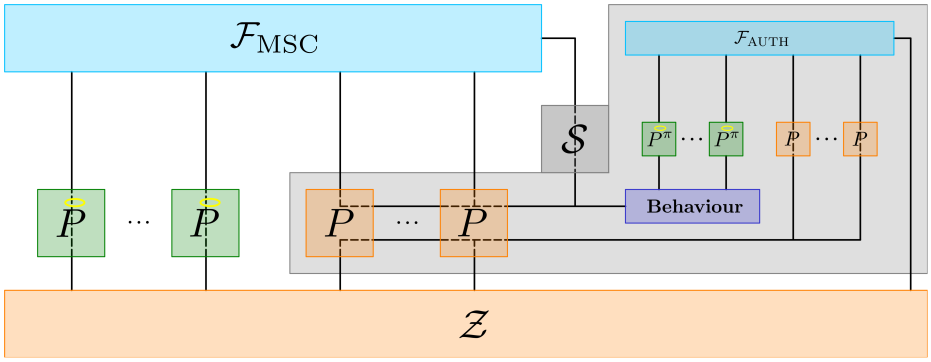

Fig. 8. Overview of Simulator $\mathcal{S}_{\mathrm{MSC}}$ adapted from [4].

apparent later on—the simulator swaps symmetric keys for random values if the two involved parties are both honest. The only situations in which $\mathcal{S}_{\mathrm{MSC}}$ is unable to provide perfect simulation due to lack of knowledge are actual messages between two honest parties. In this case it sends encryptions of zeros instead. The formal definition of $\mathcal{S}_{\mathrm{MSC}}$ looks as follows:

$\mathcal{S}_{\mathrm{MSC}}$

Realizes:
Multiple secure two-party communication sessions.

Parameters:

- Security parameter λ.
- KEM ($\mathtt{gen}, \mathtt{enc}, \mathtt{dec}$).
- DEM ($\mathtt{DEM.enc}, \mathtt{DEM.dec}$).

In-the-head Parties:

- Functionality $\mathcal{F}_{\mathrm{AUTH}}$. This functionality communicates in-the-head with all honest in-the-head parties as well as with the environment $\mathcal{Z}$ as adversary.
- Copies of honest parties running a modified version of the protocol π_{MSC}, which we will denote as P^{π}. These parties communicate in-the-head with the

in-the-head functionality $\mathcal{F}_{\text{AUTH}}$. Their interface to the environment is played by the simulator (defined in "Behaviour" below). The modification from π_{MSC} looks as follows:

- Upon input $(\texttt{establish}, sid, P')$ from the environment:
  (3) Check that $f_{\text{act}}(sid, P') = \mathsf{false}$, set $f_{\text{act}}(sid, P') = \mathsf{true}$, ask $\mathcal{S}$ for freshly drawn random key $K_1 \xleftarrow{\$} \{0,1\}^{|K|}$ for parties $\{P, P'\}$ and append $(sid, P') \mapsto K_1$ to f_{SK}.
- Upon input $(\texttt{sent}, sid_{\text{AUTH}}, P', P, (sid, C))$ from $\mathcal{F}_{\text{AUTH}}$:
  (3) Check that $f_{\text{act}}(sid, P') = \texttt{init}$, set $f_{\text{act}}(sid, P') = \mathsf{true}$, ask $\mathcal{S}$ for key K_1 corresponding to parties $\{P, P'\}$ and append $(sid, P') \mapsto K_1$ to f_{SK}.

- Dummy corrupted parties. Whenever the simulator is asked by the environment to call functionality $\mathcal{F}_{\text{AUTH}}$ in the name of a corrupted party, this in-the-head dummy calls the in-the-head functionality correspondingly and reports all outputs back to the environment $\mathcal{Z}$.

State:

- Everything the in-the-head parties and functionalities store in their states.
- Partial key function $\{\{P, P'\} \mid P, P' \text{ honest}\} \to \{0,1\}^{n(\lambda)}, \{P, P'\} \mapsto K_1$.

Behaviour:

\\ Initialization by honest party

- Upon receiving $(\texttt{inited}, sid, A, B)$ from $\mathcal{F}_{\text{MSC}}$ for honest party A, start in-the-head party A^π with input $(\texttt{init}, sid, B)$ from the environment $\mathcal{Z}$.
- Upon receiving $(\texttt{established}, sid, A, B)$ from $\mathcal{F}_{\text{MSC}}$ for honest party B, start in-the-head party B^π with input $(\texttt{establish}, sid, A)$ from the environment $\mathcal{Z}$.

\\ Initialization by corrupted party

- Upon in-the-head party B^π receiving output $(\texttt{sent}, sid_{\text{AUTH}}, A, B, (sid, pk))$ from $\mathcal{F}_{\text{AUTH}}$ for corrupted A, call $\mathcal{F}_{\text{SC}}$ with input $(\texttt{init}, sid, B)$ in the name of A.
- Upon in-the-head party A^π setting $f_{\text{act}}(sid, B)$ from $\texttt{init}$ to true, call $\mathcal{F}_{\text{MSC}}$ with input $(\texttt{establish}, sid, A)$ in the name of B.

\\ Message from honest to honest party

- Upon receiving $(\texttt{send}, sid, mid, S, R)$ from $\mathcal{F}_{\text{MSC}}$ to $\mathcal{A}$ for honest parties S and R:
  (1) Start in-the-head party S^π with input $(\texttt{send}, sid, R, 0)$ from the environment $\mathcal{Z}$.
  (2) If in-the-head party R^π at some point reports output $(\texttt{sent}, sid, S, 0)$, call $\mathcal{F}_{\text{MSC}}$ with input $(\texttt{send ok}, sid, mid)$.[7]

\\ Message from honest to corrupted party

- Upon receiving $(\texttt{send}, sid, mid, S, R)$ from $\mathcal{F}_{\text{MSC}}$ to $\mathcal{A}$ for honest party S and corrupted party R:
  (1) Call $\mathcal{F}_{\text{MSC}}$ with input $(\texttt{send ok}, sid, mid)$.
  (2) Receive output $(\texttt{sent}, sid, S, m)$ from $\mathcal{F}_{\text{MSC}}$ to R.

(3) Start in-the-head party S^{π} with input $(\texttt{send}, sid, R, m)$ from the environment $\mathcal{Z}$.

\\ Message from corrupted to honest party

- Upon in-the-head honest party R^{π} reporting output $(\texttt{sent}, sid, S, m)$:
  (1) Call $\mathcal{F}_{\text{MSC}}$ with input $(\texttt{send}, sid, R, m)$ in the name of S.
  (2) Receive output $(\texttt{send}, sid, mid, S, R)$ from $\mathcal{F}_{\text{MSC}}$ to $\mathcal{A}$.
  (3) Call $\mathcal{F}_{\text{MSC}}$ with input $(\texttt{send ok}, sid, mid)$.

Security Theorem and Proof. Now that we have constructed both protocol and simulator it remains to show that together they make the real and ideal world indistinguishable for any environment. We do so by first explicitly stating the differences between the simulators efforts and perfect simulation. Then we go on to define several hybrid experiments which help us conduct the proof of our security theorem.

Remark 1: It is easy to see that the simulator $\mathcal{S}_{\text{MSC}}$ provides nearly perfect simulation. The two notable exceptions are:

(1) Symmetric keys of sessions between two honest parties: The modification of protocol π_{MSC} for the in-the-head honest parties P^{π} changes the session keys for each session between two honest parties. While a session key K is generated and the corresponding ciphertext C is sent via $\mathcal{F}_{\text{AUTH}}$—just like in the real protocol—all messages of the session are encrypted with a randomly drawn and unrelated key $K_1 \xleftarrow{\$} \{0,1\}^{|K|}$.
(2) Message content between two honest parties: Let S, R and m be the honest partys and message in question. In this case a message $(sid, S, \texttt{DEM.enc}(K_1, 0))$ will be sent from S to R in the ideal experiment while the protocol execution contains message $(sid, S, \texttt{DEM.enc}(K, m))$ instead.

Hence any environment $\mathcal{Z}$ which distinguishes experiments $\text{EXEC}^{\pi_{\text{MSC}}}_{\mathcal{D},\mathcal{Z}}$ and $\text{IDEAL}^{\mathcal{F}_{\text{MSC}}}_{\mathcal{S}_{\text{MSC}},\mathcal{Z}}$ can only do so by session keys or messages between honest parties.

Before we proceed to our security theorem and proof we need several hybrid experiments and also prove an auxiliary lemma which lets us deal with infinite chains of hybrids.

Definition (Hybrids $H^{\neg}$,$H^{\vdash}_{k}$,$H^{\dashv}_{k,m}$):

- We use a "middle" hybrid $H^{\neg}$ where all honest parties swap encapsulated session keys K for randomly drawn K_1's, while still using ciphertexts C corresponding to K. I.e. parties conduct the same modified protocol as the

[7] We assume the simulator to internally track the protocol executions to know which *mid* to use.

simulator's in-the-head honest parties P^{π} which means that session keys of two honest parties are handled exactly as in the ideal experiment. Note that in contrast to the ideal experiment for every message m between two honest parties in $H^{\dashv}$ there is a message $(sid, S, \texttt{DEM.enc}(K_1, m))$ which contains an encryption of m and *not* an encryption of 0.

- Let $k \in \mathbb{N}_0$ be a natural number or zero. We define $H_k^{\vdash}$ to be almost identical to the real-world execution of π_{MSC} with the sole difference that for the first k sessions between two honest parties, the encapsulated key K is swapped for a randomly drawn K_1. Hence we have $H_0^{\vdash} = \mathrm{EXEC}_{\mathcal{D},\mathcal{Z}}^{\pi_{\mathrm{MSC}}}$ and $\lim_{k\to\infty} H_k^{\vdash} = H^{\dashv}$.
- Let $k \in \mathbb{N}$, $m \in \mathbb{N}_0$ again be natural numbers with m possibly zero. We define $H_{k,m}^{\dashv}$ to be almost identical to $H^{\dashv}$ with the exception that for all messages in the first $k-1$ sessions between two honest parties and the first m messages sent in the k-th session between two honest parties, encryptions of zeros are sent over the channel instead of encryptions containing the real messages. Hence we have $H_{1,0}^{\dashv} = H^{\dashv}$, individual limits $\lim_{m\to\infty} H_{k,m}^{\dashv} = H_{k+1,0}^{\dashv}$ for all $k \in \mathbb{N}$ and overall limit $\lim_{k\to\infty} H_{k,m}^{\dashv} = \mathrm{IDEAL}_{\mathcal{S}_{\mathrm{MSC}},\mathcal{Z}}^{\mathcal{F}_{\mathrm{MSC}}}$.

These hybrid definitions give us the following double-chain of hybrids connecting the real-world execution of π_{MSC} and the ideal experiment with $\mathcal{F}_{\mathrm{MSC}}$:

$$\mathrm{EXEC}_{\mathcal{D},\mathcal{Z}}^{\pi_{\mathrm{MSC}}} = H_0^{\vdash}, H_1^{\vdash}, \ldots \to H^{\dashv} = H_{1,0}^{\dashv}, H_{2,0}^{\dashv}, \ldots \to \mathrm{IDEAL}_{\mathcal{S}_{\mathrm{MSC}},\mathcal{Z}}^{\mathcal{F}_{\mathrm{MSC}}}$$

where each $H_{k,0}^{\dashv}$ is again connected to $H_{k+1,0}^{\dashv}$ by a chain of hybrids $\{H_{k,m}^{\dashv}\}_m$. The following lemma will help us deal with this infinite series of infinite hybrid series:

Lemma 3: *Let $\{H_k\}_{k\in\mathbb{N}_0}$ be series of PPT experiments where executions of H_{k-1} and H_k do not differ before their k-th activation. Let furthermore limit $H_\infty := \lim_{k\to\infty} H_k$ exist and $\mathcal{Z}$ be a PPT environment which distinguishes experiments H_0 and H_∞. Then there is a $\kappa \in \mathbb{N}$ such that a PPT environment $\mathcal{Z}_\kappa$ exists which distinguishes consecutive experiments $H_{\kappa-1}$ and H_κ.*

Proof. Let $p_{\mathcal{Z}}$ be a polynomial which bounds the runtime of the distinguishing PPT environment $\mathcal{Z}$. Since $\mathcal{Z}(\lambda)$ takes at most $p_{\mathcal{Z}(\lambda)}$ steps for the execution of any experiment, all experiments $\{H_k\}_{k>p_{\mathcal{Z}(\lambda)}}$ are necessarily indistinguishable for $\mathcal{Z}$, since they do not differ before their $p_{\mathcal{Z}(\lambda)}$-th activation. Hence $\mathcal{Z}$ is a distinguisher for H_0 and $H_{p_{\mathcal{Z}}}$. We now use the fact that computational indistinguishability is an equivalence relation and in particular transitive. This yields the existence of a $\kappa < p_{\mathcal{Z}}$ and distinguisher $\mathcal{Z}_\kappa$ for experiments $H_{\kappa-1}$ and H_κ. □

Now we are finally ready to formally state and prove that π_{MSC} realizes secure channels:

Theorem 2: *Under static corruption the protocol π_{MSC} with IND-SB-CPA secure SB-KEM and IND-CCA2$_{DEM}$ secure DEM realizes $\mathcal{F}_{MSC}$ in the $\mathcal{F}_{AUTH}$-hybrid model. I.e.*

$$\pi_{MSC}^{\mathcal{F}_{AUTH}} \geq_{UC} \mathcal{F}_{MSC}.$$

Proof. We conduct the proof in two steps, we separately show that (1) $\text{EXEC}^{\pi_{\text{MSC}}}_{\mathcal{D},\mathcal{Z}}$ is indistinguishable from $H^{\dashv}$, and (2) $H^{\dashv}$ is indistinguishable from $\text{IDEAL}^{\mathcal{F}_{\text{MSC}}}_{\mathcal{S}_{\text{MSC}},\mathcal{Z}}$. We reduce the first step to the IND-SB-CPA security of the underlying SB-KEM scheme and the second step to the $\text{IND-CCA2}_{\text{DEM}}$ security of the DEM scheme. For both parts we employ Lemma 3 to go from the corresponding infinite hybrid chain to two consecutive hybrids.

(1) Assume that $\text{EXEC}^{\pi_{\text{MSC}}}_{\mathcal{D},\mathcal{Z}}$ and $H^{\dashv}$ are computationally distinguishable. Then by Lemma 3 there is a $\kappa_1 \in \mathbb{N}$ and environment $\mathcal{Z}_1$ which can distinguish consecutive hybrids $H^{\vdash}_{\kappa_1-1}$ and $H^{\vdash}_{\kappa_1}$, i.e. $H^{\vdash}_{\kappa_1-1} \not\sim_{\mathcal{Z}_1} H^{\vdash}_{\kappa_1}$. We use this to construct a non-negligibly successful adversary $\mathcal{A}_1 = \mathcal{A}_{\text{SB-KEM}}$ in the following way: The adversary $\mathcal{A}_1$ is started by $\mathcal{C}_{\text{SB-KEM}}$ with input $(S, pk_S, R, pk_R, (K_b, C^*))$ and in turn starts $\mathcal{Z}_1$ in its head, playing all other parties just like they would conduct hybrid $H^{\vdash}_{\kappa_1-1}$ or $H^{\vdash}_{\kappa_1}$. If $\mathcal{Z}_1$ corrupts either S or R, the adversary aborts. Since S and R were randomly drawn by the challenger and since by Remark 1 $\mathcal{Z}_1$ needs a message between honest parties to distinguish anything, $\mathcal{A}_1$ has a polynomial chance to not abort at this point.
When $\mathcal{Z}_1$ asks honest party S or R to initialize for the first time, $\mathcal{A}_1$ inserts pk_S/pk_R as S/R's public key respectively for the KEM scheme. Every time in-the-head party S or R send a cipher C encrypted under pk_S/pk_R by some corrupted party P, $\mathcal{A}_1$ decrypts it via the $\text{IND-SB-CPA}_{\text{SB-KEM}}$ oracle. This is possible since S and R are honest and hence $P \notin \{S, R\}$. Since honest parties only get interface inputs from $\mathcal{Z}_1$, $\mathcal{A}_1$ already knows the content of all ciphertexts C sent from honest parties to S and R and does not need the oracle to decrypt them.
If the κ_1-th request of $(\texttt{establish}, sid, P)$ by $\mathcal{Z}$ to establish a session between two honest parties is not made to S with $P = R$, abort. This again gives $\mathcal{A}_1$ a polynomial chance not to abort at this stage. Otherwise insert the challenge cipher C^* into the message $(\texttt{send}, sid_{\text{AUTH}}, P', (sid, C^*))$ from S to R via $\mathcal{F}_{\text{AUTH}}$ and have S and R use challenge key K_b as the DEM key throughout this session. For all following sessions use the encapsulated session keys K just as $H^{\vdash}_{\kappa_1-1}$ and $H^{\vdash}_{\kappa_1}$ both specify. When $\mathcal{Z}_1$ halts, $\mathcal{A}_1$ outputs $b = 0$ if $\mathcal{Z}_1$ outputs $H^{\vdash}_{\kappa_1-1}$, and $b = 1$ if $\mathcal{Z}_1$ outputs $H^{\vdash}_{\kappa_1}$. This way $\mathcal{A}_1$ wins the $\text{IND-SB-CPA}_{\text{SB-KEM}}$ game whenever it did not abort and $\mathcal{Z}_1$ successfully distinguished $H^{\vdash}_{\kappa_1-1}$ and $H^{\vdash}_{\kappa_1}$, i.e. with non-negligible probability. This contradicts the $\text{IND-SB-CPA}_{\text{SB-KEM}}$ security of the underlying KEM scheme and shows that $\text{EXEC}^{\pi_{\text{MSC}}}_{\mathcal{D},\mathcal{Z}}$ must be indistinguishable from $H^{\dashv}$.

(2) Assume that $H^{\dashv}$ and $\text{IDEAL}^{\mathcal{F}_{\text{MSC}}}_{\mathcal{S}_{\text{MSC}},\mathcal{Z}}$ are computationally distinguishable. Then by Lemma 3 there is a $\kappa_2 \in \mathbb{N}$ such that consecutive hybrids $H^{\dashv}_{\kappa_2,0}$ and $H^{\dashv}_{\kappa_2+1,0}$ are computationally distinguishable as well. Again by Lemma 3 there is a $\mu \in \mathbb{N}$ and environment $\mathcal{Z}_2$ which can distinguish consecutive hybrids $H^{\dashv}_{\kappa_2,\mu-1}$ and $H^{\dashv}_{\kappa_2,\mu}$, i.e. $H^{\dashv}_{\kappa_2,\mu-1} \not\sim_{\mathcal{Z}_2} H^{\dashv}_{\kappa_2,\mu}$. We use this to construct a non-negligibly successful adversary $\mathcal{A}_2 = \mathcal{A}_{\text{CCA2-DEM}}$ in the following way: After the challenger $\mathcal{C}_{\text{CCA2-DEM}}$ has randomly drawn the challenge key, the adversary $\mathcal{A}$ is started without input and in turn starts $\mathcal{Z}_2$ in its head,

playing all other parties just like they would conduct hybrid $H^{\dashv}_{\kappa_2,\mu-1}$ or $H^{\dashv}_{\kappa_2,\mu}$. When $\mathcal{Z}_2$ asks for the κ-th session between two honest parties—call them S and R—to be established, $\mathcal{A}$ does not draw a fresh random session key K_1 but rather inserts the (unknown) challenge key instead. This is no problem as all necessary encryptions and decryptions can be obtained via the IND-CCA2$_{\text{DEM}}$ oracle.[8] For the μ-th message m_μ of this session—which by Remark 1 has to be send by an honest party and hence S or R—$\mathcal{A}$ hands m_μ and 0 to the challenger and in return obtains ciphertext c^* which it uses as the channel content reported to $\mathcal{Z}_2$. Now continue to use encryptions of zeros for all further messages of this session, just as $H^{\dashv}_{\kappa_2,\mu-1}$ and $H^{\dashv}_{\kappa_2,\mu}$ require. Whenever the challenge ciphertext c^* is sent to S or R within this session again, act as if the decryption oracle had yielded message m_μ. When $\mathcal{Z}_2$ halts, $\mathcal{A}_2$ outputs $b = 0$ if $\mathcal{Z}_2$ outputs $H^{\dashv}_{\kappa_2,\mu-1}$, and $b = 1$ if $\mathcal{Z}_2$ outputs $H^{\dashv}_{\kappa_2,\mu}$. This way $\mathcal{A}_2$ wins the IND-CCA2$_{\text{DEM}}$ game whenever $\mathcal{Z}_2$ successfully distinguished $H^{\dashv}_{\kappa_2,\mu-1}$ and $H^{\dashv}_{\kappa_2,\mu}$, i.e. with non-negligible probability. This contradicts the IND-CCA2$_{\text{DEM}}$ security of the underlying DEM scheme and shows that H^{-} must be indistinguishable from IDEAL$^{\mathcal{F}_{\text{MSC}}}_{\mathcal{S}_{\text{MSC}},\mathcal{Z}}$.

With these two steps transitivity of computational indistinguishability concludes our proof. □

Just as with many other applications of CCA2 security, the building block can be swapped for one satisfying the strictly weaker RCCA security if the message space is super-polynomial in size.

Theorem 3: *Under static corruption the protocol π_{MSC} with IND-SB-CPA secure SB-KEM and IND-RCCA secure DEM with super-polynomial message size realizes $\mathcal{F}_{MSC}$ in the $\mathcal{F}_{AUTH}$-hybrid model as well.*

Proofsketch. Because the proof largely follows the proof of Theorem 2, we will only sketch the differences. Instead of sending encryptions of 0 for messages between honest parties, the simulator draws a uniformly random value r from the message space $\mathbf{M}$ at the start of the execution and uses this value throughout the protocol. This is vital for when in proof step (2)—after the insertion of c^* as the ciphertext of the μ-th message—other ciphertexts are sent within the same session which the IND-RCCA$_{\text{DEM}}$ oracle refuses to decrypt. Whenever this happens, let $\mathcal{A}_2$ act as if decryption yielded message m_μ. By definition of the oracle the ciphertext may also contain r instead of m_μ which would lead to a simulation error and hence we have no guarantees on the output of $\mathcal{Z}_2$ in this case. But since r was randomly drawn from a super-polynomial message space, the probability that $\mathcal{Z}_2$ tries to send a ciphertext containing it is negligible and the

[8] Note that although $\mathcal{A}$ knows the content of any message that $\mathcal{Z}_2$ asks S or R to send, this communication is not handled via $\mathcal{F}_{\text{AUTH}}$ and hence every corrupted party may send ciphertexts to S or R expecting them to decrypt as if they were from the other party.

error does not impede our construction of a non-negligibly successful adversary $\mathcal{A}_2$. ▽

7 Efficient LWE-based Construction

After the very theoretic definitions and transformation from Sections 4 and 5 we now go on to show the real-world benefit of the new IND-SB-CPA$_{\text{SB-KEM}}$ notion. We do so by giving an LWE based SB-KEM construction in the standard model which is even simpler than the, as far as we know, most efficient standard model construction previously used to construct IND-CCA2$_{\text{PKE}}$ security [6] and show that it still satisfies our IND-SB-CPA$_{\text{SB-KEM}}$ notion. Our construction is a tweaked version of the KEM part from [22, 6], where we use sender IDs instead of a hash and remove the employed MAC entirely.

Building blocks needed for this construction are the trapdoor function and gadget matrix G from [22] as well as the corresponding `invert` function, a full-rank difference encoding function `FRD` from [23] translating sender IDs to suitable matrices, a key derivation function (KDF) `KDF` and gaussian distributions $\mathcal{D}$. Using these building blocks we define an SB-KEM $\Sigma := (\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ as follows:

$\mathtt{gen}(1^\lambda)$:
- $\mathsf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$
- $\mathsf{R} \leftarrow \mathcal{D}^{m \times o}_{\omega(\sqrt{\log(n)})}$
- $\mathsf{A}_1 := \mathsf{A} \cdot \mathsf{R}$

$\hookrightarrow$ Return $(sk, pk) := \big(\mathsf{R}, (\mathsf{A}, \mathsf{A}_1)\big)$.

$\mathtt{enc}(pk, S) = \mathtt{enc}\big((\mathsf{A}, \mathsf{A}_1), S\big)$:
- $\mathsf{e} \leftarrow \mathcal{D}^n_{\alpha \cdot q}$; $\mathsf{e}_0 \leftarrow \mathcal{D}^m_{\alpha \cdot q}$; $\mathsf{e}_1 \leftarrow \mathcal{D}^o_\sigma$, where $\sigma^2 = \big(\|\mathsf{e}_0\|^2 + m(\alpha q)^2\big) \cdot \omega\big(\sqrt{\log(n)}\big)^2$.
- $\mathsf{k} \xleftarrow{\$} \{0,1\}^n$
- $\mathsf{s} = \mathsf{k} \cdot \lfloor \frac{q}{2} \rfloor + \mathsf{e}$
- $\mathsf{c}_0 = \mathsf{s}^\top \mathsf{A} + \mathsf{e}_0$
- $\mathsf{c}_1 = \mathsf{s}^\top (\mathsf{A}_1 + \mathtt{FRD}(S)\mathsf{G}) + \mathsf{e}_1$

$\hookrightarrow$ Return $(K, C) := \big(\mathtt{KDF}(\mathsf{k}), (\mathsf{c}_0, \mathsf{c}_1)\big)$.

$\mathtt{dec}(sk, S, C) = \mathtt{dec}\big(\mathsf{R}, S, (\mathsf{c}_0, \mathsf{c}_1)\big)$:
- $(\mathsf{s}, \mathsf{e}_0, \mathsf{e}_1) \leftarrow \mathtt{invert}\big(\mathsf{R}, [\mathsf{A} | \mathsf{A}_1 + \mathtt{FRD}(S)G], [\mathsf{c}_0^\top, \mathsf{c}_1^\top]\big)$
- Check $\|\mathsf{e}_0\| \leq \alpha q \sqrt{m}$ and $\|\mathsf{e}_1\| \leq \alpha q \sqrt{2mo} \cdot \omega\big(\sqrt{\log(n)}\big)$.[a]
- For $i \in \{0, \ldots, n-1\}$: $\mathsf{k}[i] := \begin{cases} 0, \text{ if } \mathsf{s}[i] \text{ closer to } 0 \\ 1, \text{ if } \mathsf{s}[i] \text{ closer to } \frac{q}{2} \end{cases}$.
- Check $\|\mathsf{s} - \mathsf{k}\| \leq \alpha q \sqrt{n}$.[a]

$\hookrightarrow$ Return $K = \mathtt{KDF}(\mathsf{k})$.

[a] If any check fails, abort with output $\perp$.

The correctness of the scheme directly carries over from the similar scheme in [6] which is why we concentrate on its security properties in this work. The security of Σ is based on the hardness of the normal form LWE (NLWE) problem. NLWE is an equivalent version of the standard LWE problem where the secret vector is drawn from an error distribution as well [6]. From the straightline reduction to LWE follows the post-quantum security of our construction.

Theorem 4: *The SB-KEM $\Sigma = (\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ is IND-SB-CPA secure, given that the LWE assumption holds. In particular, let $\mathcal{A}$ be an IND-SB-CPA$_{SB\text{-}KEM}$ adversary against the SB-KEM. Then there are distinguishers $\mathcal{A}_{\mathrm{LWE}}$ for NLWE and $\mathcal{A}_{\mathrm{KDF}}$ for KDF $\mathtt{KDF}$, such that for all $\lambda \in \mathbb{N}$*

$$\mathsf{Adv}^{\text{SB-CPA}}_{\mathcal{A},\Sigma}(\lambda) \leq \mathsf{Adv}^{\mathrm{LWE}}_{\mathcal{A}_{\mathrm{LWE}}}(\lambda) + \mathsf{Adv}^{\mathrm{KDF}}_{\mathcal{A}_{\mathrm{KDF}}}(\lambda) + \varepsilon,$$

where ε is negligible in λ.

Proof. We roughly follow the proof idea of [6], constructing a series of games which slowly transform the original IND-SB-CPA$_{\text{SB-KEM}}$ game into a one which is obviously unwinnable. At each definition of a new game we show how the adversary's view changes from the last one.

Game 0: This is the IND-SB-CPA$_{\text{SB-KEM}}$ game.

Game 1: At this point $A_1 = AR$ is swapped for $\big(AR - \mathtt{FRD}(S)G\big)$ in the generation of $pk_R = (A, A_1)$. Since the distributions of AR and $\big(AR - \mathtt{FRD}(S)G\big)$ are both statistically close to uniform randomness over $\mathbb{Z}_q^{n\times o}$ they are by transitivity statistically close to each other. Since $\mathtt{FRD}$ is a full rank difference encoding $\mathtt{FRD}(S') - \mathtt{FRD}(S)$ is invertible if and only if $S' \neq S$. I.e. with the new definition of pk_R decryption of ciphertexts is still possible for any sender ID other than S. As oracle queries with $S' = S$ are not permitted for IND-SB-CPA$_{\text{SB-KEM}}$ anyway, this does not change the oracle at all. Hence the adversary's view in Game 1 is statistically close to the view in Game 0.

Game 2: This game is identical to Game 1, other than the definition of the challenge (c_0^*, c_1^*). Instead of using r we draw a new vector $\overline{c} \xleftarrow{\$} \mathbb{Z}_q^m$ uniformly at random and set $c_0^* := \big(\overline{c} + (k^* \cdot \lfloor \frac{q}{2} \rfloor)^\top A\big)$. For the construction of c_1^* a new random error $\overline{e} \leftarrow \mathcal{D}^{\omega}_{\overline{\sigma}}$ with $\overline{\sigma}^2 = m(\alpha q)^2 \cdot \omega\big(\sqrt{\log(n)}\big)^2$ is drawn and c_1^* set to $c_1^* := \big((c_0^*)^\top R + \overline{e}\big)$. We reduce this change to the hardness of NLWE by showing that from an adversary $\mathcal{A}_{1|2}$ distinguishing Game 1 and Game 2 with non-negligible success probability we can construct an adversary $\mathcal{A}_{\mathrm{LWE}}$ with the same success probability in breaking the NLWE assumption: After getting input (B, b) from the challenger $\mathcal{C}_{\mathrm{LWE}}$, $\mathcal{A}_{\mathrm{LWE}}$ follows Game 1 apart from two definitions. In R's public key $pk_R = (A, A_1)$ the first value is taken to be $A := B$ which also results in $A_1 = BR$. The value $\overline{c}$ is not drawn randomly but set to b. The rest–including oracle queries–is handled as in Game 1 (which is the same as in Game 2). When $\mathcal{A}_{1|2}$ outputs bit b, which indicates that $\mathcal{A}_{1|2}$ thinks it interacts with Game $(b+1)$, $\mathcal{A}_{\mathrm{LWE}}$ outputs the same b to $\mathcal{C}_{\mathrm{LWE}}$.

For the analysis of the reduction firstly note that the distribution of the public key A has not changed at all. In case b is of the form $b = x^\top B + y$, we have

$$c_0^* = \big(b + (k^* \cdot \lfloor \frac{q}{2} \rfloor)^\top A\big) = (k^* \cdot \lfloor \frac{q}{2} \rfloor + x)^\top A + y \sim s^\top A + e_0 \tag{1}$$

$$c_1^* = (c_0^*)^\top R + \overline{e} \overset{(1)}{\sim} (s^\top A + e_0)^\top R + \overline{e} \overset{(*)}{\sim} s^\top \big(A_1 + \mathtt{FRD}(S)G\big) + e_1,$$

where the second statistic closeness $(*)$ is gained by adapting Theorem 3.1 of [24] and Corollary 3.10 of [25]. This means the view of $\mathcal{A}_{1|2}$ is statistically close to Game 1 if b is an NLWE sample. If, on the other hand, $b \xleftarrow{\$} \mathbb{Z}_q^m$ is random, $\overline{c}$ and hence (c_0^*, c_1^*) is obviously distributed the same as in Game 2.

Game 3: Instead of the construction via $\overline{c}$ from Game 2, c_0^* is drawn uniformly at random from $\mathbb{Z}_q^m$. This means the challenge ciphertext C^* is now completely independent of the key K_0. As the value $\overline{c}$ acted as a one-time-pad on $\big((k^* \cdot \lfloor \frac{q}{2} \rfloor)^\top A\big)$ to define c_0^* in Game 2, the statistical view of the adversary does not change by this modification.

Game 4: As the last step, the key K_0 is drawn uniformly at random rather than generated via the KDF as $\mathtt{KDF}(k)$. It is obvious that with this change, an adversary distinguishing Game 3 and Game 4 can be used to directly construct a KDF distinguisher with the same success probability.

In Game 4 we see that the adversary is tasked to decide which of two randomly drawn keys K_0 and K_1 it was sent while the rest of its view is completely independent of these keys. This gives the adversary an even one half chance to win Game 4 and overall provides us with the inequality claimed in Theorem 4.□

8 Conclusion

In this paper we have introduced the new notion of a sender-binding key encapsulation mechanism (SB-KEM) with corresponding IND-SB-CPA security, building on the works of Beskorovajnov et al. [4]. Although slightly stronger than plain CPA, IND-SB-CPA security is weaker than all other previously proposed (tag-)KEM notions, giving CPA security only for the encapsulated key and non-malleability for the sender ID. Despite its weakness we showed that the sender-binding property makes up for the lack of key non-malleability: It is still possible to realize secure communication via authenticated channels from an IND-SB-CPA secure SB-KEM. This is true both for single-message and session communication, where the SB-KEM needs to be paired with IND-$\mathrm{OT}_{\mathrm{DEM}}$ and IND-$\mathrm{RCCA}_{\mathrm{DEM}}$ respectively. This means it is now possible to get secure communication from weaker assumptions. We show the real world merit of this advancement by providing a post-quantum secure SB-KEM construction based on the standard assumption of LWE. The efficiency of our construction is directly derived from the previous KEMs construction [6] ours is based on.

An interesting theoretic problem for future work is whether IND-SB-CPA security is in fact the weakest possible KEM notion to allow for UC-secure communication via hybrid encryption and authenticated channels.

Acknowledgements. We thank the PKC 2023 anonymous reviewers for their valuable feedback. The work presented in this paper has been funded by the German Federal Ministry of Education and Research (BMBF) under the project "PQC4MED" (ID 16KIS1044) and by KASTEL Security Research Labs.

References

1. Diffie, W., and Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory **22**(6), 644–654 (1976)
2. Shoup, V.: *A Proposal for an ISO Standard for Public Key Encryption*, Cryptology ePrint Archive, Paper 2001/112 (2001). https://eprint.iacr.org/2001/112. 2001
3. Canetti, R., Krawczyk, H., and Nielsen, J.B.: Relaxing Chosen-Ciphertext Security. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003, pp. 565–582. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
4. Beskorovajnov, W., Gröll, R., Müller-Quade, J., Ottenhues, A., and Schwerdt, R.: A New Security Notion for PKC in the Standard Model: Weaker, Simpler, and Still Realizing Secure Channels. In: Hanaoka, G., Shikata, J., and Watanabe, Y. (eds.) Public-Key Cryptography – PKC 2022, pp. 316–344. Springer International Publishing, Cham (2022)
5. Cramer, R., and Shoup, V.: Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. SIAM Journal on Computing **33** (2002). DOI: 10.1137/S0097539702403773
6. Boyen, X., Izabachène, M., and Li, Q.: Secure Hybrid Encryption in the Standard Model from Hard Learning Problems. In: Cheon, J.H., and Tillich, J.-P. (eds.) Post-Quantum Cryptography, pp. 399–418. Springer International Publishing, Cham (2021)
7. Nagao, W., Manabe, Y., and Okamoto, T.: A Universally Composable Secure Channel Based on the KEM-DEM Framework. In: pp. 28–38 (2006). DOI: 10.1007/978-3-540-30576-7_23
8. Information technology — Security techniques — Encryption algorithms — Part 2: Asymmetric ciphers. Standard, Geneva, CH: International Organization for Standardization (2006)
9. Abe, M., Gennaro, R., Kurosawa, K., and Shoup, V.: Tag-KEM/DEM: A New Framework for Hybrid Encryption and A New Analysis of Kurosawa-Desmedt KEM. In: Cramer, R. (ed.) Advances in Cryptology – EUROCRYPT 2005, pp. 128–146. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
10. MacKenzie, P.D., Reiter, M.K., and Yang, K.: Alternatives to Non-malleability: Definitions, Constructions, and Applications (Extended Abstract). In: Naor, M. (ed.) TCC 2004. LNCS, pp. 171–190. Springer Berlin Heidelberg (2004). DOI: 10.1007/978-3-540-24638-1_10
11. Herranz, J., Hofheinz, D., and Kiltz, E.: Some (in)sufficient conditions for secure hybrid encryption. Inf. Comput. **208**, 1243–1257 (2010). DOI: 10.1016/j.ic.2010.07.002
12. Katz, J., and Yung, M.: Characterization of Security Notions for Probabilistic Private-Key Encryption. Journal of Cryptology **19**, 67–95 (2006). DOI: 10.1007/s00145-005-0310-8
13. Canetti, R.: Security and composition of multiparty cryptographic protocols. Journal of CRYPTOLOGY **13**(1), 143–202 (2000)

14. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: Proceedings 42nd IEEE Symposium on Foundations of Computer Science, pp. 136–145 (2001)
15. Canetti, R.: Universally composable signature, certification, and authentication. In: Proceedings. 17th IEEE Computer Security Foundations Workshop, 2004. Pp. 219–233 (2004)
16. Canetti, R., and Krawczyk, H.: Universally composable notions of key exchange and secure channels. In: International Conference on the Theory and Applications of Cryptographic Techniques, pp. 337–351 (2002)
17. Kurosawa, K., and Desmedt, Y.: A New Paradigm of Hybrid Encryption Scheme. In: Franklin, M. (ed.) Advances in Cryptology – CRYPTO 2004, pp. 426–442. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
18. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., and Stehle, D.: CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In: 2018 IEEE European Symposium on Security and Privacy, pp. 353–367 (2018). DOI: 10.1109/EuroSP.2018.00032
19. Choi, S.G., Herranz, J., Hofheinz, D., Hwang, J.Y., Kiltz, E., Lee, D.H., and Yung, M.: The Kurosawa–Desmedt key encapsulation is not chosen-ciphertext secure. Information processing letters **109**(16), 897–901 (2009)
20. Kurosawa, K., and Trieu Phong, L.: Kurosawa-Desmedt Key Encapsulation Mechanism, Revisited. In: Pointcheval, D., and Vergnaud, D. (eds.) Progress in Cryptology – AFRICACRYPT 2014, pp. 51–68. Springer International Publishing, Cham (2014)
21. Canetti, R., and Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In: International conference on the theory and applications of cryptographic techniques, pp. 453–474 (2001)
22. Micciancio, D., and Peikert, C.: Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In: Pointcheval, D., and Johansson, T. (eds.) Advances in Cryptology – EUROCRYPT 2012, pp. 700–718. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
23. Agrawal, S., Boneh, D., and Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) Advances in Cryptology – EUROCRYPT 2010, pp. 553–572. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
24. Peikert, C.: An Efficient and Parallel Gaussian Sampler for Lattices. In: Rabin, T. (ed.) Advances in Cryptology – CRYPTO 2010, pp. 80–97. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
25. Regev, O.: On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. J. ACM **56**(6) (2009). DOI: 10.1145/1568318.1568324
26. Chow, S.S.M., Franklin, M., and Zhang, H.: Practical Dual-Receiver Encryption. In: Benaloh, J. (ed.) Topics in Cryptology – CT-RSA 2014, pp. 85–105. Springer International Publishing, Cham (2014)
27. Kiltz, E., Masny, D., and Pietrzak, K.: Simple Chosen-Ciphertext Security from Low-Noise LPN. In: Krawczyk, H. (ed.) PKC 2014. LNCS, pp. 1–18. Springer Berlin Heidelberg (2014). DOI: 10.1007/978-3-642-54631-0_1

A SB-KEM from DR-KEM

In this appendix we will comprehensively show how an IND-SB-CPA secure SB-KEM can be constructed via DR-KEM. We start by recapitulating the necessary theoretic basics, then provide a generic transformation and finally develop a new and efficient construction from LPN in the standard model. Most of these constructions follow a similar structure as the corresponding SBE constructions in [4].

A.1 DR-KEM Preliminaries

The basics needed to understand our transformation include the definitions of DR-KEM, IND-CPA$_{\text{DR-KEM}}$ security and soundness as well as the ideal key registration functionality $\mathcal{F}_{\text{KRK}}$.

The following definition of a DR-KEM is based on [26]. Note that [26] present the definition in the CRS-model while we assume group parameters to be fixed out of scope.

Definition (DR-KEM): A dual receiver key encapsulation mechanism (DR-KEM) is given by a set of three PPT algorithms $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ with

$$\begin{aligned} \mathtt{gen}: &\quad 1^\lambda \mapsto (sk, pk) \\ \mathtt{enc}: &\quad (pk_1, pk_2) \mapsto (K, C) \\ \mathtt{dec}: &\quad (sk_i, pk_1, pk_2, C) \mapsto K \end{aligned}$$

such that the following correctness property holds:

$$K = \mathtt{dec}(sk_i, pk_1, pk_2, C)$$

whenever $(sk_1, pk_1), (sk_2, pk_2) \leftarrow \mathtt{gen}(1^\lambda)$, $i \in \{1, 2\}$ and $(K, C) \leftarrow \mathtt{enc}(pk_1, pk_2)$.

The basic CPA security notion corresponding to the DR-KEM setting looks as follows:

Definition (IND-CPA$_{\text{DR-KEM}}$): A DR-KEM $\Sigma = (\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ satisfies *indistinguishability under chosen plaintext attack (IND-CPA)* (cf. [26]), iff for any PPT adversary $\mathcal{A}$ the advantage

$$\mathsf{Adv}^{\text{CPA}}_{\mathcal{A},\Sigma}(\lambda) := \left| \mathbb{P}\Big[b \leftarrow \mathcal{A}(pk_1, pk_2, K_b, C^*) \;\Big|\; (pk_1, sk_1), (pk_2, sk_2) \leftarrow \mathtt{gen}(1^\lambda); \right.$$
$$\left. (K_0, C^*) \leftarrow \mathtt{enc}(pk_1, pk_2); K_1 \xleftarrow{\$} \{0,1\}^{|K_0|}; b \xleftarrow{\$} \{0,1\} \Big] - \frac{1}{2} \right|$$

is negligible in λ.

Another property often required of dual-receiver schemes is soundness, so we will also state its formal definition here.

Definition (Soundness): A DR-KEM $\Sigma = (\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ satisfies *soundness* (cf. [26]), iff for any PPT adversary $\mathcal{A}$ the advantage

$$\mathsf{Adv}^{\mathrm{Sound}}_{\mathcal{A},\Sigma}(\lambda) := \mathbb{P}\Big[\mathtt{dec}(sk_1, pk_1, pk_2, C) \neq \mathtt{dec}(sk_2, pk_1, pk_2, C) \;\Big|\; C \leftarrow \mathcal{A}(sk_1, pk_1, sk_2, pk_2); (sk_1, pk_1), (sk_2, pk_2) \leftarrow \mathtt{gen}(1^\lambda)\Big]$$

is negligible in λ.

Lastly, in our generic transformation we encounter the key registration functionality $\mathcal{F}_{\mathrm{KRK}}$. The following definition is taken from [4].

$$\mathcal{F}^{\mathbf{f_{Key}}}_{\mathbf{KRK}}$$

Provides:
Key registration with knowledge.
Parameters:

- Function $\mathtt{f}_{\mathrm{Key}} : (sk, pk) \mapsto \begin{cases} \mathsf{true}, & \text{well-formed key pair} \\ \mathsf{false}, & \text{otherwise} \end{cases}$

State:

- Function $p_{\mathrm{Reg}} : mid \mapsto (P, sk, pk)$ of pending registrations.
- Function $p_{\mathrm{Ret}} : mid \mapsto (P_i, P_j)$ of pending retrievals.
- Set $\mathbf{R}$ of registered tuples (P, sk, pk).

Behaviour:

- Upon receiving $(\mathtt{register}, sid, sk, pk)$ from a party P, draw fresh mid, send $(\mathtt{register}, sid, mid, P, pk)$ to the adversary $\mathcal{A}$ and append $mid \mapsto (P, sk, pk)$ to p_{Reg}.
- Upon receiving $(\mathtt{register\ ok}, sid, mid)$ from the adversary $\mathcal{A}$, retrieve $(P, sk, pk) := p_{\mathrm{Reg}}(mid)$, check
  - $\mathtt{f}_{\mathrm{Key}}(sk, pk) = \mathsf{true}$
  - $\nexists\ sk', pk' : (P, sk', pk') \in \mathbf{R}$
  - $\nexists\ P', sk' : (P', sk', pk) \in \mathbf{R}$

  and append (P, sk, pk) to $\mathbf{R}$ if all checks were successful.
- Upon receiving $(\mathtt{retrieve}, sid, P_i)$ from a party P_j, draw fresh mid, send $(\mathtt{retrieve}, sid, mid, P_i, P_j)$ to the adversary $\mathcal{A}$ and append $mid \mapsto (P_i, P_j)$ to p_{Ret}.
- Upon receiving $(\mathtt{retrieve\ ok}, sid, mid)$ from the adversary $\mathcal{A}$, look up $(P_i, P_j) := p_{\mathrm{Ret}}(mid)$ and $(P_i, sk_i, pk_i) \in R$. If no such entry exists in $\mathbf{R}$, set $pk_i := \bot$. Send $(\mathtt{retrieved}, sid, pk_i, P_i)$ to P_j.

A.2 Transformation from DR-KEM to SB-KEM

To use a DR-KEM in conjunction with the key registration functionality $\mathcal{F}_{\text{KRK}}$, we assume it to permit an efficiently computable boolean function $\mathtt{f}_{\text{Key}}$. This function indicates whether a key pair (sk, pk) is well formed, i.e. , whether it could have been output by the DR-KEMs key generation algorithm or not:

$$\mathtt{f}_{\text{Key}} : (sk, pk) \mapsto \begin{cases} \mathsf{true}, & (sk, pk) \leftarrow \mathtt{gen}(1^\lambda) \\ \mathsf{false}, & \text{else.} \end{cases}$$

This is mainly for convenience. In [4] it was discussed how the need for a function $\mathtt{f}_{\text{Key}}$ can easily be disposed of by having the registration functionality (partially) generate the keys.

Let $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ be a DR-KEM with key function $\mathtt{f}_{\text{Key}}$. We define a new SB-KEM $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$:

$\mathtt{Gen}(1^\lambda)$ executed by party P:
- $(sk, pk) \leftarrow \mathtt{gen}(1^\lambda)$.
- Register (sk, pk) with $\mathcal{F}_{\text{KRK}}^{\mathtt{f}_{\text{Key}}}$.

$\hookrightarrow$ Return $(SK, PK) := ((sk, pk), P)$.

$\mathtt{Enc}(PK_R, S) = \mathtt{Enc}(R, S)$ executed by party S:
- Retrieve pk_R and pk_S from $\mathcal{F}_{\text{KRK}}^{\mathtt{f}_{\text{Key}}}$.

$\hookrightarrow$ Return $(K, C) \leftarrow \mathtt{enc}(pk_R, pk_S)$.

$\mathtt{Dec}(SK_R, S, C) = \mathtt{Dec}((sk_R, pk_R), S, C)$ executed by party R:
- Retrieve pk_S from $\mathcal{F}_{\text{KRK}}^{\mathtt{f}_{\text{Key}}}$.

$\hookrightarrow$ Return $K := \mathtt{dec}(sk_R, pk_R, pk_S, C)$.

The intuition behind this construction is the same as when an SBE scheme is constructed from DRE: By encapsulating the key such that both sender and receiver may decapsulate it with their respective secret keys, soundness of the DR-KEM guarantees to the receiver that the sender has knowledge of the key regardless of who might have constructed the ciphertext C.

Lemma 4: *If the DR-KEM* $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ *is IND-CPA secure and sound, then in the* $\mathcal{F}_{KRK}^{\mathtt{f}_{Key}}$ *hybrid model* $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ *is an IND-SB-CPA*$_{SB\text{-}KEM}$ *secure SB-KEM scheme.*

Proof. We conduct the proof by contradiction. Let $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ be a sound DR-KEM scheme with key function $\mathtt{f}_{\text{Key}}$ and $\mathcal{A}_{\text{SB}}$ be an adversary which has non-negligible success probability in winning the IND-SB-CPA$_{\text{SB-KEM}}$ game with respect to $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$. We use $\mathcal{A}_{\text{SB}}$ to construct an adversary $\mathcal{A}_{\text{DR}}$ with non-negligible success probability in winning the IND-DR-CPA-KEM game with respect to $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$.

Key point in this proof is that while $\mathcal{A}_{\text{DR}}$ has to provide $\mathcal{A}_{\text{SB}}$ with the correct answers to any oracle queries it makes, it also acts as $\mathcal{F}_{\text{KRK}}^{\mathtt{f}_{\text{Key}}}$ for $\mathcal{A}_{\text{SB}}$ and hence has access to any keys $\mathcal{A}_{\text{SB}}$ registers. Note that we let $\mathcal{A}_{\text{DR}}$ handle all interactions

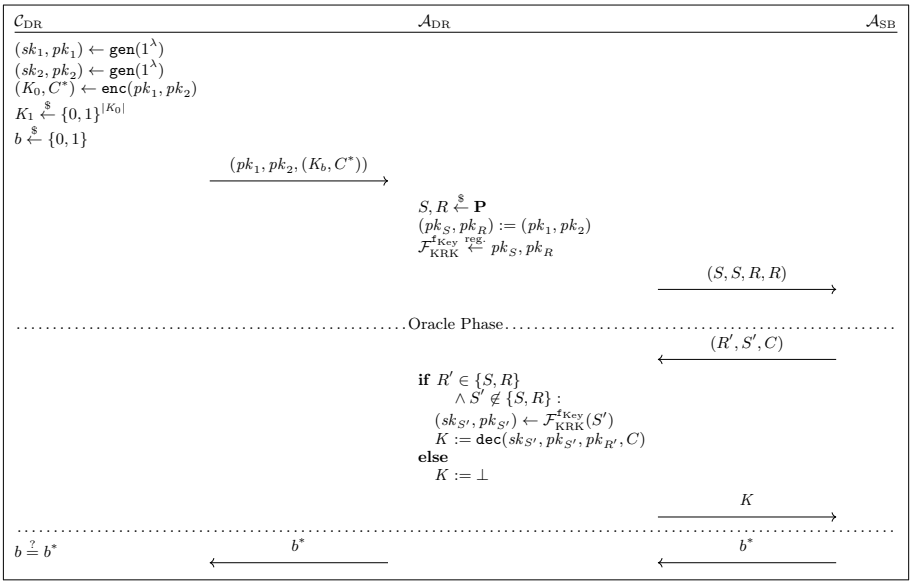

Fig. 9. Reduction for DR-KEM Construction

of $\mathcal{A}_{\text{SB}}$ with $\mathcal{F}_{\text{KRK}}^{\mathtt{f}_{\text{Key}}}$ exactly as the functionality itself would do, with the exception that instantaneous `ok`s are assumed whenever the functionality would ask the adversary for permission. Now for any oracle query (R', S', C), $\mathcal{A}_{\text{DR}}$ looks up the keys $(sk_{S'}, pk_{S'})$ which $\mathcal{A}_{\text{SB}}$ must have registered for the decapsulation not to fail. Due to the soundness property of the DR-KEM those keys can now be used to correctly decapsulate the key

$$K = \mathsf{dec}(sk_{S'}, pk_{S'}, pk_{R'}, C) = \mathsf{dec}(sk_{R'}, pk_{R'}, pk_{S'}, C)$$

and answer the oracle query. This gives $\mathcal{A}_{\text{DR}}$ the same non-negligible success probability as $\mathcal{A}_{\text{SB}}$. The reduction is shown in Figure 9. □

We can now show that IND-SB-CPA$_{\text{SB-KEM}}$ is in fact strictly weaker than IND-gtag-CCA$_{\text{tag-KEM}}$ with party IDs as the tag space by showing that the above construction does not satisfy IND-gtag-CCA security.

Lemma 5: $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ *is not IND-gtag-CCA secure.*

Proof. We prove this by constructing an adversary $\mathcal{A}_{\text{gtag-CCA}}$ which has non-negligible probability of winning the IND-gtag-CCA game. The challenger $\mathcal{C}_{\text{gtag-CCA}}$ provides $\mathcal{A}_{\text{gtag-CCA}}$ with input (pk_R, S) where public key pk_R has been registered with $\mathcal{F}_{\text{KRK}}$ for some party R. $\mathcal{A}_{\text{gtag-CCA}}$ goes on to generate keys (sk_S, pk_S) for party S and registeres them with $\mathcal{F}_{\text{KRK}}$ as well before the challenge is created. Now when the challenger hands challenge (K_b, C^*) to $\mathcal{A}_{\text{gtag-CCA}}$ it can use sk_S to decrypt the challenge ciphertext and win the security game. □

B Efficient LPN-based Construction

In addition to our LWE-based construction in Section 7, we now provide an LPN and McEliece-based SB-KEM construction as well. This construction is based on the idea from [4], which in turn adapted the construction from [27] by replacing the trapdoor mechanism with McEliece. We augment this construction by adding sender IDs in such a way that we are still able to use the same public key replacement trick for our proof [27, 22]. Sender IDs are encoded in suitable matrix form which we denote by $M(S)$. We define an SB-KEM $\Sigma := (\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ as follows:

$\mathtt{gen}(1^\lambda)$:
- $\big((S, G', P), (G, t)\big) \leftarrow \mathtt{gen}_{\mathrm{McEliece}}(1^\lambda)$
- $C \xleftarrow{\$} \mathbb{Z}_2^{l \times n}$

$\hookrightarrow$ Return $(sk, pk) := \big((S, G', P), (G, C, t)\big)$.

$\mathtt{enc}(pk, S) = \mathtt{enc}\big((G, C, t), S\big)$:
- $r \xleftarrow{\$} \mathbb{Z}_2^l$.
- $k \xleftarrow{\$} \{0,1\}^{\nu(\lambda)}$
- $e_0, e_1 \leftarrow \mathcal{B}_\theta^n$
- $c_0 := r^\top \cdot G \oplus e_0 \big(\cong \mathtt{enc}_{\mathrm{McEliece}}((G, t), r)\big)$
- $c_1 = r^\top \cdot (C \oplus M(S)) \oplus e_1 \oplus \mathtt{encode}(k)$

$\hookrightarrow$ Return $(K, C) := \big(\mathtt{KDF}(k), (c_0, c_1)\big)$.

$\mathtt{dec}(sk, S, C) = \mathtt{dec}\big((S, G', P), S, (c_0, c_1)\big)$:
- $r \leftarrow \mathtt{dec}_{\mathrm{McEliece}}\big((S, G', P), c_0\big)$
- $k' := c_1 \oplus r \cdot (C \oplus M(S))$.
- $k \leftarrow \mathtt{decode}(k')$

$\hookrightarrow$ Return $K := \mathtt{KDF}(k)$.

For the encoding and decoding $(\mathtt{encode}, \mathtt{decode})$ we propose to use a suitable Goppa code, which is fixed for all parties. More details can be found in [4].

Theorem 5: *The SB-KEM $\Sigma = (\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ is IND-SB-CPA$_{SB\text{-}KEM}$ secure, given that both the McEliece indistinguishability assumption and the learning parity with noise decisional problem (LPNDP) hold. In particular, let $\mathcal{A}$ be an IND-SB-CPA$_{SB\text{-}KEM}$ adversary against the cryptosystem. Then there is a distinguisher $\mathcal{A}_{\mathrm{Goppa}}$ for Goppa codes and a distinguisher $\mathcal{A}_{\mathrm{LPN}}$ for LPNDP, such that for all $\lambda \in \mathbb{N}$:*

$$\mathsf{Adv}^{\text{SB-CPA}}_{\mathcal{A},\Sigma}(\lambda) \leq \mathsf{Adv}^{\mathrm{LPNDP}_\theta(2n,l)}_{\mathcal{A}_{\mathrm{LPN}}}(\lambda) + \mathsf{Adv}^{\mathrm{IND}}_{\mathcal{A}_{\mathrm{Goppa}},G_R}(\lambda).$$

Proofsketch.

Game 0: This is the IND-SB-CPA$_{\text{SB-KEM}}$ game.
Game 1: Swap C in the public key for $C \oplus S$.
Game 2: Replace Goppa code matrix G with uniformly randomly drawn matrix for the challenge. Reduce to Goppa code indistinguishability.

Game 3: Draw c_0 and $c_1 - \texttt{encode}(k)$ randomly. Reduce to LPN.
Game 4: Draw c_1 completely at random. Was a one-time pad on $\texttt{encode}(k)$ anyway. Challenge is now independent of k.
Game 5: Swap $K_0 = \texttt{KDF}(k)$ for uniform randomness. $\triangledown$